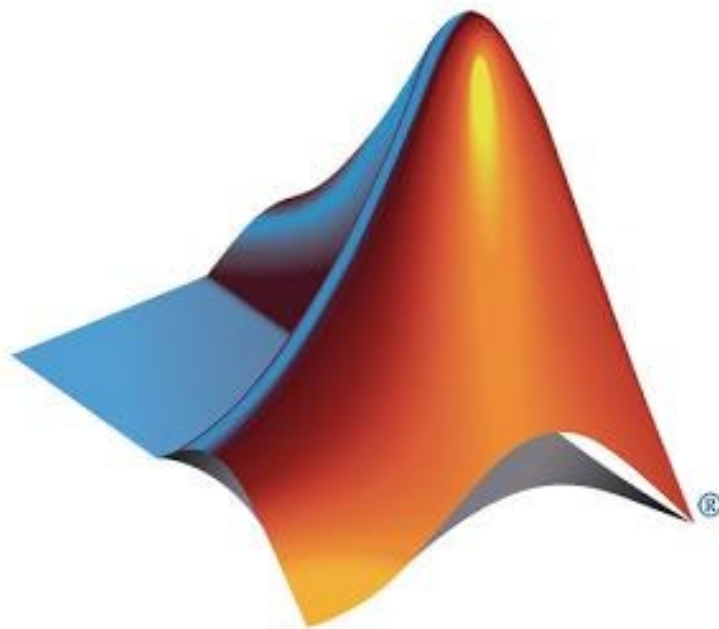
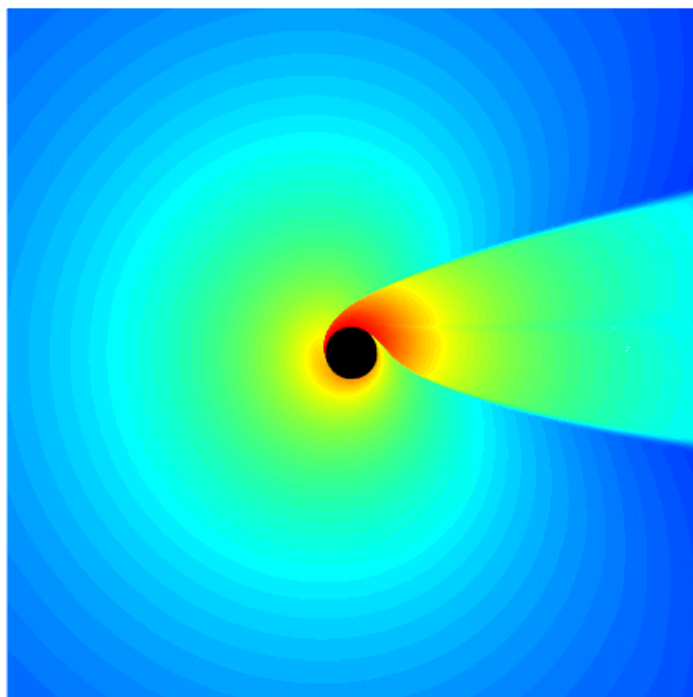


数学实验

Mathematical Experiments



MATLAB简介

Introduction to MATLAB

MATLAB概述

MATLAB (Matrix Laboratory, 矩阵实验室) 是由美国The MathWorks公司出品的商业数学软件。MATLAB是一种用于
算法开发、数据可视化、数据分析以及数值计算
的高级技术计算语言和交互式环境。

除矩阵运算、绘制函数/数据图像等常用功能外，
MATLAB还可用来创建用户界面，以及调用其它语言（包括C、C++、Java、Python、FORTRAN）编写的程序。

MATLAB 界面



MATLAB概述

丰富的内置函数和工具箱(toolbox):

数学和优化, 统计和数据分析, 深度学习与机器学习, 影像处理和电脑视觉, 控制系统设计和分析, 信号处理和通讯, 测试和测量, 金融建模和分析, 应用程序部署, 数据库连接和报表, 分布式计算

MATLAB的基本数据元素是**没有维数限制**的数组(array)。

- “0维”数组: 数/标量 (scalar)
- 1维数组: 向量 (vector)
- 2维数组: 矩阵 (matrix)

其他结构/类型: 如元胞(cell), 结构体(structure)等

北太天元简介

Introduction to Baltamatica

北太天元 概述

北太天元是面向科学计算与工程计算的国产通用型科学计算软件。本软件具有自主知识产权，提供科学计算、可视化与交互式程序设计环境，具备丰富的底层数学函数库，支持数值计算、数据分析、数据可视化、数据优化、算法开发等工作，可通过SDK与API接口，扩展各类学科与行业场景应用能力，为各领域科学家与工程师提供优质、可靠的科学计算平台。

