

MATLAB 软件学习笔记

By John.Lee or emc1207

第一部分 MATLAB 软件入门基础

1.1 MATLAB 软件介绍

在欧美各高等学校，MATLAB 软件成为线性代数、自动控制理论、数字信号处理、时间序列分析、动态系统仿真、图像处理等诸多课程的基本教学工具，成为本科生、硕士生和博士生的必须掌握的基本技能。

主要用途：

(1) 数值和符号计算；

(2) 绘图；

(3) 一种语言体系，也可以方便地与 Fortran、C 等语言接口。

(4) 工具箱 (Toolbox)，分为功能性和学科性两种类。同时，其工具箱的数据文件代码完全开发，用户也可以开发自己的工具箱。

功能性工具箱主要用来扩充 MATLAB 软件的符号计算功能、图视建模仿真功能、文字处理功能以及硬件实时交互功能。这种功能性工具箱用于多种学科。而学科性工具箱是专业性比较强的，如控制工具箱 (Control Toolbox)、信号处理工具箱 (Signal Processing Toolbox)、通信工具箱 (Communication Toolbox) 等都属此类。

MATLAB 语言的主要特点：

(1) 语法规则简单。与其他编程语言相比更接近于常规数学表达，对于数组变量的使用，不需类型声明，也无需事先申请内存空间。

(2) 提供了数以千计的计算函数，极大的提高了用户的编程效率。

(3) 是一种脚本式 (scripted) 的解释型语言，无论是命令、函数或变量，只要在命令窗口的提示符下键入并以回车键结束，则 MATLAB 都会予以解释执行。

(4) 可移植性，可跨平台运行。MATLAB 软件可以运行在很多不同的计算机系统平台上，包括大部分的 UNIX 和 Linux 系统，其编写的程序对应的数据文件是一致的，绘图功能也与平台无关。

1.2 MATLAB 系统环境与数据操作

1.2.1 系统路径设置

两个路径概念：

(1) 当前目录窗口：指 MATLAB 运行时的工作目录。只有在当前目录和搜索路径下的文件、函数才可以被运行和调用；如果没有特殊指明，数据文件也将存放在当前目录下。

(2) 搜索路径：指 MATLAB 执行过程中对变量、函数和文件进行搜索的

路径。

搜索路径是使用 MATLAB 时一个很重要的概念, MATLAB 具有大量的工具箱, 用户在使用时也会设置大量的用户文件夹和文件, 怎样让 MATLAB 能够识别出这些文件在哪里, 这就是搜索路径需要确定的内容。

(注: MATLAB 启动后的默认目录是 C:\MATLAB\BIN 若不建子目录则 MATLAB 环境产生的数据文件就登陆在这个缺省目录上。为了更加便于用户创建、修改 M 文件和其他文件, 用户应建立自己的工作目录。)

键入字符识别的搜寻次序为:

->>是否为变量

->>是否为函数

->>是否为当前目录下的 M 文件

->>是否为搜索路径下其他 M 文件

系统路径设置方法

(1) path 指令, 可设置扩展(临时)搜索路径。例:

path(path, 'C:\tools\mydir')

(2) cd 指令, 可设置当前工作目录, 该目录必须已存在。例:

cd E:\MATLAB

(3) userpath 指令, 可修改运行时默认路径。例:

userpath('D:\MATLAB')

(4) pathtool 指令, 可直接更改、添加搜索路径。

1.2.2 变量的基本表达

变量命名规则: 以英文字母开头, 可包括英文字母、数字和下划线, 区分大小写, 最多可包含 63 个字符 (6.5 及以后版本)。

变量书写特点:

(1) 无需声明

(2) 末尾有分号, 只执行不输出

(3) 回车符代表这句结束

(4) 一行写不开, 可用...续行

(5) 注释以%开头

变量赋值的两种常见方式:

->>变量=表达式, 例:

s = 1-1/2+3

输出结果: s = 3.5000

->>表达式 (赋值给默认变量 ans)

2016/81

输出结果: ans = 24.8889

MATLAB 预定义的变量:

(1) ans, 最近表达式计算结果的变量名;

