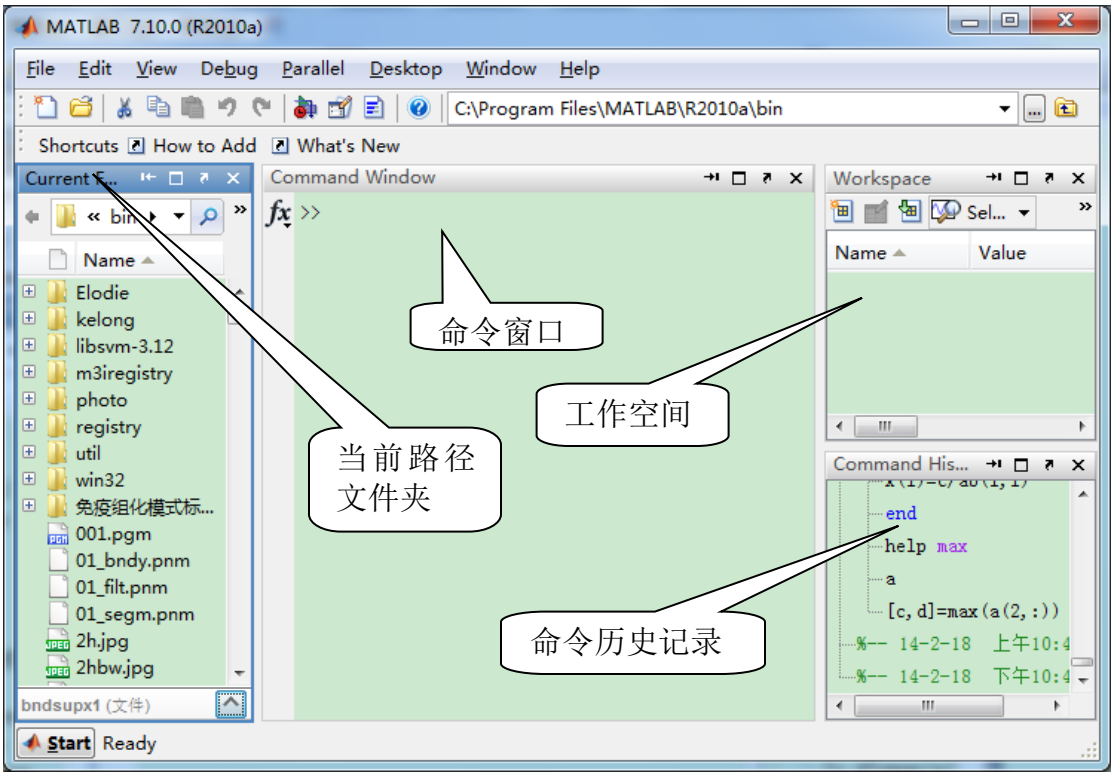
实验指导 1

Matlab 入门

1. Matlab 的主要功能

1.1 Matlab 简介

Matlab 界面上有很多可以移动、改变大小和关闭的活动区域，默认的桌面布局包含四个主要的区域。中部面积最大的是命令窗口，在这里输入命令并返回执行结果；右上角的是工作空间，可以显示在命令窗口中生成的对象的各种信息；右下角的是命令的历史记录，在这里可以很方便地重复执行以前输入过的命令；左侧显示了 Matlab 当前路径下的文件。



在命令窗口中有一个由两个大于号组成的命令提示符“>>”，在其右侧有闪烁的输入光标。在命令提示符后面输入：

```
>>a=1+2
```

按下回车键，Matlab 会立刻进行运算并且在命令窗口返回：

```
a =
     3
```

也可以不用把结果赋给一个变量而象下面这样直接进行计算：

```
>>1+2
```

执行后 Matlab 返回

```
ans =
     3
```

ans 是 Matlab 中默认的结果变量。 每次 Matlab 进行运算后，结果都要储存在指定的变量中。如果只输入表达式，却不指定把表达式的运算结果储存在哪个变量里面，Matlab 会

自动地将结果储存在 ans 变量中。直到下一次不带指定存储变量的运算结束前，ans 中所储存的值不变。因此可以在下一次运算中用 ans 调用上一次运算的结果。

变量命名规则

Matlab 变量名的第一个字符必须是英文字母，最多可包含 63 个字符（英文、数字和下划线）。变量名中不得包含空格、标点、运算符，但可以包含下划线（中连接符-已经作为减号使用，不可以包含）。如变量名 my_var_201 是合法的，而 my，var201 由于逗号的分隔，表示的就不是一个变量名。Matlab 的变量名、函数名区分字母大小写。如变量 myvar 和 MyVar 表示两个不同的变量。