标量、向量、矩阵的 创建和运算

标量的创建和运算

例1：球的半径为 $r=2$ ，求球的体积 $V=4*\pi*r^3/3$

例2：复数

MATLAB里 i 和 j 变量预留给了虚数单位，尽量不要用于其他用途

```
>> a=1+2i;b=5-4i;c=a/b
```

标量的创建和运算

例3：其它预定义变量

`pi`, `eps`, `Inf`, `NaN` 等

例4：用户变量

```
>> A=5+4*i; b=5-4*i; A*b    %未定义A*b的输出变量
```

```
ans=
```

```
41          %ans来接受计算结果
```

标量的创建和运算

例5: MATLAB创建变量并运行后, 它就存于工作空间, 可以被直接调用。

```
>> whos    %查询Workspace中的变量列表
```

```
>> clear %清除所有变量
```

```
>> clear A %清除变量A
```

向量的创建和运算

创建方法1：直接输入向量

```
>> x1=[1 2 4], x2=[1, 2, 1], x3=x1'
```

运行结果

```
x1 =    1    2    4
```

```
x2 =    1    2    1
```

```
x3 =
```

```
    1
```

```
    2
```

```
    4
```

向量的创建和运算

创建方法2：冒号创建向量

```
>> x1=3.4:6.7,
```

```
    x2=3.4:2:6.7,
```

```
    x3=2.6:-0.8:0
```

运算结果

```
x1 =
```

```
    3.4000    4.4000    5.4000    6.4000
```

```
x2 =
```

```
    3.4000    5.4000
```

```
x3 =
```

```
    2.6000    1.8000    1.0000    0.2000
```

向量的创建和运算

创建方法3：生成线性等分向量

指令 `x=linspace(a, b, n)` 在 $[a, b]$ 区间产生 n 个等分点(包括端点)

```
x=linspace(0, 1, 5)
```

结果

```
x =
```

```
0 0.2500 0.5000 0.7500 1.0000
```

向量的创建和运算

向量的运算

设 $x=[x1 \ x2 \ x3]$; $y=[y1 \ y2 \ y3]$;为两个三维向量, a, b 为标量。

向量的数乘: $a*x=[a*x1 \ a*x2 \ a*x3]$

向量的平移: $x+b=[x1+b \ x2+b \ x3+b]$

向量和: $x+y=[x1+y1 \ x2+y2 \ x3+y3]$

向量差: $x-y=[x1-y1 \ x2-y2 \ x3-y3]$

数的乘幂: 如 a^2

向量的创建和运算

元素群运算(四则运算)

$x.*y=[x1*y1 \ x2*y2 \ x3*y3]$ (元素群乘积)

$x./y=[x1/y1 \ x2/y2 \ x3/y3]$ (元素群右除, 右边的y做分母)

$x.\backslash y=[y1/x1 \ y2/x2 \ y3/x3]$ (元素群左除, 左边的x做分母)

$x.^5=[x1^5 \ x2^5 \ x3^5]$ (元素群乘幂)

$2.^x=[2^x1 \ 2^x2 \ 2^x3]$ (元素群乘幂)

$x.^y=[x1^y1 \ x2^y2 \ x3^y3]$ (元素群乘幂)

矩阵的创建和运算

直接输入法创建简单矩阵

小实验1:

```
>> A=[1 2 3 4; 5 6 7 8; 9 10 11 12]
```

```
B=[-1.3, sqrt(3); (1+2)*4/5, sin(5); exp(2), 6]
```

观察运行结果

矩阵的创建和运算

创建特殊矩阵

全1阵

`ones(n)`, `ones(m, n)`, `ones(size(A))`

全零阵:

`zeros(n)` , `zeros(m, n)`, `zeros(size(A))`

常常用于对某个矩阵或向量赋0初值

单位阵: `eye(n)`, `eye(m, n)`

随机阵:

`rand(m, n)`, `rand(n)=rand(n, n)`用于随机模拟, 常和
`rand('seed', k)`配合使用。

矩阵的创建和运算

矩阵的四则运算

矩阵的加减法分别使用运算符“+”、“-”运算符，格式与数字运算相同，但要求运算时两者是同阶的。例如：

```
>>a=[1, 2, 3;6, 8, 9];
```

```
>>b=[2, 1, 5;-3, 7, 8];
```

```
>>a+b
```

```
ans =
```

