MATLAB

0 历史与发展

0.1 全称

Matrix Laboratory

0.2 用途

- 数值和符号计算
- 绘图
- 工具箱（toolbox）:功能性/学科性

---

1 安装、启动与设置

1.1 操作界面

工作区/命令行窗口

1.2 设置

1.2.1 搜寻过程

1. 是否变量
2. 是否函数
3. 是否当前目录下的M文件
4. 是否其他路径的M文件

1.2.2 常用函数

- `path`：MATLAB包含的所有路径
- `help`：查看函数的功能（+文件名）
- `cd`（current directory）：查看当前目录/进入目录
- `userpath`：查看/修改默认路径
- `savepath`：保存默认路径的更改
- `pathtool`：打开路径设置窗口，手动设置
- `clc`(clear command window)：清屏

---

2 数据操作和语法

2.1 变量和语句

2.1.1 变量定义和命名

- 字母开头
- 命名字符：字母、数字、下划线，区分大小写
- 不声明，不定义，拿来即用

2.1.2 语句

- 回车键表示一句结束

- 末尾有分号不输出执行结果，否则输出执行结果
- 注释以%开头
- 用...续行

2.1.3 变量赋值

- 变量=表达式
- 表达式：赋值时赋给默认结果变量(ans)

2.2 变量管理

2.2.1 工作空间中驻留的变量查看

- 工作空间窗口显示
- 命令方式:
  - `who`
  - `whos`

2.2.2 清除变量

- `clear` （+ 变量名）

2.2.3 保存数据

- save函数的使用
  - 语法：`save [文件名][变量名][-append][-ascii]`
  - `append`：添加保存内容

2.2.4 数据输出格式控制

- format函数：
  - format 格式符
  - 不影响计算和存储，只控制输出
  - 默认存储格式：short

2.3 数据类型

2.3.1 数值型

- 分类：双精度、单精度、带符号和无符号整型
- 建立方法：变量=表达式
- 类型转换：如`a = uint8(a)`,即将a转化为uint8（无符号8位整型）类型。
- 数据类型查看：`class(a)`

2.3.2 字符串

- 概念：使用单引号括起来的字符序列
- 常见操作：
  - 获取字符的ascii码值：double或abs函数，如：`double('a')`
  - ascii码转化为字符输出：char函数,如：`char(65)`
  - 执行字符串内容：eval函数，如：`eval('t=1')`
  - 字符串和数值的互换：`str2num`和`num2str`

2.3.3 结构体

- 建立：结构体.成员名=表达式

- 常用函数和操作：
  - 判断变量是否结构体：`isstruct`函数(1/0)
  - 输出结构体的成员名：`fieldnames`函数
  - 判断名称是否结构体的成员名：`isfield`函数
  - 删除某结构体的成员：`rmfield`函数
  - 输出某结构体的成员值：`getfield`函数

2.3.4 单元

- 建立：使用大括号括起，可包含矩阵等多种数据类型

2.3.5 多维矩阵

2.3.6 稀疏矩阵

---

3 矩阵的建立与操作

3.1 矩阵的建立

3.1.1 直接输入法

- 在命令行直接输入矩阵元素
- 按行输入
  - 同行元素：用空格或逗号隔开
  - 不同行元素：用分号或回车隔开

3.1.2 M文件建立法

- 启动编辑器（输入edit命令或使用按钮）
- 输入待建立矩阵，方法与直接输入类似
- 运行M文件：在命令行输入文件名或按钮操作

3.1.3 特殊矩阵的建立

- 零矩阵的建立：`zeros(3,4)`:3*4的矩阵
- 全一矩阵的建立：`ones(5,4)`:5*4的矩阵

3.1.4 冒号表达式法

- 语法：`e1:e2:e3`
  - e1为初始值，e2为步长，e3为终值
  - 产生行向量

3.1.5 linspace建立法

- 语法：`linspace(e1,e2,e3)`
  - e1和e2为行向量的第一个和最后一个元素，e3为元素总数
  - 省略e3时自动产生100个元素的行向量

3.2 矩阵的简单操作

3.2.1 矩阵元素查找

- 可以使用行标、列标或位置索引矩阵元素
- 行标和列标共同使用的形式：find函数（一个/两个输出参数：位置/行列表）
- 使用序号索引，涉及MATLAB矩阵的存储形式

- 序号和下标的转换：sub2ind函数和ind2sub函数
- 例：`A(2,3)`，`A(5)`，`ind=find（A == 5）`，`sub2ind(size(A),行标,列标)`，`ind2sub(size(A),位置)`

