数值计算方法与MATLAB

杨扬

[qdyyang@zjut.edu.cn](mailto:qdyyang@zjut.edu.cn)

理A-108

第一讲： Introduction（绪论）

参考书：

《数值方法(Matlab版)》（第四版）

John H. Mathews, Kurtis D. Fink 著

（选用习题）

《An introduction to computational physics》 （阅读材料）

(计算物理学导论)

Tao Pang

世界图书出版公司

考核方式：

总分 ：100分；

10分 书面作业；

10分 电子作业+随堂小测验；

10分 期中考试

10分 到课率、课堂讨论

60分 考试。

注意：

1请假需有假条；

2 作业过期不记分；

应当具有的基本知识:

高等数学 如微积分、级数展开、微分方程

线形代数、概率统计\*

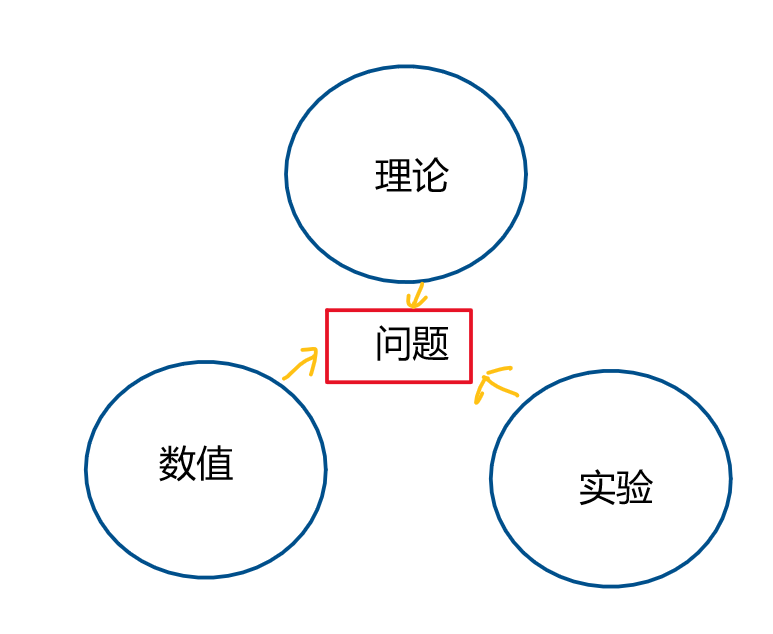
什么是数值计算方法，为什么要学：

你在赶路，如没赶上时代的高速列车，多少会失去一些东西。

——郝柏林

数值计算是当代科学研究的基本技能.

现代计算科学是实验科学和理论科学之外科学的第三分支



数值计算方法是当代自然科学研究的必备工具，甚至包括社会科学。

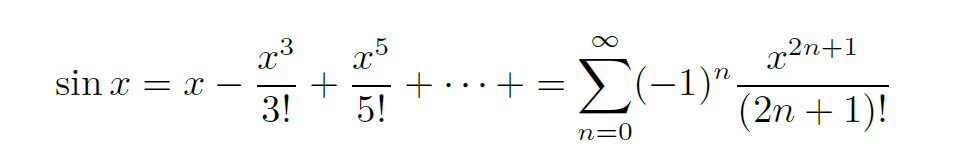
简单讲，数值计算方法是将科学计算问题变成一系列加减乘除运算的计算机指令合集（四则运算）。

例子：sin(x)，cos(x)