3	3	8
3	15	17

矩阵的创建和运算

矩阵的四则运算

矩阵的乘法使用运算符“*”，在使用时注意符合矩阵相乘的意义，即第一个矩阵的列数等于第二个矩阵的行数。例如：

```
>>a=[1, 2, 3;4, 5, 6];
```

```
>>b=[1, 1;2, 2;3, 3];
```

```
>>a*b
```

```
ans =
```

```
14    14
```

```
32    32
```

注意区别运算符“*”与“.*”（点乘 element-wise product）

矩阵的创建和运算

矩阵的四则运算

左除 “ \ ” :

求矩阵方程 $AX=B$ 的解; (A 、 B的行要保持一致)

解为 $X=A\backslash B$;

当A为方阵且可逆时有 $X=A\backslash B=\text{inv}(A)*B$;

右除 “ / ” :

求矩阵方程 $XA=B$ 的解 (A 、 B的列要保持一致)

解为 $X=B/A$,

当A为方阵且可逆时有 $X=B/A=B*\text{inv}(A)$

矩阵的创建和运算

小实验2:

求矩阵方程:

设A、B满足关系式: $AB=2B+A$, 求B。

其中 $A=\begin{bmatrix} 3 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 4 \end{bmatrix}$ 。

解: 有 $(A-2I)B=A$

程序 :

```
A=[3 0 1; 1 1 0;0 1 4];
```

```
B=inv(A-2*eye(3))*A,
```

```
B=(A-2*eye(3))\A
```

观察结果:

矩阵的创建和运算

数组运算符

运算	符号	说明
数组加与减 数组乘数组 数组乘方 数组除法 数与数组混合运算	$A+B$ 与 $A-B$ $A.*B$ $A.^B$ 左除 $A.\backslash B$, 右除 $B./A$ $k+A$, $k-A$, $k*A$, $A*k$, $A.^k$, $k.^A$, $k./A$	对应元素之间加减, 不加点 点运算只有点乘、点乘方、点除三个, 表示对应元素之间的运算. “ $.*$ ” 是一个整体, 点 “ $.$ ” 不能漏掉, “ $.$ ” 和 “ $*$ ” 之间也不能有空格. “ $.$ ” 和 “ $./$ ” 类似 将数 k 当作与 A 同阶的矩阵来做相应的数组运算. 如 $k+A=k*\text{ones}(\text{size}(A))+A$, $k./A=k*\text{ones}(\text{size}(A))./A$, 其他的数与数组混合运算以此类推

矩阵的创建和运算

分块矩阵:

1. 矩阵元素的标识 :

$A(i, j)$ 表示矩阵 A 的第 i 行 j 列的元素;

2. 向量标识方式 $A(vr, vc)$:

$vr=[i_1, i_2, \dots, i_k]$ 、 $vc=[j_1, j_2, \dots, j_u]$ 分别是含有矩阵 A 的行号和列号的单调向量。

$A(vr, vc)$ 是取出矩阵 A 的第 i_1, i_2, \dots, i_k 行与 j_1, j_2, \dots, j_u 列交叉处的元素所构成新矩阵。

矩阵的创建和运算

小实验3:

- 取出A的1、3行和1、3列的交叉处元素构成新矩阵A1
- 程序

```
A=[1 0 1 1 2;0 1 -1 2 3;  
    3 0 5 1 0;2 3 1 2 1];  
vr=[1, 3];vc=[1, 3];  
A1=A(vr, vc)
```

- 观察结果

矩阵的创建和运算

小实验4:

- 将A分为四块，并把它们赋值到矩阵B中，观察运行后的结果。

- 程序

$A_{11}=A(1:2, 1:2), A_{12}=A(1:2, 3:5),$

$A_{21}=A(3:4, 1:2), A_{22}=A(3:4, 3:5)$

$B=[A_{11} \ A_{12}; A_{21} \ A_{22}]$

- 猜测结果：A和B的关系？

矩阵的创建和运算

小实验5:

- 修改矩阵A，将它的第1行变为0。
- 程序：

```
A=[1 0 1 1 2;0 1 -1 2 3;  
    3 0 5 1 0;2 3 1 2 1];  
A(1,:)= [0 0 0 0 0]; A
```

小实验6:

- 删除上面矩阵A的第1、3行。
- 程序：
- $A([1, 3], :)=[]$
- 结果

常用的内置数学函数

三角函数和双曲函数

名称	含义	名称	含义	名称	含义
<code>sin</code>	正弦	<code>csc</code>	余割	<code>atanh</code>	反双曲正切
<code>cos</code>	余弦	<code>asec</code>	反正割	<code>acoth</code>	反双曲余切
<code>tan</code>	正切	<code>acsc</code>	反余割	<code>sech</code>	双曲正割
<code>cot</code>	余切	<code>sinh</code>	双曲正弦	<code>csch</code>	双曲余割
<code>asin</code>	反正弦	<code>cosh</code>	双曲余弦	<code>asech</code>	反双曲正割
<code>acos</code>	反余弦	<code>tanh</code>	双曲正切	<code>acsch</code>	反双曲余割
<code>atan</code>	反正切	<code>coth</code>	双曲余切	<code>atan2</code>	四象限反正切
<code>acot</code>	反余切	<code>asinh</code>	反双曲正弦		
<code>sec</code>	正割	<code>acosh</code>	反双曲余弦		

指数函数

名称	含义	名称	含义	名称	含义
<code>exp</code>	E 为底的指数	<code>log10</code>	10 为底的对数	<code>pow2</code>	2 的幂
<code>log</code>	自然对数	<code>log2</code>	2 为底的对数	<code>sqrt</code>	平方根

复数有关的函数

名称	含义	名称	含义	名称	含义
<code>abs</code>	绝对值	<code>conj</code>	复数共轭	<code>real</code>	复数实部
<code>angle</code>	相角	<code>imag</code>	复数虚部		

圆整函数和求余函数

名称	含义	名称	含义
ceil	向 $+\infty$ 圆整	rem	求余数
fix	向 0 圆整	round	向靠近整数圆整
floor	向 $-\infty$ 圆整	sign	符号函数
mod	模除求余		

矩阵变换函数

名称	含义	名称	含义
fipplr	矩阵左右翻转	diag	产生或提取对角阵
fipud	矩阵上下翻转	tril	产生下三角
fipdim	矩阵特定维翻转	triu	产生上三角
Rot90	矩阵反时针 90 翻转	det	行列式的计算

与矩阵有关的常用内置函数：

- `det(A)` : 方阵的行列式(determinant)
- `rank(A)`: 矩阵的秩(rank)
- `eig(A)`: 方阵的特征值和特征向量(eigenvalue and eigenvector)
- `trace(A)`: 矩阵的迹(trace)
- `rref(A)`: 初等变换阶梯化矩阵A
- `svd(A)`: 矩阵奇异值分解(singular value decomposition)
- `cond(A)`: 矩阵的条件数(condition number)

注意:

- 将 sin, cos, exp, log, 等用于向量和矩阵, 都是 `element-by-element` applied
- 运行以下程序, 观察结果

```
>> A=[1, 1;2, 3]
```

```
>> exp(A)
```

思考, 如何用MATLAB计算 e^A ?

(假设A可以实对角化)

用于简单的数据分析的函数：

`max` (求最大)、`min` (求最小)、`mean` (求平均值)、`sum` (求和)、`std` (求标准差Standard Deviation)、`cumsum` (求累积和 Cumulative Sum)、`median` (求中值)、`diff` (差分 Difference)、`sort` (升序排列)、`sortrows` (行升序排列) 等等。

1. 当数据为行向量或列向量时，函数对整个向量进行计算。
2. 当数据为矩阵时，命令对列进行计算，即把每一列数据当成同一变量的不同观察值。

关系运算(Relational Operators) 和逻辑运算(Logical Operator)

关系运算符和逻辑运算符的结果是特殊的逻辑数组(Logical Array). 在MATLAB中, “真(True)”用1表示, “假(False)”用0表示, 逻辑运算中所有非零元素均作为1 (True) 处理。关系与逻辑运算的两个操作对象需具有相同的尺寸(size)或者其中一个为标量。

运算符	含义	运算符	含义
<	小于	~=	不等于
<=	小于或等于	&或&&	与
>	大于	或	或
>=	大于或等于	~	非
==	等于		

小实验:

```
>> A = -2:4, B=4:-1:-2
```

```
>> A>B
```

```
>> A==B
```

```
>> A&B
```

```
>> A|B
```

```
>> abs(A) >=2
```

```
>> find( abs(A) >=2 ) % 返回绝对值>=2的元素下标
```

```
>> any( abs(A) >=2 ) %若A存在绝对值>=2的元素，则返回1
```

```
>> all( abs(A) >=2 ) %若A所有元素绝对值>=2，则返回1
```

控制语句(Control Statements)

- If: If evaluates a logical expression and executes a group of statements based on the value of the expression.
- Syntax of **If Statement**