3.2.2 矩阵重排（维数）

- 元素数不变，改变矩阵形式
  - reshape函数，如：`reshape(A,9,1)`
  - 冒号表达式，如：`a(:)`,将a作为列向量返回(执行效率高)

3.2.3 矩阵转置

- 语法：单引号表示（共轭转置）

3.2.4 矩阵拆分

- 语法：`A(1,:)`，运用冒号表达式
- 前一位为行，后一位为列，如：1:2指第一行到第二行
- 或者可以写作：[1,2]
- `A([1,2],[1,2])`的索引顺序：1,1;1,2;2,1;2,2

3.2.5 删除矩阵的元素

- 赋空值
- 可以使用冒号表达式
- 如：[]

3.2.6 其他操作

1. 矩阵扩增：repmat函数，语法：`repmat(矩阵名，行数，列数)`
2. 矩阵压缩：unique函数，删除重复的值

---

4 矩阵处理

4.1 其他特殊矩阵的建立

4.1.1 零矩阵和幺矩阵

zeros和ones函数

4.1.2 单位矩阵

`eye(3,4)`:4*3的单位矩阵 `eye(10)`:10*10的单位矩阵

4.1.3 随机矩阵

- `rand`：0和1之间均匀分布的随机矩阵，如：`rand(10,1)`为1*10的行向量
  - 用(10-0)调整想要的区间
  - 从a到b之间的随机函数：`a+(b-a)\*rand`
- `randn`：标准正态分布(均值为0，方差为1)的随机矩阵
  - 用`a + sqrt(b)*randn`调节想要的均值和方差
  - 用`mean`和`std`函数获取变量的均值和方差

4.1.4 魔方矩阵

- 语法：`magic`函数

4.1.5 Hilbert矩阵和Toeplitz矩阵

- 使用的函数: `hilb` 函数和 `toeplitz` 函数

4.2 矩阵和向量的运算

4.2.1 矩阵的加和

- 语法：`A + B`

4.2.2 矩阵的数乘

- 语法：`3 * B`

4.2.3 矩阵的行列式

`det` 函数

4.2.4 矩阵的秩、逆和转置

- 逆矩阵：`inv` 函数(如果矩阵的行列式不为0)
- 秩
- 转置：单引号

4.2.5 向量的内积运算

- 语法：`A' * B`
- `dot` 函数

4.3 线性方程组的求解

设定参数向量a和常数向量b，求: `a\b` 或 `inv(a)*b`

4.4 矩阵的相似化简和分解

4.4.1 求解Jordan标准型

`ordan` 函数

4.4.2 求矩阵的特征值

`eig` 函数或 `[W Z]=jordan(A)`

4.4.3 向量和矩阵的范数

norm函数（包括1范数，2范数，无穷范数和f范数） 如：`norm(A,1)` `norm（A,2）` `norm(A,inf)` `norm(A,'fro')`

4.5 矩阵分析

4.5.1 函数矩阵的导数求解

- 函数矩阵的建立： 如：`syms x` `A = [sin(x) exp(x) 1+ 2 * log(x)]`
- `diff(A,a)`:求a阶导数

4.5.2 矩阵函数

如：`A=[0 1 2]`

- 矩阵的指数: `expm` 函数
- 矩阵的正余弦值：通用矩阵函数— `funm(A,@func)`

5 M文件及其程序控制结构

5.1 M文件

5.1.1 分类

- 命令文件：脚本文件、Script File等
- 函数文件（Function File）
- 两类文件的区别：
    - 命令文件没有输入和返回
    - 命令文件可以对工作空间的变量操作，函数文件中的变量是局部变量
    - 命令文件可以直接运行，函数文件需要调用运行（除了特殊的函数文件之外）
- 类M文件：不常用

5.1.2 建立和打开

5.1.2.1 M文件的建立

- 快捷键或菜单栏
- 命令行输入edit

5.1.2.2 M文件的打开

- 在命令行输入文件名

5.2 脚本文件的程序控制结构

5.2.1 顺序结构

- 按代码位置执行
- 使用的输入输出函数
    - 数据输入：input函数
        - 语法：`input("提示语")`
    - 数据显示：`disp`函数（显示，去除变量值的无意义的空格）
    - 程序暂停：`pause(秒数)`

5.2.2 分支结构

- 选择结构
- if分支/switch分支/try分支

5.2.2.1 if分支

- 语法：

```
if 条件
   语句组;
end
```

- 分段函数的实现：可以由更好的办法
- 多分支if语句的语法：

```
if 条件
    语句组;
           elseif 条件
               语句组;
           end
```