(2) eps, 预设正的极小值, 其值为 2.2204e-16;

(3) pi, 圆周率 π ;

- (4) *inf*, 无穷大, ∞ 值;
- (5) *i* 或 *j*, 虚数单元, $\text{sqrt}(-1)$;
- (6) *NaN*, 非数, $0/0$, ∞/∞ 。

注: 用户只能临时覆盖这些预定义变量的值, **Clear** 或重启 **MATLAB** 可恢复其值。

变量查询方法:

- (1) 工作空间窗口显示;
- (2) **who** 和 **whos** 指令, 列出在 **MATLAB** 工作空间中已经驻留的变量名清单。其中, **whos** 指令在给出变量名的同时, 还给出它们的维数及性质。

清除变量方法:

- >>**clear** 清除所有变量
- >>**clear a** 只清除变量 **a**

变量保存与调用

命令 **quit** 和 **exit** 都可退出 **MATLAB**, 结束工作任务会删除工作空间中的变量, 可采用如下命令来保存工作空间, 以备再次调用这些变量。

保存的指令格式:

- (1) **save**, 将工作空间中的所有变量保存到 **matlab.mat** 文件中。
- (2) **save [文件名] [变量名]**, 将指定的变量保存在指定文件中。如:

save temp x y z

把 **x,y,z** 这三个变量保存在文件 **temp.mat** 中。

在下次加载 **MATLAB** 时, 可以利用 **load** 指令将保存在文件中的变量恢复到工作空间中, 其格式有:

- (1) **load**, 将保存在 **matlab.mat** 中的变量载入到 **MATLAB** 工作空间中。
- (2) **load [文件名] [变量名]**, 从指定的文件中将指定的变量装入 **MATLAB** 工作间。如:

load temp x

只将 **temp.mat** 文件的变量 **x** 载入到 **MATLAB** 工作空间中。

1.2.3 数据操作与显示

数字的计述:

MATLAB 的数值采用习惯的十进制表示, 可以带小数点和负号, 其缺省的数据类型为双精度浮点型 (**double**)。例:

3 -99 0.0013 1.243e-6

表达式按常规相同的优先级从左至右执行运算, 指数运算级别最高, 乘除次之, 加减最低。

加+ 减- 乘* 左除\ 右除/ 幂^

注: 对于标量而言, 左除和右除的作用相同, 但对矩阵运算产生不同的影响。

在缺省的状态下 **MATLAB** 以短格式 **short** 格式显示计算结果。可以用 **MATLAB** 命令窗口中 **format** 指令来改变数字的显示格式。由于 **MATLAB** 以双精度执行所有运算, 显示格式的设置仅影响矩阵的显示, 不影响矩阵的计算与存

储。

如果矩阵的所有元素都是整数，则矩阵以不带小数点的格式显示。如果有一个元素不是整数，则有几种输出格式。默认格式为 **short** 格式，只显示 5 位有效数字，其他的显示格式可显示更多的有效数字，还可用科学表示法。

```
->>format short e 短格式科学表示；  
->>format short g 在 short 和 short e 中自动选择最佳方式；  
->>format long 长格式（15 位有效数字）；  
->>format long e 长格式科学表示；  
->>format long g 在 long 和 long e 中自动选择最佳方式；  
->>format bank 银行格式；  
->>format rat 近似有理数表示；  
->>format hex 十六进制格式；  
->>format compact 紧凑格式。  
.....
```