若只有一个选择，使用格式一：

if 条件(表达式1)

语句组

end

如果在表达式1中的所有元素为真(非零)，那么就执行 if 和 end 语言之间的语句。

- If: If evaluates a logical expression and executes a group of statements based on the value of the expression.
- Syntax of **If Statement**

假如有两个选择，使用格式二：

if 条件(表达式1)

 语句组1

else

 语句组2

end

在这里，如果表达式1为真，则执行语句组1；如果表达式是假，则执行语句组2。

当有三个或更多的选择时，使用格式三：

if 条件1(表达式1)

语句组1

elseif 条件2(表达式2)

语句组2

.....

elseif 条件m(表达式m)

语句组m

else

语句组m+1

end

```
>> a=7
```

```
a =
```

```
7
```

```
>> if a>0
```

```
disp('a is positive');
```

```
elseif a<0
```

```
disp('a is negative')
```

```
else
```

```
disp('a is zero')
```

```
end
```

- Switch Statement

Switch, case, and otherwise: Switch executes certain statements based on the value of a variable or expression. Its basic form is

switch 表达式

 case 值1

 语句组1

 case 值2

 语句组2

 case 值m

 语句组m

 otherwise

 语句组m+1

end

Example:

```
>> mynumber=input('enter a number')
enter a number -1
mynumber =
    -1
>> switch mynumber
case -1
disp('negative one')
case 0
disp('zero');
case 1
disp('positive one');
otherwise
disp('other value');
end
```

for Statement

格式:

```
for    循环变量=表达式1: 表达式2: 表达式3  
    循环体语句  
end
```

其中表达式1的值为循环变量的初值，表达式2的值为步长，表达式3的值为循环变量的终值。步长为1时，表达式2可以省略。

for Statement: A for loop is a repetition control operation that allows us to accurately write a loop that wants to perform a particular number of times.

格式:

```
for  循环变量=表达式1: 表达式2: 表达式3  
    循环体语句  
end
```

其中表达式1的值为循环变量的初值，表达式2的值为步长，表达式3的值为循环变量的终值。步长为1时，表达式2可以省略。

小实验：求

$$S_{100} = \sum_{n=1}^{100} \frac{1}{n^2}$$

for Statement: A for loop is a repetition control operation that allows us to accurately write a loop that wants to perform a particular number of times.

格式:

```
for  循环变量=表达式1: 表达式2: 表达式3  
    循环体语句
```

end

其中表达式1的值为循环变量的初值，表达式2的值为步长，表达式3的值为循环变量的终值。步长为1时，表达式2可以省略。

小实验：求
$$S_k = \sum_{n=1}^k \frac{1}{n^2}$$

for Statement: A for loop is a repetition control operation that allows us to accurately write a loop that wants to perform a particular number of times.

格式:

```
for  循环变量=表达式1: 表达式2: 表达式3  
    循环体语句
```

end

其中表达式1的值为循环变量的初值，表达式2的值为步长，表达式3的值为循环变量的终值。步长为1时，表达式2可以省略。

小实验:

$$\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6} \approx \underline{1.6449}$$

while Statement: The while loop repeatedly executes statements while a specified statement is true.

格式:

```
while 条件表达式  
    循环体语句  
end
```

小实验: 求

$$S_{100} = \sum_{n=1}^{100} \frac{1}{n^2}$$

M-文件

M文件有两种形式：命令式文件(Script)与函数式文件(Function)。

- 命令式文件就是命令行的简单叠加，**MATLAB**会自动按顺序执行文件中的命令。这样就解决了用户在命令窗中运行许多命令的麻烦，还可以避免用户做许多重复性的工作。
- 函数式文件主要用于解决参数传递与函数调用问题，它的第一句以**function** 语句为标识。函数式文件可以有返回值，也可以只执行操作无返回值。函数式文件在**MATLAB**中应用十分广泛，**MATLAB**所提供的绝大多数功能函数都是由函数式文件实现的。函数式文件在执行之后，只保留最后结果，不保留中间过程，所定义的变量也仅在函数内部起作用，并随调用的结束被清除。

在编写函数式文件时，**要注意文件名与函数名的对应问题，两者最好是保持一致(同名)**，这样函数调用时不易出错。另外，编写函数式文件，要养成注释的习惯，以方便自己或其他用户调用。

- **例1： 编写程序计算前n个斐波那契(Fibonnaci)数**

```
function bb=examl_1(n)
    %Fibonnaci计算
    bb=zeros(1,n);
    bb(1)=1;
    bb(2)=1;
    for i=3:n
        bb(i)=bb(i-1)+bb(i-2);
    end
```