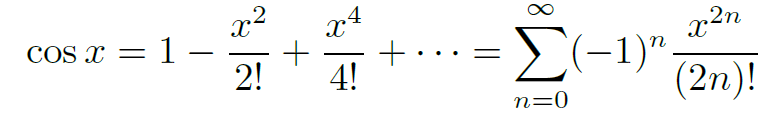
级数展开

****

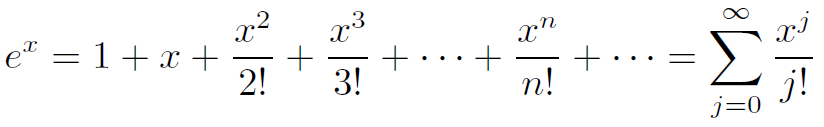
答： \* 当*x*较小时，可取前面几项作为函数的近似



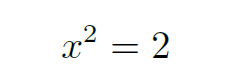
类似地，



例子：e≈2.718



例子：



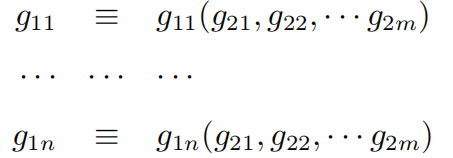
Naive method：半分法。

例子：深度学习

构造一个多元函数：



这里



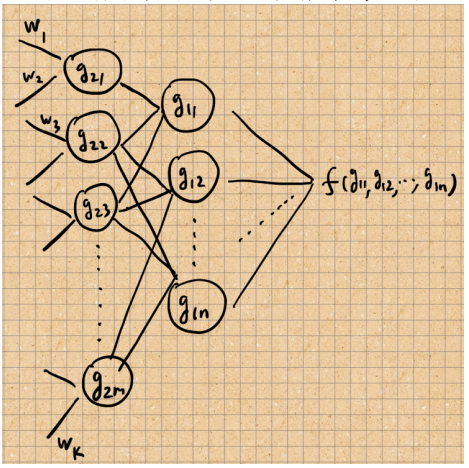
上面所有的又都是关于个自变量的多元函数



深度学习的目的是寻找到合适的的取值是的取极小值。即我们需要计算对任何的都有

，

一个简单粗糙的示意图如下



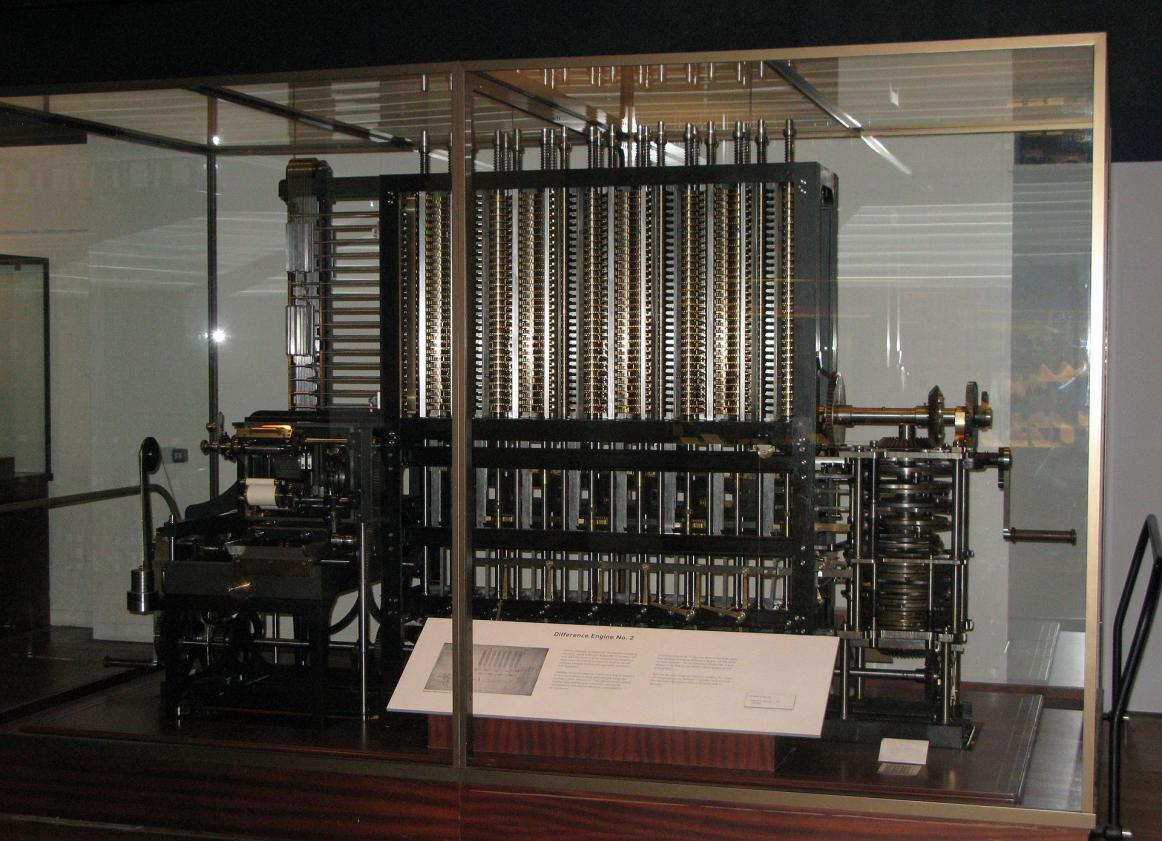
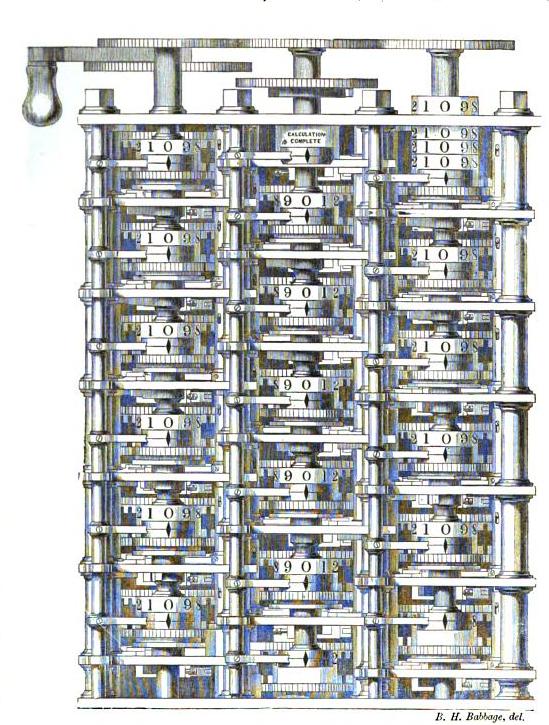
为求解上述问题，大家不断改进算法，直到本世纪，算力正好达到了深度学习的要求，才有了最近大数据和人工智能的提法。

计算机（算力）的发展

五千年前 巴比伦人发明珠算

19世纪初 英国人C Babbage提出计算机的构想(差分机)

但当时技术不够发达



19世纪末 西班牙人 Quevedo提出可用电动机械技术构造计算机，当时的技术可以提供足够的支持

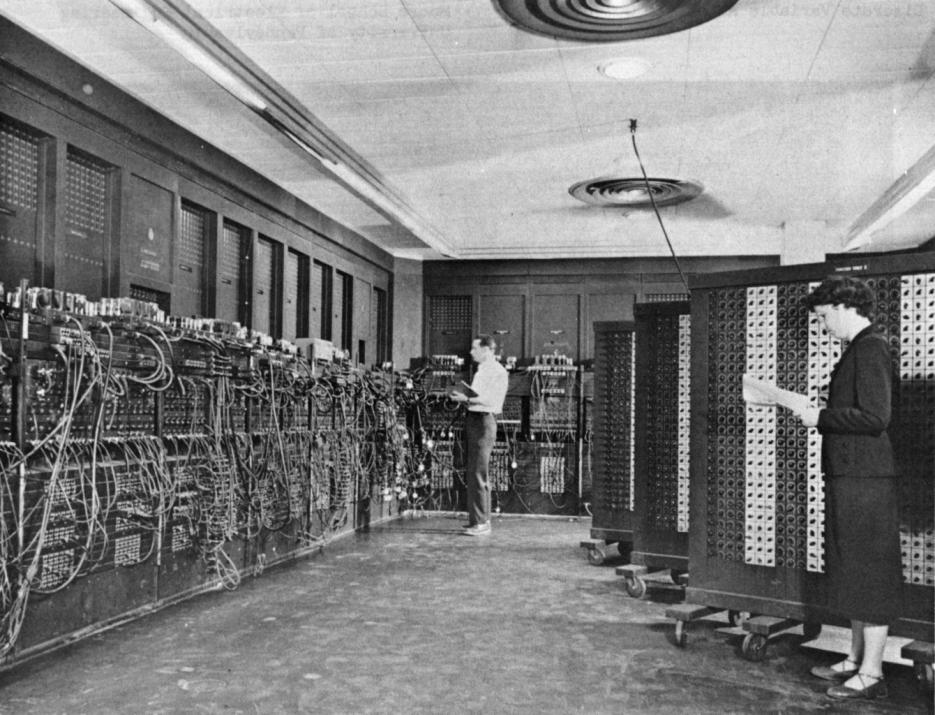
(但他没有资金)