- 大小写字母间的转化：ascii码

5.2.2.2 switch分支

- 语法：

```
switch 表达式
     case 表达式：
            语句组；
     otherwise 表达式：
            语句组；
end
```

- 不需要使用break
- case 后的语句可以是标量
- 浮点数：.03=0.03

5.2.2.3 try分支

- 语法：

```
try （查错）
       语句组1
catch（若错了则执行该组语句）
       语句组2
end
```

功能：试探性的语句，若这条语句有错误，则不执行，或执行其他的语句

5.2.3 循环结构

- 按给定条件重复执行某些语句
- for/while循环

5.2.3.1 for循环

- 语法：

```
for 循环变量 = 表达式1;表达式2;表达式3
          循环体;
end
```

注意：循环变量自动增加，循环内不要对其赋值

- 不要使用i,g进行循环（MATLAB的保留字）

5.2.3.2 while循环

- 语法：

```
while 循环体
      语句组;
```

- `isempty`函数:判断是否空字符

5.2.3.3 嵌套循环

- 循环执行效率低，不建议使用，避免使用二重以上的循环
- 原因：解释型语言
- 常使用矩阵/向量/特殊工具箱/预分配代替嵌套循环提升效率

5.3 函数文件

5.3.1 基本结构

- `function[输出形参]=函数名([输入形参])`
- 如果是独立的函数文件，则这一句前面不可以有语句
  - 如果该函数在其他的函数文件中定义，则该语句之前可以有该函数定义的内容
  - 如果函数文件没有输入和输出形参，则其可以独立运行
- [注释说明]
- 可以通过`help`或`lookfor`命令查看函数文件的注释说明
  - 快捷键
    - Ctrl+R 加注释
    - Ctrl+T 取消注释
- 函数体

5.3.2 函数文件名

- 函数文件的名称通常与函数名一致
- 若二者不一致，在调用时忽略函数名，使用函数文件名

5.3.3 函数参数可调标识变量

- `nargin`
- `nargout`
- `varargin`
- `varargout` **注：**

1. 上述参数可以完成传递函数的可调功能，类似于高级语言中的重载和多态；
2. 前两者为输入输出参数的个数，后两者可以代表位置的输入输出变量。 **例1** 输入元素个数可调的加法程序 MATLAB002.m **问题：如果提供了超过输入元素最大总数的参数怎么办?**

例2 输入元素可能未知的加法程序 MATLAB003.m **问题：如果输入参数超过输入元素最大总数，如何显 示错误信息?**

使用`nargchk`函数显示错误信息

5.3.4 全局变量

- 使用`global`声明可以使不同的M文件访问同一个变量（也可以使用返回参数）
- 在GUI中可以通过其他方式完成

5.4 程序调试

- 单步调试
- 断点调试

- 调试程序时，命令行双小于号之前有一个大写的K

6 二维绘图操作

6.1 二维高层绘图操作

6.1.1 基本函数：plot函数

- 基本用法：`plot(x,y);`
- 要求：x和y为相同长度的向量（？，特殊用法可以不满足该条件）
- 特殊用法：
  - x为向量，y为矩阵，其中一维与x长度相同，绘制多条不同色彩的曲线；（颜色怎么确定？见6.2）
  - x和y为同维的矩阵，以对应列绘制不同色彩的曲线；
  - 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纵坐标绘图；
  - 多个输入参数；
  - 曲线选项
    - 曲线颜色选项：参数在变量之后
      - `‘r’`：红色
      - `‘k’`：黑色
      - `‘b’`：蓝色
      - `‘g’`：绿色
      - `‘y’`：黄色
      - `‘m’`：品红
    - 数据点的形状和散点图or曲线图调整
      - `‘*’`:用星号标注
      - `‘.’`：用点标注
      - `‘p’`：用五角星标注
      - `'<'`和`'>'`：用三角号标注
      - `'-'`：用平滑曲线连接
      - `':'`：用虚线连接
      - `‘--’`：用粗虚线连接
      - `‘-.’`：用点划线连接
    - 上述参数可以混用

6.1.2 辅助操作

6.1.2.1 图形标注

- 坐标轴标签：`xlabel`和`ylabel`函数
- 曲线标题：`title`函数
- 附注：`text`函数（可定位，见6.2）
- 图例：`legend`函数（可定位，见6.2）
- 其他标注：下划线代表下标，三角号代表上标（LaTex语法）

