

# 上机辅导讲义

## MatLab 简介

### 1、Matlab 产生发展的历史

MATLAB 语言的首创者是 Cleve Moler: 1980 年前后, 当时的新墨西哥大学计算机系主任 Moler 教授在讲授线性代数课程时, 发现了用其他高级语言编程极为不便, 便构思并开发了 MATLAB (MATrix LABoratory, 即矩阵实验室), 这一软件利用了当时数值线性代数领域最高水平的 EISPACK 和 LINPACK 两大软件包中可靠的子程序, 用 Fortran 语言编写了集命令翻译、科学计算于一身的一套交互式软件系统。

所谓交互式语言, 是指人们给出一条命令, 立即就可以得出该命令的结果。该语言无需像 C 和 Fortran 语言那样, 首先要求使用者去编写源程序, 然后对之进行编译、连接, 最终形成可执行文件。这无疑会给使用者带来了极大的方便。早期的 MATLAB 是用 Fortran 语言编写的, 只能作矩阵运算; 绘图也只能用极其原始的方法, 即用星号描点的形式画图; 内部函数也只提供了几十个。但即使其当时的功能十分简单, 当它作为免费软件出现以来, 还是吸引了大批的使用者。

Matlab 中我们主要使用到的三个空间: 命令空间 (输入命令并执行之), 工作空间, 编辑器空间。

### 2、MATLAB 特色举例

考虑两个矩阵 A 和 B 的乘积问题, 在 C 语言中要实现两个矩阵的乘积并不仅仅是一组双重循环的问题。双重循环当然是矩阵乘积所必需的, 除此之外要考虑的问题很多。例如: A 和 B 有一个是复数矩阵怎么考虑; 其中一个是复数矩阵时怎么考虑; 全部是实系数矩阵时又怎么管理; 这样就要在一个程序中有 4 个分支, 分别考虑这 4 种情况。然后还得判断这两个矩阵是否可乘。而考虑两个矩阵是否可乘也并不仅仅是判断 A 的列数是否等于 B 的行数这么简单。其中一个若为标量, 则它们可以无条件地相乘。其中有标量时又得考虑实数与复数的问题等。所以说, 没有几十分钟的时间, 用 C 语言并不可能编写出考虑各种情况的子程序。有了 MATLAB 这样的工具, A 和 B 矩阵的乘积用  $A*B$  这样简单的算式就能表示了。

【例 1-1】矩阵生成与运算。考虑金庸作品中经常提及的一个“数学问题”, 该问题用半数学语言描述就是: 如何生成一个  $3 \times 3$  矩阵, 并将自然数 1, 2, ..., 9 分别置成这 9 个矩阵元素, 才能使得每一行、每一列、且主、反对角线上元素相加都等于一个相同的数。

这样的矩阵称为“魔方矩阵”。用 MATLAB 的 magic() 函数, 我们可以由下面的命令立即生成这样的矩阵:

```
>> A=magic(3)
A =
    8    1    6
    3    5    7
    4    9    2
```

还可以由  $B=magic(10)$  一次生成  $10 \times 10$  的魔方矩阵。

## 一、脚本文件和函数文件

- ◆ 将 MATLAB 命令窗口中一行一行输入的命令汇集在一个文件中,这样一来要输入这一组命令时,只需输入这一文件名。这种文件称为 MATLAB 的脚本文件,其文件扩展名为.m。
- ◆ 另一类非常重要的 M 文件是 MATLAB 函数文件,它由五部分构成
  - ◆ 函数定义行
  - ◆ H1 行
  - ◆ 函数帮助文本
  - ◆ 函数体
  - ◆ 注释

脚本文件(stat1.m):

```
% 求阵列 x 的平均值和标准差
%
[m,n]=size(x);
if m==1 (这里==是等于,=是个赋值符号)
    m=n;
end
s1=sum(x); s2=sum(x.^2);
mean1=s1/m;
stdev=sqrt(s2/m-mean1.^2);
```

函数文件(stat2.m) :

```
function [mean1,stdev]=stat2(x) %函数定义行
% 求阵列 x 的平均值和标准差 %H1 行
% 调用格式为 |
% [mean,stdev]=stat2(x) | 函数帮助文本
[m,n]=size(x); %以下为函数体,其中可包含注释
if m==1
    m=n;
end
s1=sum(x); s2=sum(x.^2);
mean1=s1/m;
stdev=sqrt(s2/m-mean1.^2);
```

## 二、函数工作空间

对以上的脚本文件 stat1.m 和函数文件 stat2.m, 分别执行后, 发现执行 stat1.m 时, 产生了所有由 stat1.m 产生的变量, 而执行 stat2.m 则只产生函数返回的结果变量, 其它由函数文件产生的变量只存在于特定的函数工作空间中。

这样, 利用 MATLAB 函数设计程序时, 只需考虑由函数返回的结果变量, 方便设计人员; 但同时也给调试程序带来了不便, 幸好 MATLAB 给程序设计提供了专用的调试器。