1889年 美国人Hollerith造出第一台计算机，用于人口普查。卖掉这机器，Hollerith建立了IBM公司



1937年 美国理论物理学家Atanasoff制造出电子数字计算机，但没引起广泛注意

1945年 历史书常常引述，美国人Mauchly和Eckert制造的ENIAC为第一台电子计算机



Metropolis和Frankel用ENIAC研究核裂变

1. MENIAC I

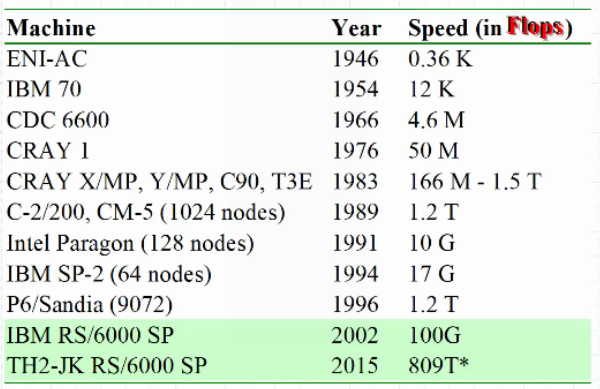
Metropolis提出Monte Carlo方法，并应用于核、反应固液相变的研究 （1953）

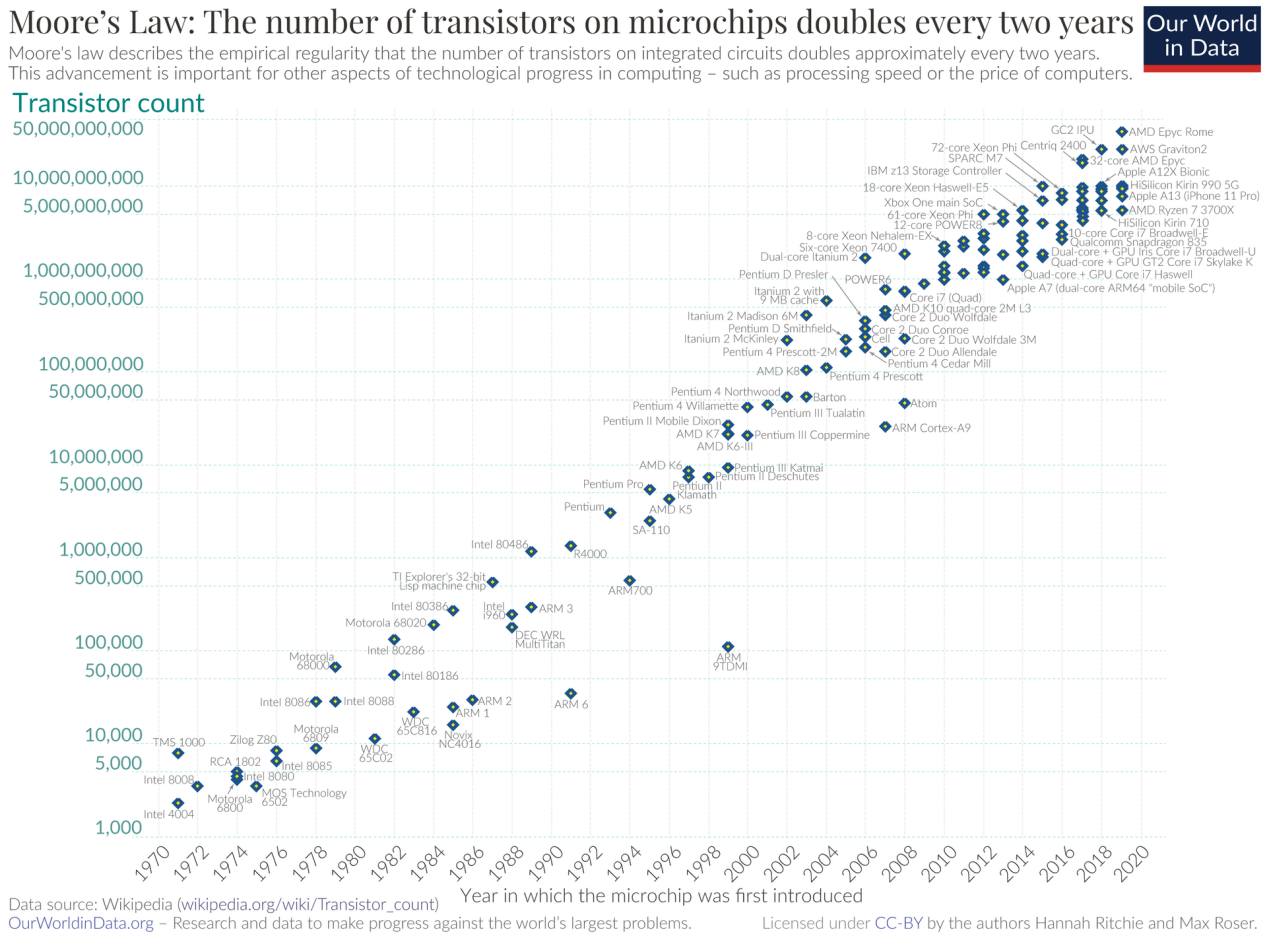
很多重要的研究工作由MENIAC I完成



80年代 我国的大机器







增长的前景

\* 计算机器件面临量子极限

计算科学的挑战

全球环境动力学模拟

DNA序列机制

药物设计

材料结构和器件

……

数值计算是计算科学的基础, 在各个领域有不同的作用，每个人亦有每个人的贡献。

计算与科学

例子：

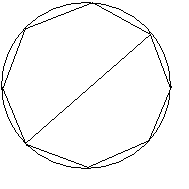
**圆周率的计算**

刘徽 割圆术

祖冲之、祖暅之（父子）

用正n边型逼近圆

周长 **** *（2r＝1）*



近似公式

****  (1.1)

便是的准确值，是待定常数

问题：(1.1) 怎么来的？

****作为n的函数，可对1/n 做级数展开

例如，

****

截断(1.1)式到第四项，可求得



与相差不大。

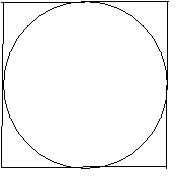
**计算的要点**：

* 问题的表达： 建立模型
* 计算技巧： 计算方法 ---- 我们课程的要点
* 计算能力的估计：研究计划

（如何截断，精度要求，需要的计算时间、资源等等）

上面的方法是确定论方法。***随机方法***在现代计算科学也十分重要

构造外切四边形



**均匀地随机地**在四边形‘抛石子’，统计石子总数*M，* 和圆内石子数*Ms，*则面积的比 

问题： 如何随机地‘抛石子’