注释与标点

命令行中“%”符号后的所有文字为注释，计算机不会执行。符号“…”表示语句的余下部分将出现在下一行，但它不能出现在变量名或运算符之间。

在命令窗口回车表示执行，所以在这里通常是一行代码一行代码地执行的。如果一行中有多个命令，这些命令之间要用逗号或者分号隔开。每一行的末尾可以加分号或者不加，分号除了分隔同一行的多个命令外还有屏蔽输出的作用。下列包含两条命令：

```
>>x = 3;
>>y = x + 5
```

由于第一行代码的结尾有分号，所以赋值的结果并不显示，Matlab 只返回第二行命令的结果：

```
y =
8
```

1.2 Matlab 的数据及数值分析

数据分析

Matlab 的基本数据单元是矩阵。在作数据分析时，如果输入的是向量，运算是对整个向量进行的；若输入的是数组（矩阵），则运算按列进行。

利用 Matlab 可进行数据的基本统计计算，如下列的函数。运算时如果调用格式中有 dim，则指明运算按指定维数进行。更详细的使用帮助可以通过指令 doc 或 help 获得具体函数的详细信息，语法是：doc <函数名> 或 help <函数名>。

max(x,dim)：求最大元素。

min(x,dim)：求最小元素。

median(x,dim)：求中位值。

mean(x,dim)：求平均值。

std(x,flag)：求标准差，flag 指明标准差的不同计算方式。

prod(x,dim)：求积。

sum(x,dim)：求和。

cumsum(x,dim)：求累计和。

cumprod(x,dim)：求累计积。

cov(x)：求协方差阵。

cov(x,y)：求相关阵。

corrcoef(x)：求两随机变量的协方差。

corrcoef(x,y)：求两随机变量的相关系数。

sort(x)：以升序排列元素。

微积分的分析

Matlab 除了能够进行数值运算外，还可以进行各种符号运算。符号计算可以用推理解析的方式进行，避免数值计算带来的截断误差。与数值计算不同的是，符号计算需要先定义符号变量。下面只简单介绍符号运算在微积分计算中的应用，更多的使用方法和函数可在符号数学工具箱或 Maple 中找到。

例1.　用 limit 函数求函数 sin(x)/x 在 x->0 时的极限。

```
>> syms x;
>> a=limit(sin(x)/x)
a =
1
```

例2.　用 fminbnd 函数求单变量函数$f(x)=2x^3-6x^2-18x+7$在区间(-2,4)的极小值。

```
>> f=@(x)2*x.^3-6*x.^2-18*x+7;
>> [x,f]=fminbnd(f,-2,4)
x =
     3.0000
f =
   -47.0000
```

这个例子中使用了匿名函数快速创建表达式定义的简单函数，其语法格式为

fhandle=@(arglist)expr

匿名函数可在命令窗口或者程序文件中使用，expr 是用来定义函数的表达式，相当于函数体；arglist 是以逗号分隔的输入变量名列表；@是用来创建函数句柄的运算符；fhandle 是某个变量名，用以保存该匿名函数的函数句柄，通过该变量名调用对应的匿名函数。在一段程序中，后定义的匿名函数可以调用已定义过的匿名函数。

例3.　求函数的微积分

```
>>syms x y t
>> d1=diff(sin(x^2)*y^2,2)                %计算∂²y/∂x² sinx²
d1 =
2*y^2*cos(x^2) - 4*x^2*y^2*sin(x^2)
>> d2=diff(d1,y)                          %计算∂/∂y (∂²y/∂x² sinx²)
d2 =
4*y*cos(x^2) - 8*x^2*y*sin(x^2)
>> syms x z t alpha
>> R1=int(-2*x/(1+x^2)^2)                 %计算∫ −2x/(1+x²)² dx
R1 =
1/(x^2 + 1)
>> R2=int(x/(1+z^2),z)                    %计算∫ x/(1+z²) dz
R2 =
```

```
x*atan(z)
>> R3=int(x*log(1+x),0,1)                    %计算∫_0^1 xln(1+x)dx
R3 =
1/4
```