编写完毕后，以examl_1.m 为文件名存盘。然后在命令窗口中执行：

```
>>f =examl_1(10)
```

- 例2: 设 $f(x) = \begin{cases} x^2 + 1, & x > 1 \\ 2x, & 0 < x \leq 1 \\ x^3, & x \leq 0 \end{cases}$, 求 $f(3)$, $f(0.5)$, $f(-1.5)$

```
function y = examl_4(x)
```

```
if x>1
```

```
    y=x^2+1;
```

```
elseif x>0&x<=1
```

```
    y=2*x;
```

```
else
```

```
    y=x^3;
```

```
end
```

内联函数和匿名函数:

比较简单的函数表达式可以不用写成外部M函数，而是用更简捷的内联函数或匿名函数方式出现在命令行中.使用格式分别为

`fun=inline('expr',arg1,arg2,...)`

`fun=@(arg1,arg2,...)expr`,这里 `expr`为函数表达式，`arg1,arg2,...`为自变量名字字符串

内联函数和匿名函数:

```
>>k=2; fname=@(m)sum(1./(1:m).^k);
```

```
>>fname(5000)
```

```
ans =
```

```
1.6447
```

```
>>k=3:fname=@(m)sum(1./(1:m).^k);
```

%注意k值修改后匿名函数要重新执行一遍

```
>>fname(5000)
```

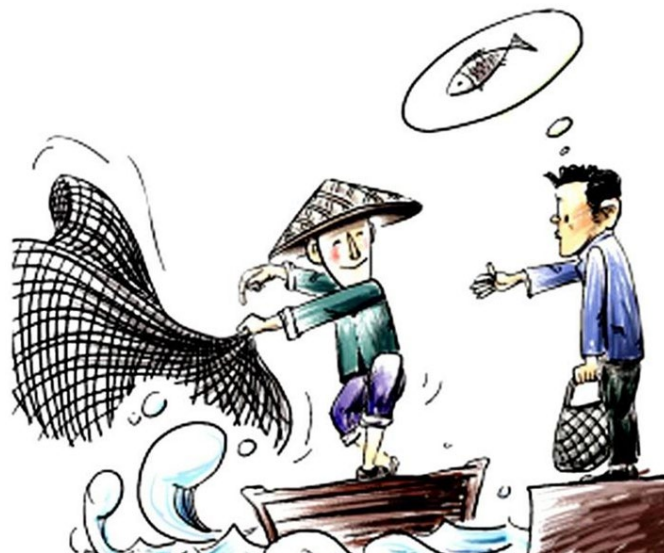
```
ans =
```

```
1.2021
```

➤ **提醒**: 初学者在M文件的建立和保存上经常出现下列几种错误:

- (1)将M函数错误地写在了命令行窗口(应在“编辑器”窗口);
 - (2)文件修改后没有保存;
 - (3)文件保存的文件夹不在当前文件夹和MATLAB路径中;
 - (4)文件名使用了常量或内存中的变量, 如1.m, pi.m等;
 - (5)文件名用了**减号、空格等非法字符**, 如eg2-1.m, eg2.1.m等;
 - (6)文件名与MATLAB内置(Build-in)函数和其他内部函数冲突, 如sin.m, mesh.m, fitfun.m等.
- 能执行的M文件的位置可用which查到, 并可用type显示文件内容.

“授人以鱼 不如授人以渔”



- ✓ 结合MATLAB入门书籍，多动手练习；
- ✓ 结合实验目标，网上查找相应MATLAB函数/功能学习
- ✓ 善于运用MATLAB强大的帮助(Help)系统