设为均匀的随机数, 

可取



则  均匀地随机地分布在四边形内。

在计算机上如何产生 ？

延申阅读：**计算机上没有真正的随机数**，只有“近似均匀”的“看起来无规”的数列

这样的“赝随机数列产生器”质量标准 pseudo-random

• 足够长的数列

• 足够无规（随机）的数列

例如：可用



作图，图形应当均匀

• 速度足够快

最简单的随机数列



的长度为，速度非常快，随机性也不错，但对大规模计算，还不够好。

为了获得

可取 

如果不是专门研究算法，不适宜花太多精力在随机数上，抓到一个比较好的使用即可。

讨论：是否可以用上述思想解决积分问题

**比较随机方法和确定论方法**

* 随机方法较简单
* 随机方法较‘普适’

例如，简单应用抛石子的方法，可以计算不规则图形的面积

* 随机方法的误差收敛较慢



不过，对多自由度问题这不是弱点

* 随机方法依赖于相应的随机模型

即必须能构造出有效的模型，有时相当困难，

比如，牛顿方程和量子力学的动力学

算法

计算问题的逻辑步骤称计算机算法

牛顿方程 

时间分立化，一小段时间间隔为 



以上称Euler算法。给出初始位置和速度，可以一步一步求解。

课程学习讨论内容

* 误差分析；
* 非线性方程求解；
* 线性方程组求解；
* 插值与拟合；
* 微分与积分；
* 常微分方程求解。

计算机语言

50年代之前 打孔机、汇编

50-60年代 FORTRAN 针对数值计算编程容易

70 年代 C语言

80年代 C++

90年代 Python、 java 功能强大，但是对数值计算来说不够便捷.

其他 Julia等

MATLAB 优点：针对数值计算，比FORTRAN、python更容易入门(代码易读，计算效率高)；集成环境；有大量函数库。

计算机语言的学习不应当成一种理论学习，而是一种技能的学习，工多手熟 （参考英语学习）

Matlab简介

* MATLAB 历史

Prof. Cleve Moler（university of New Mexico）1970年代晚期发明，方便学生编程

The MathWorks Inc. 1984年成立

Cofounder Moler & Jack Little

MATLAB 为高级技术计算语言及交互式环境，用于算法实现，数据可视化，数据分析及数值计算，相比传统语言：C, C++, Fortran等，具有更快速的开发效果。

* Matlab 下载安装
* Matlab环境

主要组成部分:

1.桌面工具及开发环境

MATLAB desktop Command Window

Editor and debugger Folders

Help Workspace

2. 数学函数库

基础函数：sum, sine, cosine, and complex arithmetic

复杂函数： matrix inverse, matrix eigenvalues, Bessel functions, fast Fourier transforms, etc..

3.语言

高层矩阵数组语言、控制流程语句、函数、数据结构、输入\输出，面向对象编程特征

4.图形化

可显示2D/3D数据可视化，可定制图形，动画效果，GUI.

5.外部接口

外部接口库可与C/C++ 及 Fortran 与 MATLAB交互。

* Desktop操作桌面简介

指令窗

该窗是进行各种MATLAB操作的最主要窗口。在该窗内，可键入各种送给MATLAB运作的指令、函数、表达式；显示除图形外的所有运算结果；运行错误时，给出相关的出错提示。

当前目录浏览器

在该浏览器中，展示着子目录、M文件、MAT文件等。对该界面上的M文件，可直接进行复制、编辑和运行；界面上的MAT数据文件，可直接送入MATLAB工作内存。

工作空间浏览器

该浏览器默认地位于当前目录浏览器的后台。该窗口罗列出MATLAB工作空间中所有的变量名、大小、字节数；在该窗中，可对变量进行观察、图示、编辑、提取和保存。

历史指令窗

该窗记录已经运作过的指令、函数、表达式，及它们运行的日期、时间。该窗中的所有指令、文字都允许复制、重运行及用于产生M文件。

Matlab基本使用：

MATLAB最基本的规则和语法结构。

基本语法：变量=表达式

变量命名规则：区分大小写；

第一个字符必须是字母，可以有下划线，但是标 点运算符号空格不行；

避免与matlab的保留命令重名

分号的使用；语句后不加分号，程序运行后会在命令窗口返回计算结果。加上分号后则不会返回结果。做计算时通常会加分号，以节约时间。

省略号的使用 ... 语句太长可以换行写，换行时候加省略号

Exm： 1+2+2+3+4+5+...