字符串相关函数：

`class()` 函数，可查看数据类型

`double()` / `abs()` 函数，可将字符串转换成 ASCII 码

`str2num()` / `num2str()` 函数，实现字符串与数字之间的转换。

`eval()` 函数，执行字符串变量中的内容。

1.3 MATLAB 帮助/命令查询

MATLAB 的在线帮助系统相当完备，就查询系统的调用方式而言，可分为两种：

(1) 从 MATLAB 指令窗的 **help** 菜单选项中寻求帮助此与一般 windows 的求助方法一样

(2) 在 MATLAB 指令窗中直接键入帮助指令。

->>help 不带任何参数，显示出 MATLAB 的目录项产生清单信息；

->>help 目录名 (lang/ matfun/ elfun)，显示指定目录中的所有命令及其函数；

->>help 命令名/函数名/符号，显示指定的命令名/函数名/符号的详细信息。

lookfor 指令可以根据用户提供的完整或不完整的关键词，去搜索出一组与之有关的指令。

exist 指令，检查指定名字的变量或函数文件的存在性；

what 指令，按扩展名分类列出在搜索路径中指定目录上的文件名；

which 指令，列出指定名字文件所在的目录。

注：**help** 的工作机理是把指定名字的那个 **M** 文件的第一段注释内容显示出来，以构成自己文件的再线求助。**lookfor** 指令的机制是对 MATLAB 中的每个 **M** 文件注释区的第一行进行扫描，一旦发现包含要查询的字符串就显示出来。用户也可利用此机理建立自己文件的在线帮助。

1.4 MATLAB 的矩阵操作

1.4.1 矩阵的建立

(1) 直接输入法

->>在命令窗口直接输入矩阵元素;
->>按行输入矩阵元素, 同一行元素使用空格或者逗号隔开, 不同行元素使用分号或者回车作为间隔。

(2) M 文件建立矩阵

->>启动编辑器 (输入 `edit` 命令, 或者使用菜单栏按钮);
->>输入待建立矩阵, 输入格式要求和直接输入法类似。

(3) `zeros()` 和 `ones()` 函数

(4) 冒号表达式建立向量

冒号是一个重要的运算符, 语法:

`e1:e2:e3`

`e1` 为初始值, `e2` 为步长 (默认为 1), `e3` 为终止值, 产生行向量;

(5) `linspace` 建立行向量

语法:

`linspace(a,b,n)`

`a` 和 `b` 分别为行向量的第一个和最后一个元素, `n` 为总元素, 省略 `n` 自动产生 100 个元素的行向量

1.4.2 矩阵的简单操作

(1) 索引矩阵

->>可以使用矩阵元素的行标和列标或者矩阵元素的位置索引来索引矩阵元素;

->>行标和列标共同使用的形式;

->>使用序号索引, 涉及到 **MATLAB** 中矩阵的存储形式;

矩阵元素的存放是按照列来排序

`find(A==8)` 或 `find(A,8)` 返回 `A` 矩阵内值为 8 的元素序号; `find(A<8)` 返回所有小于 8 的序号;

->>序号和下标的转换。

`sub2ind(size(A),m,n)` 返回序号

`[m,n]=ind2sub(size(A),3)` 返回下标

(2) 重排矩阵

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

->>`E=reshape(A,9,1)` 重排为 9×1 列向量

->>`I=A[:]` 同上效果

(3) 矩阵的转置

->>`L=K'`

(4) 矩阵的拆分

->>K=A(1,:)获得第一行所有列的内容
->>K=A(1,1:end)效果同上
->>K=A(1,1:2) 获得第一行 1 到 2 列的内容
->>K=A(1,[1 2])效果同上
->>K=A([1 2],[1 2])

(5) 删除矩阵元素

->>赋空值，例如：

M(2)=[]

A(1,:)=[]

(6) 扩展矩阵元素

->>Q= repmat(A,2,1)
将[1 2 3 4]扩展为[1 2 3 4;1 2 3 4]

(7) 压缩矩阵元素

->>R=[1 2 3 3 3 4 4]
->>Q=unique(R)只保留不重复的元素

1.4.3 矩阵的处理基础

MATLAB 运算以矩阵为单位，可以直接实现矩阵的加法、数乘等运算以及矩阵的行列式、矩阵的秩、矩阵的逆以及转置等运算。

计算 A 的行列式；

->>使用 **det(A)**函数

如果 A 的行列式不为 0，那么计算 A 的逆；

->>使用 **inv(A)**函数

向量可以看作是 $1 \times n$ 矩阵或者 $n \times 1$ 矩阵，因此向量的加法和数乘等运算和矩阵式一样的。

求向量内积 $(a,b)=b^H a$;

->>sum(conj(b).*a)