例4.　求函数$f=a/(x-10)$的二阶泰勒级数展开。

```
>> syms a x
>> f=a/(x-10);
>> y1=taylor(f,x,'order',3)                  %求 f 在 x=0 处的二阶泰勒展开
y1 =
- a/10 - (a*x)/100 - (a*x^2)/1000
>> y2=taylor(f,x,'Expansionpoint',4,'order',3)    %求 f 在 x=4 处的二阶泰勒展开
y2 =
- a/6 - (a*(x - 4))/36 - (a*(x - 4)^2)/216
```

非线性方程的数值解

1） fsolve：最小二乘法，调用格式为 x=fsolve(fun,x0)

2） fzero：零点法，调用格式为 x=fzero(fun,x0)

例5.　求方程$x-e^{-x}=0$的解。

```
>> fc=@(x)x-exp(-x)
>> x1=fsolve(fc,0)
x1 =
    0.5671
>> z=fzero(fc,2)
z =
    0.5671
```

另外，还有 solve 函数、solver 函数和 dsolve 函数分别用于求代数方程的符号解、常微分方程的数值解和符号解。具体使用方法可参考帮助文档。

1.3 Matlab 矩阵的建立及基本操作

向量

向量是 matlab 中的一个基本单位，向量的每一个元素的运算包括向量的创建、向量的加减运算和向量的乘除运算。

向量的创建：在 matlab 的命令窗口键入以下字符

```
>> a = [1 2 3 4 5 6 9 8 7]
a =
     1   2   3   4   5   6   9   8   7
```

希望得到元素从 0 到 20，步距为 2 的一个向量，只需键入以下命令即可

```
>> t = [0:2:20]
t =
     0    2   4   6   8   10   12   14   16   18   20
```

向量的加减运算：设 a,b 为同维向量，则 c=a+b 或 c=a-b 得到两个向量相加减的结果。而向量与常数的相加减则为每个元素加减这个常数。

例如：

```
b = a + 2
```

得到

```
b =
     3   4   5   6   7   8   11   10   9
     c = a + b
c =
     4   6   8   10   12   14   20   18   16
```

向量的乘除运算:

向量的乘法运算

行向量和列向量可进行乘法运算，例如

a=[1:5], b=[2:6]；则 a*b' = 70

点积运算的运算符为 .*， 其意义为两个向量的对应元素进行乘法运算，例如

a=[1 2], b=[3 4] 则 c=a.*b=[3 8]

^ 为向量的乘方运算，例如

c=a.^2=[1 4]

向量的除法运算

向量没有除法，“向量 AB/向量 CD”是没有意义的。

多项式的创建和表示：多项式在 Matlab 中以向量的形式保存，只需要自高向低依次保存各幂次项的系数即可，要特别注意的是，如果多项式中缺少某一幂次的项，在向量中必须保持此项的系数 0 的存在，例如：$s^4+3s^3-15s^2-9$，如果想将其输入到 **Matlab** 中，则按下列方式输入向量

```
x = [1   3   -15   0   9]
 x =
      1   3   -15   0   9
```

可利用函数“**polyval**”计算多项式的值。例如，多项式x^4+1在 x=2 的值为

```
z = polyval([1 0 0 0 1],2)
z =
     17
```

若要求多项式 $s^4+3s^3-15s^2-2s+9$ 的根则可用函数“**roots**”实现。

```
roots([1 3 -15 -2 9])
ans =    -5.5745
          2.5836
          -0.7951
          0.7860
```

矩阵

输入矩阵时每一行元素有分号或者回车键分隔。例如：

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

矩阵转置运算：

```
C = B'
  C =
       1     5     9
       2     6    10
       3     7    11
       4     8    12
```

矩阵乘法：

```
D = B * C
  D =
      30    70   110
      70   174   278
     110   278   446
```

矩阵点乘：当两矩阵维数相同时，运算符.*的结果是两矩阵的对应元素相乘。如：

```
E = [1 2;3 4]; F = [2 3;4 5];
G = E .* F
    G =
         2     6
```

矩阵的乘方：矩阵为方阵时，可以进行矩阵的乘方运算，运算符为^。如：

```
E^3
ans =
        37    54
        81   118
```

若仅是元素进行乘方运算，可用运算符.^。如：

```
E.^3
ans =
         1     8
        27    64
        12    20
```

矩阵的逆：矩阵逆利用函数 **inv** 计算，此时，要求矩阵方阵且可逆。 如：

```
X = inv(E)
  X =
   -2.0000    1.0000
    1.5000   -0.5000
```