6+7

MATLAB为一些数学常数（Math Contants）预定义了变量名

i或j；inf；nan；pi 等

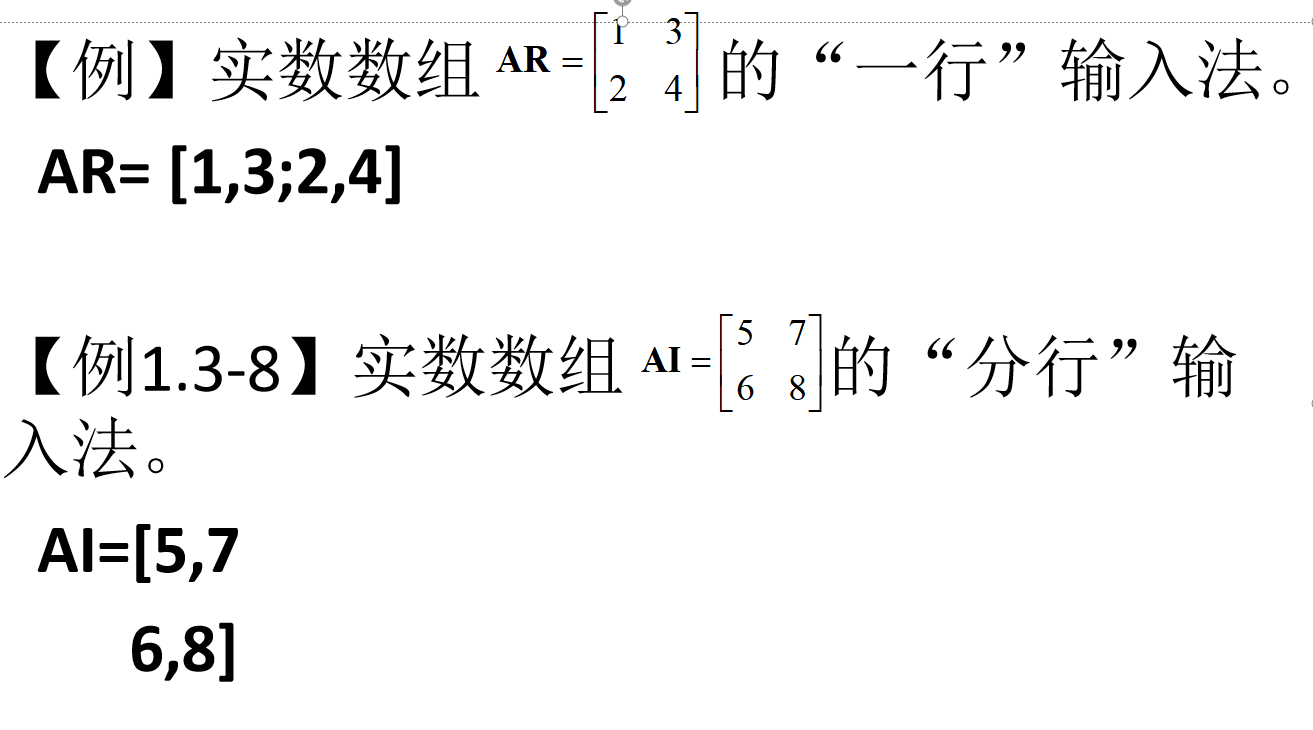
MATLAB的所有运算都是定义在复数域上的。为描述复数，虚数单位用预定义变量 i 或 j 表示。

>a=1+2i;b=2+3i;a\*b

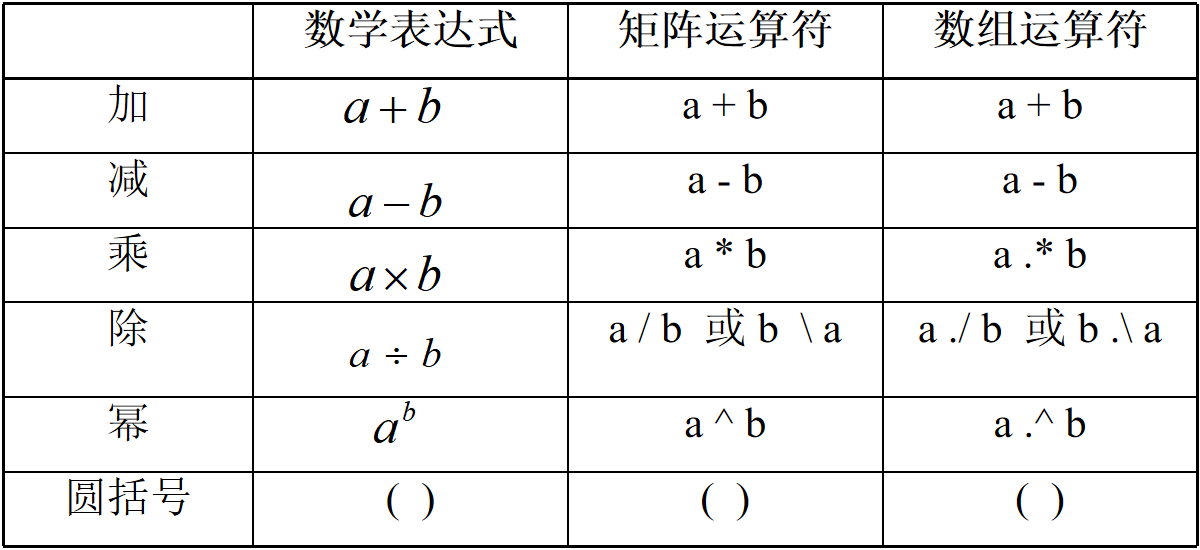
面向数组设计的运算——MATLAB特点

在MATLAB中，标量数据被看作 的数组（Array）数据。所有的数据都被存放在适当大小的数组中。

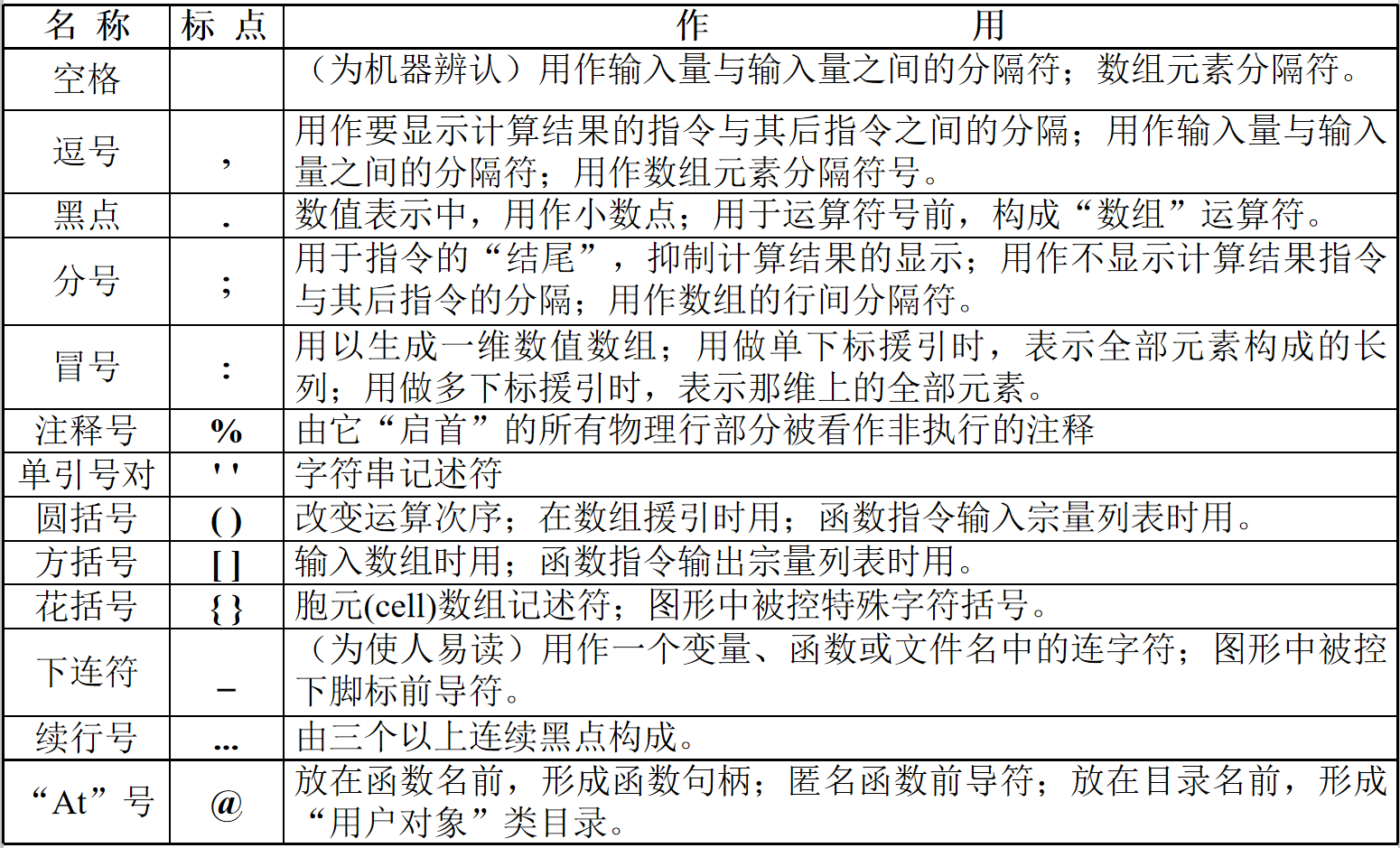
为加快计算速度（运算的向量化处理），MATLAB对以数组形式存储的数据设计了两种基本运算：数组运算 矩阵运算