->>a*b', 与上式一致

->>dot(b,a), 与上式一致

线性方程组求解

可以使用矩阵求逆的方法求解线性方程组，在一般线性代数中的解决方法类似。

->>x=inv(A)*b

->>x=A\b, 与上式一致

矩阵的相似化简和分解

求解 A 的 **Jordan** 标准型；

->>使用 `jordan(A)` 函数
求解 A 的特征值;
->>使用 `eig(A)` 函数
范数运算, 包括 1 范数, 2 范数和无穷范数以及 f 范数;
->>使用 `norm` 函数, `norm(A,1)`, `norm(A,2)`, `norm(A,inf)`, `norm(A,'fro')`

函数分析

计算函数矩阵 $A(x)$ 的一阶和二阶导数。在 MATLAB 中函数的计算, 比如求导数, 默认对矩阵的每一个元素求导, 这即为函数矩阵导数的计算方法。

->>`syms x`
->>`A=[sin(x) exp(x) 1;cos(x) x^2+1 log(x)]`
->>`diff(A)`, 一阶导数
->>`diff(A,2)`, 二阶导数

计算矩阵函数, e^A , $\sin A$, $\cos A$ 。在 MATLAB 中使用 `funm` 函数计算矩阵函数, `funm` 为通用矩阵函数计算函数, 其使用方法如下:

`funm(A,@fun)`
->>`funm(A,@exp)`, 求 e^A
->>`expm(A)`, 同上
->>`funm(A,@sin)`, 求 $\sin A$
->>`funm(A,@cos)`, 求 $\cos A$

1.5 MATLAB 程序控制结构

1.5.1 M 文件

分类:

命令文件 (脚本文件, Script File)

函数文件 (Function File)

区别:

命令文件没有输入, 没有返回;

命令文件可以对工作空间的变量操作, 结果返回工作空间, 而函数文件中的变量为局部变量, 函数执行完毕, 变量被清除;

命令文件可以直接运行, 函数文件见需要调用的形式运行 (除了特殊的函数文件外)。

1.5.2 顺序结构

按照代码顺序执行, 即顺序结构, 受控于代码前后位置;

使用于顺序结构中的简单输入和输出 (命令交互):

数据输入 (`input` 函数)

->>`A=input('输入一个参数')`

数据显示 (`disp` 函数)

->>`disp(A)`

程序暂停 (pause 函数)

->>pause(3) 暂停 3 秒

1.5.3 分支结构

分支结构，又称为选择结构，包括 if 分支，switch 分支和 try 分支。

(1) if 分支语句：

if 条件

 语句组 1

else

 语句组 2

end

(2) switch 分支语句

switch 表达式

 case 表达式 1

 语句组 1

 case 表达式 2

 语句组 2

 ...

 case 表达式 n

 语句组 n

 otherwise

 语句组 m

end

(3) try 分支语句

在经典的程序设计语言中未出现，这是一种试探性的分支语句，也就是如果这条语句执行有错误，则不执行，或者执行其他的语句，提高了程序的容错性能。

try

 语句组 1

catch

 语句组 2

end

1.5.4 循环结构

按照给定条件，重复执行某些语句，包括 for 循环和 while 循环。

(1) for 循环语句

for 循环变量=表达式 1:表达式 2:表达式 3 (可以采用矩阵的形式)

 循环体

end

注意循环变量自动增加，在循环体内对循环变量的赋值操作会带来不可预料的程序执行。

(2) while 循环语句

```
while 条件
    循环体 (break 命令可跳出)
end
```

循环可以嵌套，但是在 MATLAB 中，循环的执行效率很低，因此应该尽量避免使用循环，更要避免使用两重以上的循环。

怎样避免使用循环和提高循环效率？

有些可以通过 MATLAB 的矢量化语言，通过矩阵或者向量操作完成；

有些可以通过 MATLAB 提供的一些特殊操作工具箱函数完成；

预分配的使用，会大大增加循环效率。

1.6 MATLAB 编辑器和程序调试

(1) 函数文件

基本结构：

```
function [输出形参]=函数名 ([输入形参])
```

```
[注释说明]
```

```
函数体
```

关于函数文件名

通常和函数名一致，如果不一致，那么忽略函数名，调用时使用函数文件名。

注释部分

注释部分可以提供 help 以及 lookfor 命令的查找功能，例如：

```
function re = add_my(a,b)
%ADD_MY      完成变量 a 和 b 的相加
% 输入：      a 加数 1
%             b 加数 2
% 输出：      re 为两个参数相加之和
%
% $Author: XXX
% $Date: 2017.02.10
% Version: V1.0

re = a + b
```