## 三、子函数

在一个函数文件中, 可包含多个函数, 其中第一个函数名应与文件名同名, 其余函数统称为这一函数的子函数, 它们只能在这一函数内部才能使用。

**流程控制语句**可改变程序执行的流程, MATLAB 有四类流程控制语句:

- ◆ if, else, elseif, end 条件转移语句;

- ◆ switch , case , otherwise , end           情况切换语句；
- ◆ for , end                               指定次重复的循环语句；
- ◆ while , end                           不定次重复的循环语句。

这些语句给 MATLAB 程序设计带来了极大的方便，也给设计带来了灵活性。

### (1) 条件语句

常用的格式为：

```

if a<0
    disp('a 为负数')
elseif a>0
    disp('a 为正数')
else
    disp('a 为零')
end

```

可根据 a 的不同取值，进行不同的处理。

If 语句还可以嵌套使用。

### (2) 情况切换语句

switch 语句可根据表达式的不同取值执行不同的语句，这相当于多条 if 语句的嵌套使用。例如：

```

switch var1
    case -1
        disp('Var1 is negative one.')
    case 0
        disp('Var1 is zero.')
    case 1
        disp('Var1 is positive one.')
    otherwise
        disp('Var1 is other value.')
end

```

在 case 语句中可出现多个值，例如

```

switch var2
case {-2,-1}
    disp('Var2 is negative one or two.')
case 0
    disp('Var2 is zero.')
case {1,2,3}
    disp('Var2 is positive one, two or three.')
otherwise
    disp('Var2 is other value.')
end

```

Switch 语句还可以出现字符串，例如：

```

switch lower(method)
case {'linear','bilinear'}

```

```

    disp('Method is linear.')
case{'cubic'}
    disp('Method is cubic.')
case{'nearest'}
    disp('Method is nearest.')
otherwise
    disp('Unknown method.')
end

```

### (3) 指定次重复循环语句

for 语句用于完成指定次的重复循环语句，这是大家所熟知的形式。

例如计算 20! :

```

r=1;
for k=1:20
    r=r*k;
end
disp(r)

```

for 语句还可以利用数组(即阵列)任意指定循环变量的值，例如

```

varx=[-5 2 8 1 7];
for x=varx
    disp(x.^2-5*x);
end

```

For 循环可以嵌套使用。例如采用二重循环可方便地进行数据的排序:

```

x=fix(100*rand(1,10)); disp(x)
n=length(x);
for i=1:n
    for j=n:-1:i+1
        if x(j)>x(j-1)
            y=x(j);x(j)=x(j-1);x(j-1)=y;
        end
    end
end
disp(x)

```

For 循环中可采用 break 语句来终止循环,如上例中可利用 break 提高执行效率:

```

x=fix(100*rand(1,10)); disp(x)
n=length(x);
for i=1:n
    flag=-1;
    for j=n:-1:i+1
        if x(j)>x(j-1)
            y=x(j);x(j)=x(j-1);x(j-1)=y;flag=0;
        end
    end
    end
    if flag,break,end
end

```

计算结果为:

```

50
-6
24
-4
14

```

```
disp(x),disp(['循环次数为',num2str(i)])
```

#### 四、不定次重复循环语句

while 语句可执行不定次重复的循环，它与 for 循环不同，在每次循环前要判别其循环条件，当条件为真时或非零值时，重复循环；否则结束循环。因此，在 while 循环中，可通过改变循环变量来改变循环次数；但在 for 循环中，虽然也可以改变其循环变量，但其循环次数不受其影响

例如：

```
r=1; k=1;
while r<1e50
    r=r*k; k=k+1;
end
k=k-1; r=r./k; k=k-1;
disp([' The ',num2str(k),'! is ',num2str(r)
var=[1 2 3 4 5 6 -1 7 8 0];
a=[]; k=1;
while var(k)
    if var(k)==-1, break, end
    a=[a var(k).^2]; k=k+1;
end
disp(a)
```

```
1    4    9   16   25   36
```

注意：break 还可用于 for 循环！

MATLAB 语言与其它语言一样，程序设计的好坏，直接影响程序的执行效率。我们结合多年利用 MATLAB 进行编程的经验，给大家介绍两种行之有效的设计技术。

- ⊗ 循环的向量化；
- ⊗ 阵列预分配。

#### 一、循环的向量化

实际应用中，有些循环可直接转换成向量，这时可大大提高程序的执行效率。因此编程时，尽量采用向量或矩阵操作，而应避免循环。

为测试程序执行的快慢，可采用 tic 和 toc 函数。

例如：tic

```
i=0;
for t=0:.01:100
    i=i+1;y(i)=sin(t);
end
toc
```

```
tic
t=0:.01:100;
y=sin(t);
toc
```