6.1.2.2 坐标轴控制

- `axis equal`
- `axis square`：显示正方形坐标轴

- `axis auto`
- `axis off`
- `axis on`
- `axis([xmin xmax ymin ymax])`
- `xlim([xmin xmax])`
- `ylim([ymin ymax])`

6.1.2.3 图形保持

`hold on`命令：开启图形保持

`hold off` 命令：关闭图形保持

6.1.2.4 窗口分割

`subplot`函数+定位序号

6.2 二维底层绘图修饰

6.2.1 对象和句柄

- 图形对象：构成图形的各个基本要素
- 句柄：产生每一个图形对象时，MATLAB自动分配给这个对象的值。
- 对象间的关系
  - 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，图像，光源，区域，方框)

6.2.2 基本底层绘图函数：line对象和line函数

- line对象和line函数
  - 使用语法：`h = line(x,y)`
  - 使用line函数后会生成一个line对象，h为对象的句柄
  - line函数的属性
    - Color属性
    - LineStyle属性
    - LineWidth属性
  - Marker属性
    - MarkerSize属性
  - 对句柄的操作可进行对对象的修饰
- line函数扩展语法

```
e = line('XData',x的范围,'YData',Y的范围,'LineWidth',线宽,'LineStyle',线型,'Color',线的颜色);
```

- set函数：进一步设定对象的属性

```
set(句柄,'MarkerSize',12);
```

6.2.3 底层标注：text对象和text函数

- 使用语法：`h = text('Char',y);`
- 常用属性：

    - Color属性
    - String属性：标注的内容
    - FontSize属性
    - Rotation属性
- 获取句柄的属性值：`get`函数

6.2.4 底层坐标轴控制：axes对象和axes函数

- 使用语法：`h = axes('Char',y);`
- 常用属性：
    - Box属性
    - GridLineStyle属性
    - Position属性
    - Units属性
    - XLabel、YLabel和ZLabel属性
    - Xlim、Ylim和Zlim属性
- 获取当前对象的属性：`get(gcf)`，可参阅属性的值

7 MATLAB Notebook

MATLAB Notebook只适用于32位的Microsoft WOrd,不适用于64位的Microsoft Word。

替代品：Live Editor

8 MATLAB 符号运算

8.1 符号计算基础

8.1.1 符号常量和符号变量

1. 符号常量的产生：sym

   如：`c=sym(‘3’);`

2. 符号变量的产生：sym函数

   如：`syms b;` `a = sym('a');`

8.1.2 符号表达式

- 语法：
    - `f = ‘3*x + 6’;`
    - `syms x;f = 3 * x + 6;`
    - `f = sym('3*x + 6');`
- 三种生成符号表达式的区别：
    - 第一种：char类型（满足符号运算规则）
    - 第二和第三种：sym类型

8.1.3 符号的四则运算

- 第一版：`symadd` `symsub` `symmul` `symdiv` `sympow`(不可使用)
- 新版MATLAB支持：+-*/^

8.1.4 符号表达式的化简

- `simplify`函数
- `simple`函数(新版MATLAB已不可用)

8.1.5 符号表达式和数值的转换

- 符号表达式转化为数值：`eval`或`numeric`（不可使用）函数
- 数值转化为符号表达式：`sym`函数

8.1.6 符号表达式的因式分解和展开

- 因式分解：`factor`函数
- 展开：`expand`函数（定义在前的函数降幂排列，定义在后的函数升幂排列）
- 合并同类项：`collect`函数

8.1.7 符号矩阵

- 构建方式和数值矩阵一致
- 转置和行列式
  - 转置：`transpose`函数（T转置）
- 行列式：`determ`函数

  注：其他在数值矩阵中可以使用的函数同样适用于符号矩阵

8.2 符号函数

- 求值计算:`subs`函数

  如：`subs(f1,3);`
- 符号极限：`limit`函数

  如：`limit(y,0);`(有时结果的意义很不明确)
- 符号积分：`int`函数

  如：`int(f1,2);`
- 符号微分：`diff`函数

  如：`diff(f1,1,2);`

8.3 符号级数

- 符号级数的求和：`symsum`函数

  如：`symsum(f,x,1,inf);`
- 泰勒级数的展开：`taylor`函数

  如：`taylor(f,x,在哪一个位置处展开);`

8.4 符号方程求解

- 符号代数方程和方程组：`solve`函数

  如：[x y]= solve(x+y == 2, x ^0.5+y^0.5 = 1)
- 符号常微分方程：`dsolve`函数
  - 变量微分的表示：如`D3y`表示y的三阶导数