矩阵元素的赋值与运算：Matlab 允许用户对矩阵的单个元素进行赋值和操作，Matlab 此时命令方式为

X(i,j)=变量名

如：X(1,1)=1

```
  X =
    1.0000    1.0000
    1.5000   -0.5000
```

矩阵的特征值及特征多项式:特征值利用函数 **eig** 来计算，**特征多项式**利用函数 **poly** 来计算特征多项式的系数，此时，多项式系数以降幂形式排列。如：

```
eig(E)
```

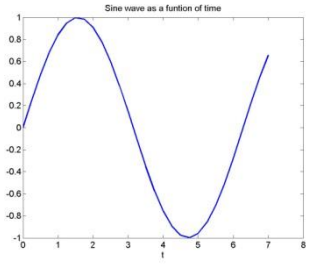
```
ans =
              -0.3723
               5.3723
p = poly(E)
p =
         1.0000      -5.0000      -2.0000
```

1.4 Matlab 的绘图功能

基本绘图函数 plot，调用格式是 plot(X,Y)，作用是以 X、Y 的对应元素为坐标绘制二维图形，其中 X、Y 的维数要匹配，更多用法请参考帮助文档。

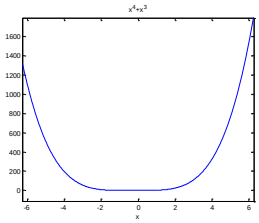
例如绘制一个作为时间函数的正弦波的图像。首先产生一个时间向量，然后计算每一时刻的正弦值。

```
t=0:0.25:7;
y = sin(t);
plot(t,y)
```



ezplot 函数用于符号函数的绘图，调用格式为 ezplot(f,[a,b])，绘出函数 f 在[a,b]上的图形。

```
x=0:0.01:3;
f=@(x)x.^4+x.^3;
ezplot(f)
```



2. Matlab 的程序编制

2.1 关系及逻辑运算

关系运算符主要用来比较数与数、矩阵与矩阵之间的大小，并返回真(1)或假(0)。基本的关系运算符主要有 6 种：>、<、>=、<=、==（等于）、~=（不等于）。

逻辑运算符有 4 种：与（&）、或（|）、非（~）、异或（xor）。变量中非零数的逻辑值为“真”，0 的逻辑值为“假”，逻辑运算结果以“1”表示“真”，以“0”表示“假”。

2.2 M 函数文件

Matlab 中有很多内嵌的库函数，如 sin(x), sum(A)等。但使用中用户往往需要编制自己的函数，以实现计算中的参数传递和函数的反复调用。函数扩展名为**.m**，必须以关键字 **function** 开头。建立函数文件的方法如下：

```
function    [y1,y2,…]=ff(x1,x2,…)
```

其中，ff 是函数名；x1,x2 是输入变量；y1,y2 是输出变量。

例如：编写一个函数实现多项式的加法运算。

```
function    [poly]=polyadd(poly1,poly2)
% 定义 polyadd.m 函数
%   polyadd(poly1,poly2)  adds two polynominals  possibly  of uneven length
% 比较两个多项式的最高次幂，在低次幂的多项式前面补上对应个数的 0，
% 使两个多项式的长度相等，然后再相加
if length(poly1)<length(poly2)
```

```
    short=poly1;
    long=poly2;
else
    short=poly2;
    long=poly1;
end
mz=length(long)-length(short);
if mz>0
    poly=[zeros(1,mz),short]+long;        %两个多项式幂次相等的项的系数相加
else
    poly=long+short;                      %两个多项式幂次相等的项的系数相加
end
```

这个例子中包含了典型的 M 函数的各个部分：函数定义行、H1 行（H1 行是帮助文本的第一行，紧跟在定义行后）、帮助文档、函数主体和注释。

函数编辑完成后，将文件保存为 polyadd.m，默认的文件名为函数名，一般不要去改动它，以免造成调用麻烦。程序中要求输入两个多项式，在命令行中输入两个多项式：

```
>>x =[1    2];
>>y =[1    4    8];
```

输入文件名调用此函数：

```
>>z=polyadd(x,y)
```

运行程序，输出如下：

```
z =
          1    5    10
```

这表示多项式$x+2$和多项式x^2+4x+8相加得到的新的多项式为$x^2+5x+10$.在这个程序中，出现了一组常用的程序控制语句 if-else-end，这是条件分支结构，在下面 2.4 小节会有详细用法介绍。