对命令行进行注释快捷方式 Ctrl+R，解除注释 Ctrl+T

(2) 函数参数的可调识别函数

可以完成传递参数的可调功能，类似于高级语言中的函数重载和多态。

nargin 返回传递给该函数调用的输入参数的数目，仅可在函数体内使用；

nargout 返回该函数调用中指定的输出参数的数目；

varargin 可变长度输入参数列表；c=varargin{1}

varargout 可变长度输出参数列表；

前两者为输入输出参数个数，后面两个可以代表未知的输入输出变量。

`error(nargchk(2,4,nargin))`检测 `nargin` 的范围在 2 到 4，如果超出则报错。

(3) 全局变量

使用 `global` 声明，可以提供不同的 M 文件访问同一个变量。

不同的函数之间访问同一个变量，可以通过返回参数来设置，或者使用全局变量。在 GUI 中，也可以通过其他方式完成（GUI 数据，app 数据等）。

(4) 程序调试

断点和单步；

根据错误提示，初步确定错误内容；

第二部分 MATLAB 软件绘图基础

2.1 二维高层绘图操作

2.1.1 plot 函数

基本用法

`plot(x,y)`

一般要求：`x` 和 `y` 为相同尺度的向量。

特殊用法：

(1) `x` 为向量，`y` 为矩阵，其中一维与 `x` 长度相同，则绘制多条不同色彩的曲线；

(2) `x` 和 `y` 为同维的矩阵，以对应列绘制不同色彩的曲线；

(3) `plot` 有一个参数，若 `x` 为实数向量，则绘制折线图，若 `x` 为复数向量，则以实部和虚部为横坐标绘图；

(4) 多个输入参数；

(5) 曲线选项。

`plot(x,y1,'k:',x,y2,'b-')`

每条曲线的线型和颜色由字符串 '`cs`' 指定，其中 `c` 表示颜色 `s` 表示线型。

常用颜色与线型

颜色符号 c	颜色	线型符号 s	线型
y	黄色	.	点
m	紫色	o	圆圈
c	青色	x	叉号
r	红色	+	加号
g	绿色	*	星号
b	蓝色	-	实线
w	白色	:	点线
k	黑色	--.	点划线
		--	虚线

2.1.2 图形标注

基本标注函数：

title(' ')标题

xlabel(' ')横坐标标题

ylabel(' ')纵坐标标题

text(x1,y1,'')定位文字信息。

legend(' ',) 图例

特殊字符的显示

LaTeX 字符简介

'x^2'显示 x^2 ; 'x_2'显示 x_2

2.1.3 坐标轴控制

(1) 基本控制命令：

axis equal 矩形的坐标轴

axis square 正方形的坐标轴

axis auto 自动坐标轴显示

axis off 关闭坐标轴

axis on 打开坐标轴

axis([xmin xmax ymin ymax]) 设置坐标轴的显示范围

xlim([xmin xmax]) 设置 x 坐标的显示范围

ylim([ymin ymax]) 设置 y 坐标的显示范围

(2) 图形保持命令

hold on

hold off

(3) 窗口分割

subplot(2,2,1)将图形绘制成 2×2 的小格，这是第一格。

2.2 二维底层绘图修饰

2.2.1 对象和句柄

MATLAB 把构成图形的各个基本要素成为图形对象，产生每一个图形对象时，MATLAB 会自动分配一个唯一的值用于表示这个对象，称为句柄。

对象之间的基本关系

计算机屏幕->图形窗口->(用户菜单，用户控件，坐标轴)

坐标轴->曲线，曲面，文字，图形，光源，区域，方框

2.2.2 基本底层绘图函数

(1) line 对象和 line 函数

示例：h = line([-pi:0.01:pi],sin([-pi:0.01:pi]))