MATLAB的帮助系统

MATLAB的一个突出的优点是帮助系统非常完善，适合自学。

从总体上来看它的帮助系统大致可以分为3大类：

- 1) 联机帮助系统 `doc` (MATLAB文档)
- 2) 联机演示系统 `examples` (MATLAB示例)
- 3) 命令窗口查询帮助系统

MATLAB的帮助系统

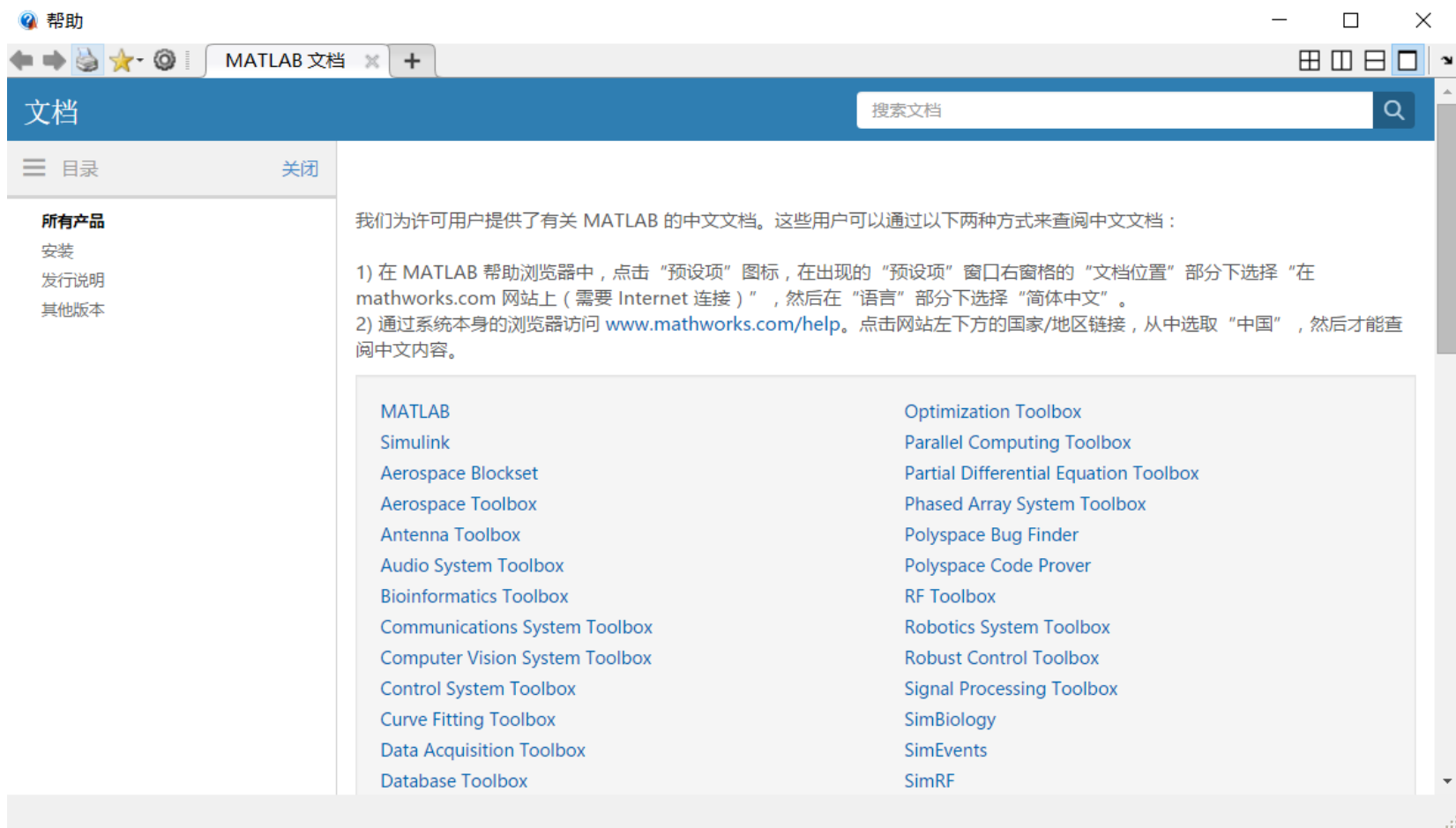
1、联机帮助系统doc (MATLAB文档)

打开方式

- 方式1：主页（HOME）下的问号按钮（Help；帮助）->(doc；文档)
- 方式2：命令窗口输入命令doc

MATLAB的帮助系统

1、联机帮助系统doc (MATLAB文档)



MATLAB的帮助系统

2、联机演示系统 examples (MATLAB示例)

打开方式

- 方式1：主页（HOME）下的问号按钮（Help；帮助）->(examples；示例)
- 方式2：命令窗口输入命令demos

MATLAB的帮助系统

3、命令查询系统

命令字	功能
help	显示帮助系统所有项目，用目录形式列出
help+<函数名>	查询与函数有关的帮助
lookfor+<关键字>	查询包含关键字的有关函数
demo	打开演示窗口
info	显示MATLAB的一般信息
whatsnew	列出MATLAB的新有信息

MATLAB的帮助系统

命令行窗口

```
>> help
```

帮助主题:

[Documents\MATLAB](#)

[matlab\datafun](#)

[matlab\datatypes](#)

[matlab\elfun](#)

[matlab\elmat](#)

[matlab\funfun](#)

[matlab\general](#)

[matlab\iofun](#)

[matlab\lang](#)

[matlab\matfun](#)

[matlab\ops](#)

fx [matlab\polyfun](#)

- (没有目录文件)
- Data analysis and Fourier transform
- Data types and structures.
- Elementary math functions.
- Elementary matrices and matrix mani
- Function functions and ODE solvers.
- General purpose commands.
- File input and output.
- Programming language constructs.
- Matrix functions - numerical linear
- Operators and special characters.
- Interpolation and polynomials.