运算符和表达式



指令行中的标点符号



指令窗的常用控制指令



当前目录、路径设置器和文件管理

当在指令窗中运行一条指令时，MATLAB是怎样从庞大的函数和数据库中，找到所需的函数和数据的呢？用户怎样才能保证自己所创建的文件能得到MATLAB的良好管理，又怎样能与MATLAB原有环境融为一体呢？

* MATLAB的所有（M、MAT）文件都被存放在一组结构严整的目录树上。MATLAB把这些目录按优先次序设计为“搜索路径”上的各个节点。MATLAB工作时，就沿着此搜索路径，从各目录上寻找所需的文件、函数、数据。
* 当用户从指令窗送入一个名为cont的指令后，MATLAB的基本搜索过程大抵如下：
* 检查MATLAB内存，若cont不是变量，则进行下一步。
* 检查cont是不是内建函数；假如不是，再往下执行。
* 在当前目录上，检查是否有名为cont的M文件存在；假如不是，再往下执行。
* 在MATLAB搜索路径的其他目录中，检查是否有名为cont的M文件存在。

工作空间浏览器和变量编辑器

查询指令 who及whos

例: 在指令窗中运用who, whos查阅MATLAB内存变量。

who

whos

从工作空间中删除变量和函数的指令 clear

最常用的几种格式：

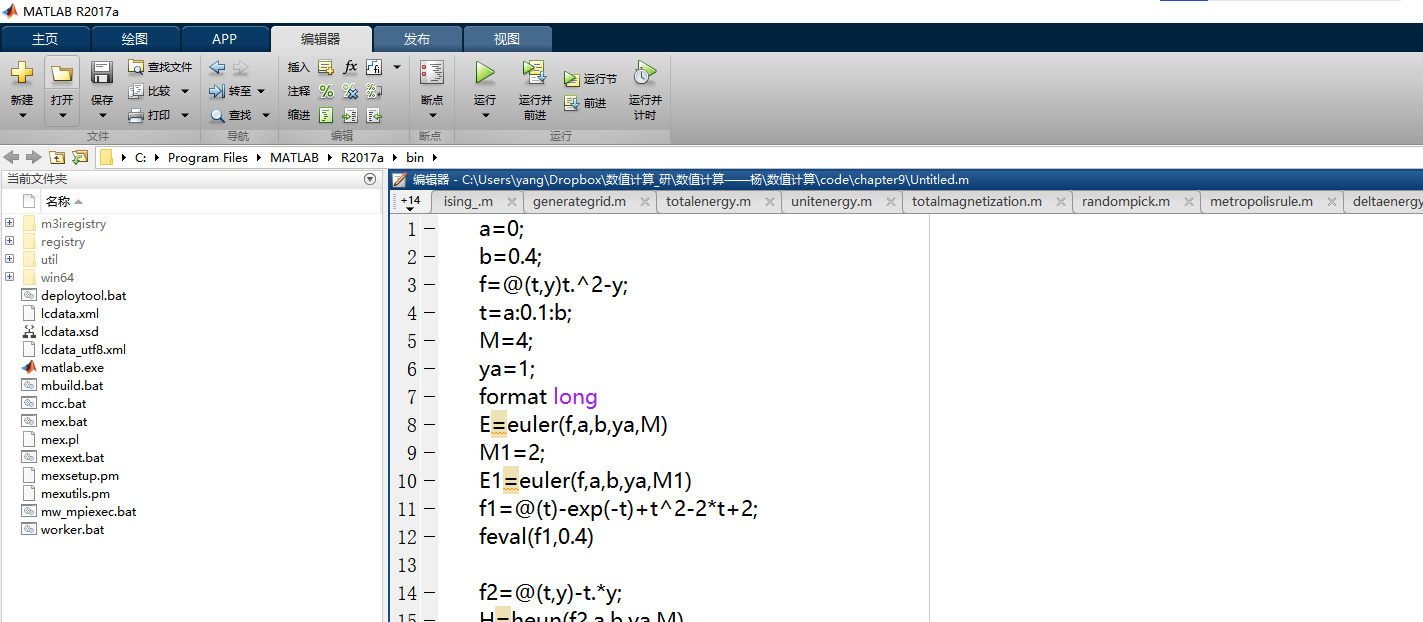
clear %清除工作空间中的所有变量

clear var1 var2 %清除工作空间中的var1和var2变量

clear all %清除工作空间中所有的变量

Editor 脚本编写初步





帮助系统及其使用

MATLAB作为一个优秀的科学计算软件，其帮助系统考虑了不同用户的不同需求，构成了一个比较完备的帮助体系。

常用帮助指令

help FunName 给出指定名称函数的使用方法

使用在线帮助

https://www.mathworks.com/help/?s\_tid=gn\_sup

常用命令

clf clc clear

help

load save

: , ;