产生 line 对象，h 为 line 对象的句柄

line 对象的修饰：

Color 属性；

LineWidth 属性；

LineStyle 属性；

Marker 属性；

MarkerSize 属性；

plot 绘制的图形也是 line 对象，可以通过 line 对象的修饰方法后期修饰。

```
h = line('XDate',[-pi:0.1:pi],'YDate',sin([-pi:0.1:pi]),...
```

```
    'LineWidth',1,'LineStyle',':', 'Color','r')
```

```
set(h,'LineWidth',2,'Marker','p','MarkerSize','3')修改对象
```

(2) 底层标注

text 对象和 text 函数

常用修饰：

Color 属性；

String 属性；修改标注的内容

FontSize 属性；

Rotation 属性；

(3) 底层坐标轴的控制

axes 对象和 axes 函数

修饰：

Box 属性；

GridLineStyle 属性；网格线型

Position 属性；

Units 属性；可以设置单位为归一化 normalized，便于图形缩放

XLabel、YLabel 和 ZLabel 属性；

Xlim、Ylim 和 Zlim 属性；

第三部分 MATLAB 特殊用法简介

3.1 Notebook 使用方法

3.1.1 安装和启动

必须装有任意一款 Word 软件，在 MATLAB 中输入安装命令即可：

```
notebook -setup
```

注意，老版本的 MATLAB 在安装 Notebook 时会提示用户选择本机中所安装的 Word 版本。

启动 Notebook

从 MATLAB 中启动 Notebook，直接输入 notebook（新建）或者 notebook

文件名；

从 Word 中启动 Notebook，新建 M-book 模板文档，如果 MATLAB 未运行，则 MATLAB 跟随 M-book 自动启动；

3.1.2 代码的运行和结果

在 Word 加载项里多出了 Notebook 的功能选项。

在 Notebook 中，需要运行代码部分要设置为“输入单元”，使用绿色加粗 Courier New 字体显示；

定义输入单元，可以使用 Notebook 菜单或者组合键“Alt+D”；

定义输入单元之后，选择 Notebook 菜单中的“Evaluate Cell”或者组合键“Ctrl+Enter”执行单元，输出结果直接在相应输入单元后显示，使用蓝色 Courier New 字体显示。注意，可以不提前设置输入单元而直接运行。

自动初始化单元可以在打开 Notebook 时自动执行，这样可以快速恢复所需要的工作空间。

定义方式：选择需要定义的单元，选择 Notebook 菜单中的“Define AutoInit Cell”。

定义为自动初始化的单元使用蓝色加粗 Courier New 字体显示。

输入单元组定义

每一个输入单元之间是相互独立的，只能一行一行的运行，在执行 for 语句时就会出现错误。

定义方式：选择需要定义的单元，选择 Notebook 菜单中的“Group Cells”。

输出的格式设置，选择菜单“Notebook Option”，可弹出相应的选项：数值类型、松散 or 紧密和输出图形格式（默认嵌入）。

3.2 符号运算操作

（1）符号常量和符号变量

语法：

```
->>a = sym('a')
```

```
->>syms a
```

```
->>c = sym('3') 符号常量 c=3
```

（2）符号表达式

语法：

```
->>f = '3*x + 6' 字符串也能参与符号运算
```

```
->>f = sym('3*x + 6')
```

```
->>syms x; f = 3*x + 6
```

（3）符号四则运算

+ - * / ^

(4) 符号表达式化简

simplify 使用多项式规则进行化简

simple 使用很多规则进行化简

(5) 符号表达式与数值的转化

sym

eval 将符号变量转化为数值

(6) 符号表达式的因式分解和展开

factor 因式分解

expand 展开为这个变量级数和的形式

collect 合并同类项

(7) 符号矩阵

构建方式和数值矩阵一致

转置和行列式

transpose

determ

其他使用在数值矩阵中的函数也可以直接用于符号矩阵

(8) 符号计算

subs(f1,n)求解

limit 极限

diff 微分

int 积分

symsum 级数求和

taylor 泰勒级数展开

solve 代数方程求解

dsolve 常微分方程求解