MATLAB的帮助系统

命令行窗口

```
>> help sin
```

```
sin    Sine of argument in radians.
```

```
sin(X) is the sine of the elements of X.
```

```
See also asin, sind.
```

```
sin 的参考页
```

```
名为 sin 的其他函数
```

fx >> |

• 在线帮助命令汇总：

help 显示MATLAB主题文件夹，包括所有已安装的工具箱

help子文件夹名 显示子文件夹中所有MATLAB系统命令及函数

help命令或函数 显示该命令或函数的说明部分

doc 启动MATLAB的超文本帮助文件

doc命令或函数 显示该命令或函数的超文本帮助

docsearch关键字 超文本搜索与该关键字有关的内容

lookfor关键字 显示与该关键字有关的命令和函数

type M文件主名 显示M文件程序代码

which M文件主名 显示指定的MATLAB文件的路径

demo 演示MATLAB功能

MATLAB主要提供了两种形式的帮助系统：纯文本帮助和超文本帮助.

- 纯文本帮助使用help命令，较简捷，直接显示在“命令行窗口”.
- 超文本帮助使用doc命令，链接检索，介绍更详细，并提供了较多实例，是边用边学MATLAB的最佳方法.

```
>>help    % 显示MATLAB及其工具箱的主题文件夹，其中有graph3d

>>help graph3d  %显示三维图形主题文件夹内所有命令和函数，其中
                %有mesh

>>help mesh      %显示M函数mesh的用法说明(即其M文件的注释部分)，
                %最后列出了一些相关的函数

>>doc mesh        %启动M函数超文本帮助，提供较多例子

>>which mesh      %显示M函数 mesh所在的文件夹

>>type mesh       %显示函数mesh的M文件程序代码

>>lookfor surface  %显示MATLAB搜索路径中凡是第一行注释
                  %surface的命令和函数，其中有函数mesh

>>docsearch surface  %比lookfor搜索出更多内容
```

MATLAB远不止是一个计算器

实验1

$3n+1$ Problem

$3n+1$ Problem (Collatz Conjecture):

- 又称为奇偶归一猜想、 $3n+1$ 猜想、冰雹猜想、角谷猜想、哈塞猜想、乌拉姆猜想或叙拉古猜想。
- $3n+1$ 猜想：对于每一个正整数，如果它是奇数，则对它乘3再加1，如果它是偶数，则对它除以2，如此循环，最终都能够得到1。

$$f(n) = \begin{cases} n/2 & \text{if } n \equiv 0 \\ 3n + 1 & \text{if } n \equiv 1 \end{cases} \pmod{2}.$$

$3n+1$ Problem (Collatz Conjecture):

- 例如, 对于初值 $n = 1, 2, 3, 4, 5$, 可以验证该猜想成立:

$1 \rightarrow 4 \rightarrow 2 \rightarrow 1 \rightarrow \dots$

$2 \rightarrow 1 \rightarrow 4 \rightarrow 2 \rightarrow 1 \rightarrow \dots$

$3 \rightarrow 10 \rightarrow 5 \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1 \rightarrow \dots$

$4 \rightarrow 2 \rightarrow 1 \rightarrow \dots$

$5 \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1 \dots$

$\dots\dots$

3n+1 Problem (Collatz Conjecture):

请进行如下数学实验:

(1) 请写一个MATLAB函数, 实现

$$f(n) = \begin{cases} n/2 & \text{if } n \equiv 0 \\ 3n + 1 & \text{if } n \equiv 1 \end{cases} \pmod{2}.$$

(2) 请编写MATLAB程序 (写在M文件), 检验 3n+1 猜想 是否对于如下初值 n 均成立: $1 < n < 10^4$?

(3) 观察从以上n开始产生的序列(series), 最后是否都落于4->2->1的循环中?

(4) 请优化你的程序, 使得计算速度尽可能的快(学习使用tic, toc命令计时)。

鼓励试着观察/探索其它潜在规律, 并结合程序实验进行总结。