pi i sin cos exp (expm) sqrt

rand

real imag abs angle

练习

1.数字1.5e2，1.5e3 中的哪个与1500相同？

2.请指出如下5个变量名中，哪些是合法的？

abcd-2 xyz­\_3 3chan a变量 ABCDefgh

3.设 a = -8 , 运行以下三条指令，问运行结果相同吗？为什么？

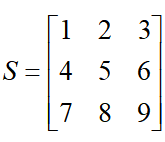
w1=a^(2/3)

w2=(a^2)^(1/3) %64^(1/3)

w3=(a^(1/3))^2 % (2\*exp(i\*pi/3))^2

4.指令clear, clc各有什么用处？

5.在MATLAB中产生二维数组



Chapter 0.1 矩阵与算符

* Array：按矩形排列的一组数

例子： 2 3 4 8 10

1 3 4 5 6

4.2 3.2 2.5 0 7 三行五列；3乘5的数列；3 by 5 array

三维数列：2 pages, 3 by 5 by 2





高维数列

常用的数列

2D array=’matrix’ (or matrices )

1D array=’vector’

Matlab=’matrix labortory’, 处理的是matrices。

绘图（plot）时，我们使用过1D数列，vector。

scalars

vectors

matrices

arrays

* 冒号（colon）的作用

一种算符，和加号减号一样类型

>> x=1:3:7;

>> y=1:100;

* 访问矩阵

行和列（row and columns）

columns

rows 

访问m行n列的元素 X(m,n)

修改m行n列的元素 X(m,n)=...

访问、修改多个元素

* 矩阵组合、变换





注意行数和列数的匹配

矩阵的转置

>>G=H’，G=H.’

G(2,1)=H(1,2)

* 算术

加法 >>Z=A+B?

行列数匹配

乘法 >>Z=A.\*B?

>>Z=A\*B?

“.\*” 对应位置元素相乘

“\*”线性代数中的矩阵乘法， 必须满足线性代数中的矩阵 乘法规则

除法 >>C=A\*B;

C\*inv(B)=A\*B\*inv(B) >>A=C/B

inv(A)\*C=inv(A)\*A\*B >>B=A\C

幂 >>x=2;y=3;z=x.^y;

如果x是矩阵？ >>x.^y ?

>>x^2? ans=x\*x

>>x^3”? ans=x\*x\*x

>>x.^2? ans=x.\*x

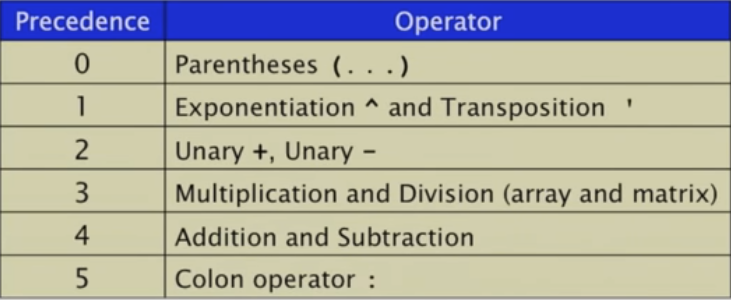
>>x.^3? ans=x.\*x.\*x

补充 >> A=[1:3;4:6]; >>A+3? >>A-2? >>2.\*A? >>2\*A?

* 算符优先级（operator precedence）

例子 >> 2+3\*4 乘法优先加法

编程中的优先级



1+2+3=（1+2）+3

1-2+3=（1-2）+3

>>2+3\*4 >>(2+3)\*4

>>1:3+10 =1:(3+10)=1:13

>>(1:3)+10 ?

利用括号调整优先级

>> help precedence

Chapter 0.2 数据类型

要求：理解matlab数值存储的类型；字符串；结构（struct）数组；cell数组。

* 数学意义上的实数
  + 可以无限大 例子 >> 10^300/10e-20
  + 可以无限精确 例子 >>1/3
* 计算机有存储限制
  + 大数有上限
  + 精度有限（不能存储的无理数）

Matlab（所有的计算机语言）存储的变量均有限（精度，大小）

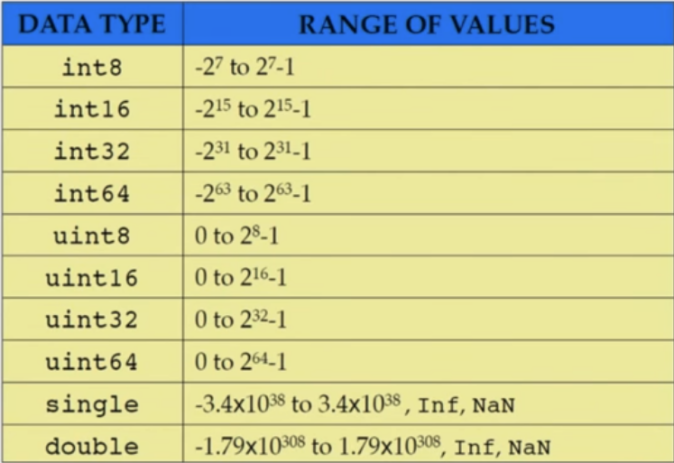
* 数值类型 双精度（matlab默认）

浮点表示 1.22=122\*10(-2)

64bits (8 bytes)

单精度 32bits

整数 8、16、32、64 bits （正负数或只包含正数）

 Inf=’infinity’, NaN=’not a number’

>>intArray = int8(array)

>>2/0

Inf

>> 0/0

NaN

尽量避免不同类型的数值计算 >>int8(2)+int16(4) ?