执行时间分别为

7.58

0.01

## 程序调试技术

MATLAB 的调试器 (debugger) 可帮助你尽快找到编程的错误。

MATLAB 的程序调试主要用来纠正两类错误：

- ◆ 格式错误 (语法错误), 这在程序运行时就会直接给出提示；
- ◆ 运行错误, 体现在执行结果不对, 执行时出现异常。

Debugger 主要用来诊断第二类错误。

在包含函数调用的 MATLAB 程序运行时, 当发生运行错误时, 不会显示出错信息, 而在执行结束或出错时, 我们只能看到基本工作空间 (即主程序执行空间) 中的变量, 各个函数工作空间已关闭, 因此无法检测各个局部变量。

为此, 应采用调试技术来查找问题。

为查找运行错误, 可采用下列技术：

- ◆ 在可能发生错误的 M 文件中, 删去某些语句行末的分号, 使显示其运行中间结果, 从中可发现一些问题；
- ◆ 在 M 文件的适当位置上加上 keyboard 命令, 使在执行时在此暂停, 从而检查局部工作空间中变量的内容, 从中找到出错的线索, 利用 return 命令可恢复程序的执行；
- ◆ 注释掉 M 函数文件的函数定义行, 使函数文件转变成脚本文件, 这样在程序运行出错时, 可查看 M 文件产生的中间变量；注意, 局部变量之间应避免冲突；

为查找运行错误, 可采用下列技术： (续)

- ◆ 使用 MATLAB 提供的 Debugger, Debugger 为我们提供了下列功能：
  - ☆ 设置 / 清除断点；
  - ☆ 单步执行；
  - ☆ 继续执行；
  - ☆ 查看和修改各个函数空间的内容。

为说明 MATLAB 的调试技术, 我们通过一简单示例加以说明：

第一步: 分段设置断点

一开始并不能肯定问题所在, 我们总是按照执行顺序分段执行, 以期期望能够找到出错的程序段。为此在 variance.m 中设置断点, 并执行程序：

第二步：清除断点

第三步：调试嵌套函数

四、单步执行

五、修改并重新执行

找到了问题所在, 修改源程序并重新执行, 如仍有问题, 继续上述过程进行调试, 直到程序执行正确。

但要注意, 对某一特定值, 程序得到了正确的结果, 并不是说程序完全正确。

程序调试技术总结：

1. 程序编写出来，并不意味着设计任务的结束；
2. 编写程序是容易的，但调试程序是一个复杂而困难的；
3. 调试程序是一个需要多次反复的任务，应确保每个支路都能正常工作，要做到这一点并不容易；
4. 程序调试技术应在不断实践中积累经验。

Matlab 使用光标健上下左右箭头来调用前面的命令。上箭头在提示符处调用出上一次的命令。下箭头调用下一条命令。在任何时刻都可通过左右箭头在命令行内移动。这样可以用来编辑命令。

**介绍一些用得着的命令：**

matlab 的命令很多很难记，为了帮助用户找到命令，matlab 通过其广泛的在线帮助功能提供帮助。这些功能主要有三种形式：help 命令（在命令窗口中输入 help ...，回车即可），或者使用 helpwin 命令显示帮助窗口，该帮助窗口提供所有命令的帮助。lookfor 命令(提供了一种通过一般的关键词找到 matlab 命令和帮助标题的方式)；还可交互使用 help 菜单条。

clear 无条件删除 matlab 工作空间中的变量。

ans(answer 的缩写)用于结果的缺省变量名。

axis- 用于设置影响当前坐标轴对象的相关值。所谓当前坐标轴对象，即最后创建的坐标轴或最后用鼠标点击选中的坐标轴。

plot-图形绘制命令，如：plot ( Y ),Y 是一个矩阵，则以矩阵的每个行列为线条绘制图形，plot(x,y)，x,y 为矢量，则绘制矢量 y 相对于矢量 x 的图形。等等，还可以有其他参数。

subplot(x,y,z) ,该命令用于创建一个图形框。x 代表行数，y 代表列数，z 为第 z 个图形。例如 subplot(2,2,1)。

title( ' ... ' )给输出图像加标题。

abs() 按所操作的变量可以计算复数的幅值或者实数的绝对值。

exp 表示指数

clc 清除命令窗

%起到文本注释的作用，其后的内容将不被执行。

**显示图像：**

image (C),函数将矩阵 C 作为图像显示，其元素可以是双精度型的或者是 unit8 型的数据。

imagesc 调整数据并显示成图像。它可以对其数据进行伸缩调整以充分利用全部的颜色图。

imshow 也用于显示图像,根据其参数不同 ,可以有不同的显示功能。例如 imshow(I,[low,high])

可以显示给定显示图像的灰度范围。Imshow(filename)可以显示 filename 所指定的图像

文件中的图像。等等，功能很大，可自行查书或利用前面讲得 help 命令。在 matlab

注意：给文件起名字时，不能只是使用数字。

例如：1.m

在执行时，命令窗口中会出现：ans=1,而不是执行了该文件。

在编写程序时要求我们一定不可直接调用 matlab 工具箱里的现成的工具，要自己通过编程来实现所要求的功能。