注意：

1. 输入变量用()括起来，输出变量用[]括起来，当输出变量只有一个时也可以不加括号。
2. 函数名和文件名一般相同，函数名开头必须用字母，并且区分大小写。
3. 程序必须以 function 开始，第二行以后可加入注释行或运算语句。
4. M 函数文件可以调用其他一般的 M 文件，M 函数文件可以反复调用自己。
5. 如果函数简单，可以直接使用匿名函数定义。
6. 只有注释部分可以使用全角符号，其他任何位置都只能使用半角符号。

2.3 M 文件

单击【File】菜单下的【New】子菜单中的【M-file】命令，或者“Ctrl+N”，进入文本编辑窗口，输入程序，完成后单击“保存”按钮，在对话框中输入文件名，即创建了一个 M 文件。M 文件中可以是一些代码段，执行文件相当于执行这些代码段；也可以是一个自定义的函数，执行文件就是调用这个函数。

运行 M 文件有以下几种方法：在命令窗口中输入文件名并按<Enter>键；单击【File】菜单下的【Open】命令，在弹出的【Open】对话框中单击*.m（文件名），打开该文件编辑窗口，再单击【Debug】菜单下的【Run 文件名.m】命令。要注意的是，当 M 文件是一个需要输入参数的函数时，应该使用第一种方法，在命令窗口输入文件名的同时在相应位置输入输入参数的值，才能正确运行程序。

2.4 程序控制语句

下面所有的控制语句都以 end 结束。

if 条件语句

格式一：

```
if  (条件式)
    条件块语句组
end
```

格式二：

```
If  (条件式)
    条件块语句组 1
else
    条件块语句组 2
end
```

格式三：

```
If  (条件式 1)
    条件块语句组 1
elseif  (条件式 2)
    条件块语句组 2
end
```

注意，格式三的条件式可以不止 2 个，如果太多的话，可考虑使用下面的 switch 分支语句。

for 循环语句

格式：

```
for  循环变量=表达式 1: 表达式 2: 表达式 3
    循环语句组
end
```

注意，for 循环结构中循环次数是给定的，除非用其他语句将循环提前结束（如 break）；表达式是一个向量；循环语句组中可使用“；”防止中间结果输出；for-end 结构可嵌套，但运算速度会受到影响。

while 循环语句

格式：

```
while  (条件式)
    循环体条件组
end
```

表达式一般由逻辑运算和关系运算组成。若表达式的值非 0，继续循环；若表达式的值为 0，中止循环。

switch 分支选择结构

这种语句是多分支选择结构，虽然有时候可以用 if 语句的多层嵌套来完成，但 switch 语句显得更加简单明了。

格式：

```
switch  表达式
    case  常量表达式 1
          语句块 1
    case  常量表达式 2
```

```
            语句块 2
    …
    case  常量表达式 n
            语句块 n
    otherwise
            语句块 n+1
end
```

注意，switch 后面的表达式可以为任意类型；当表达式的值与 case 后面的常量表达式的值相等时，就执行 case 后面的语句块；case 后面的常量表达式可以有多个，也可以是不同类型；每次只执行一个语句块，执行完一个语句块就退出 switch 语句。

例：

```
switch  var
    case{‘abc’,’12’}
        disp(‘第一种情况’);
    case{1,2,4,’www’}
        disp(‘第二种情况’);
    case{6,7,8,’Matlab’}
        disp(‘第三种情况’);
    otherwise
        disp(‘意外情况’);
end
```

注意，case 后面是{ }，而不是( )，运行结果为：

var=4, 显示“第二种情况”;

var=abc, 显示“第一种情况”;

var=10, 显示“意外情况”。

2.5 编程要点

为了尽量加快 Matlab 程序的运行速度，编程时应注意如下几点：

1） 计算时尽量避免使用循环，而使用向量或矩阵；

2） 如果要使用循环，在循环语句前也要尽量对向量、矩阵或数组预先使用 ones（生成全 1 矩阵）或者 zeros（生成全 0 矩阵）函数进行内存分配。

3） 尽量使用 Matlab 的内部函数或工具箱函数。绝大多数常见的数学计算都可以在 Matlab 中找到相应的函数命令。

4） 运算时可使用 tic（启动秒表）和 toc（停止秒表）来测试程序运行所话费的时间。

3. Matlab 的学习技巧

3.1 学会使用 help

Matlab 中 Help 的常用方法有

1)在命令窗口中直接输入“help”获得本地机器上的基本帮助信息；

2)输入“help funname”获得某个具体函数的帮助信息；

3)输入“help toolboxname”获得某个工具箱的相关信息。

3.2 参考网络资源

3.3 要敢于尝试