* 字符串

字符构成的矢量

>>a=3.14; >>size(a); >>a=’3.14’; >>size(a)

* 结构（structs）数组

结构数组可以包含不同的数值类型（array不可以）

结构数组内部包含的不是元素（elements），而是域（field）

一个结构数组可以包含另一个结构数组

>>a.numb=123; % 结构名： a；field name： numb，值： 123

>>class(a); >>class(a.numb);

>>a.name=’yang’

>>a.department.gonghao=’1234’ %结构中包含结构，department是a的field，gonghao是financial的field。

结构数组可以包含数列，数列不能包含结构数组

>>a(2).name=’liu’

>>a(2).department.gonghao=’1235’

* Cells

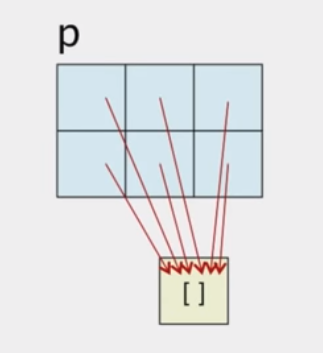
指针：每一个变量在内存中存储的位置对应一个地址，

指针是存储地址的变量

Matlab中把指针称为cell。

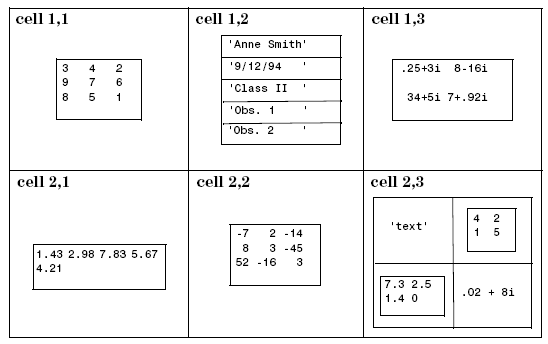
Cell同样可以存放不同类型的数据

>>p=cell(2,3);



Cell的访问和修改使用‘{}’

>>p{2,1}=pi



Cell很多情况下比struct更方便，因此更常用。

>>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}=4;

>>whos A

>>class(A{1,1})； >>class(A(1,1));

Cell可以包含Cell

>>A(1,1)={zeros(5)};

>>A(1,2)={{[5 2 8;7 3 0;6 7 3] 'Test 1';[2-4i 5+7i] {17 [ ]}}}

Chapter 0.3 函数

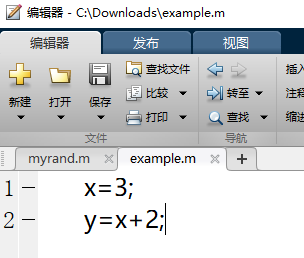
* 脚本文件（script）

每一个m文件都被称为一个脚本。

脚本是一系列matlab指令的合集。

脚本访问的变量都是workspace中的变量，本质上相当于按顺序 在指令窗口依次运行。

>>edit example



>> example

matlab的内建函数，比如“plot”，“sqrt”，“sin”...

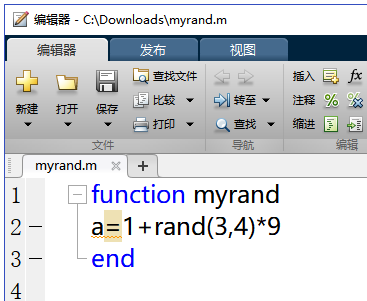
* 内建函数，也可以称为指令

例子：rand 产生0到1间平均分布的随机数

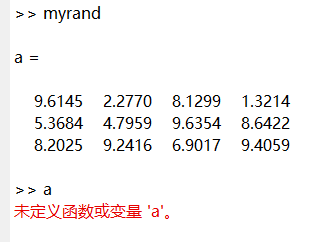
>>rand(3,4)

>>2+rand(3,4)\*5

如何定制需要的函数？



* 函数的输入变量与输出变量



函数有自己私有的临时workspace，函数中的变量存储在私人workspace，运算结束该空间被删除。

存储于这个空间的变量叫局域变量（local variable）。

* 如何把这些变量输出到外部的workspace？

function myrand → function a=myrand

* 假如要修正函数中的公式怎么办？例如 a=x+rand(3,4)\*y, x,y是我们根据需要设定的参数。

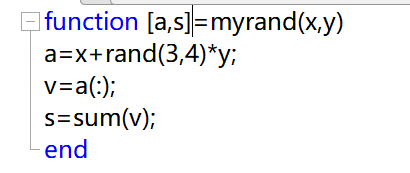
function a=myrand → function a=myrand(x,y)

目前 两个输入变量，标量

一个输出变量， 矩阵

* 假如要输出多个变量怎么办？

比如输出a矩阵外，同时输出a所有元素的和。



>>myrand(x,y) 语句没写全发生什么？只输出第一个变量结果。

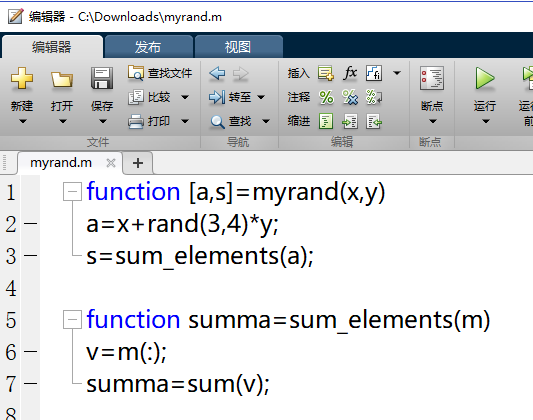
* 函数的正式定义

function [out\_arg1, out\_arg2,...]=function\_name(in\_arg1, in\_arg2, ...)

1. 函数名最好跟函数的作用有关
2. 不要使用内建函数名，如， plot， sqrt

* 子函数

如果我们考虑的问题很复杂，我们很可能将函数分为很多个，这些函数可以放在同一个m文件中。

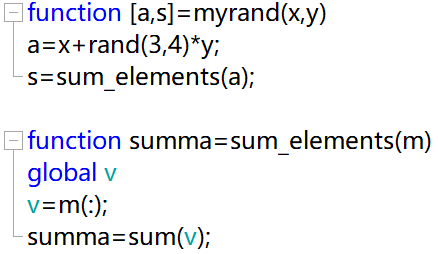


第一个函数被称为主函数。第二个被称为子函数或者局域函数，只能在主函数中被调用，不能从外部调用。

* Scope 局域变量 全局变量

局域变量只能从函数内部访问

>>v?



“global”定义全局变量

>> global v; >>[a,s]=myrand(1,2); >>v?

Warning: 初学者谨慎使用全局变量，因为任何函数都能修改全局变量的值，容易出错。