数学建模讲义

数学软件简介

Mathematical Model

MATLAB简介

Mathematica——侧重于数学方面，其它方面应用不是很广泛

Lindo, Lingo——主要求解优化问题(建议与**Matlab**结合使用)

MATLAB的发展史

- MATLAB名字由MATrix和 LABoratory 两词的前三个字母组合而成。那是20世纪七十年代，时任美国新墨西哥大学计算机科学系主任的Cleve Moler出于减轻学生编程负担的动机，为学生设计了一组调用LINPACK和EISPACK矩阵软件工具包库程序的的“通俗易用”的接口，此即用FORTRAN编写的萌芽状态的MATLAB。
- 1984年由Little、Moler、Steve Bangert合作成立MathWorks公司，并把MATLAB正式推向市场。从这时起，MATLAB的内核采用C语言编写，而且除原有的数值计算能力外，还新增了数据图视功能。
- 1997年仲春，MATLAB5.0版问世，紧接着是5.1、5.2，以及和1999年春的5.3版，现在最高版本有6.5。现今的MATLAB拥有更丰富的数据类型和结构、更友善的面向对象、更加快速精良的图形可视、更广博的数学和数据分析资源、更多的应用开发工具。

MATLAB语言的主要特点

（1）具有丰富的数学功能

- 包括矩阵各种运算。如：正交变换、三角分解、特征值、常见的特殊矩阵等。
- 包括各种特殊函数。如：贝塞尔函数、勒让德函数、伽码函数、贝塔函数、椭圆函数等。
- 包括各种数学运算功能. 如：数值微分, 数值积分, 插值, 求极值,方程求根, FFT ，常微分方程的数值解, 符号运算, 极限问题、积分问题等。

（2）具有很好的图视系统

- 可方便地画出二维和三维图形。
- 高级图形处理。如：色彩控制、句柄图形、动画等。
- 图形用户界面GUI制作工具，可以制作用户菜单和控件。使用者可以根据自己的需求编写出满意的图形界面。

（3）可以直接处理声音和图象文件。

- 声言文件。如： **WAV文件（例：wavread，sound等）.**
- 图象文件。如： **bmp 、gif 、 pcx 、tif 、jpeg等文件。**

（4）具有若干功能非常强大的应用工具箱。

- 如：**OPTIMAL、SIMULINK、DSP、 SIGNAL等30多种.**

（5）使用方便，具有很好的扩张功能。

- 使用MATLAB语言编写的程序可以直接运行，无需编译。
- 可以**M**文件转变为独立于平台的EXE可执行文件。
- MATLAB的应用接口程序**API**是MATLAB提供的十分重要的组件，由一系列接口指令组成。用户就可在**FORTRAN**或**C**中， 把MATLAB当作计算引擎使用 。

（6）具有很好的帮助功能

- 提供十分详细的帮助文件（**PDF** 、HTML 、demo文件）。
- 联机查询指令：**help**指令（例：**help elfun，help exp，help simulink**），**lookfor**关键词（例： **lookfor fourier** ）。

MATLAB简介

1 MATLAB的计算功能

补充

（1） MATLAB的数据类型

- 现有四种基本数据类型：双精度数组、字符串数组、元胞数组、构架数组。(矩阵是特殊的数组)
- 元胞数组（Cell Array）如同银行里的保险箱库一样。
  - 该数组的基本组分是元胞（Cell），以下标来区分。
  - 元胞可以存放任何类型、任何大小的数组。
  - 同一个元胞数组中各元胞的内容可以不同。
- 构架数组（Structure Array）也能存放各类数据。
  - 该数组的基本组分是构架（ Structure ），以下标来区分。
  - 构架必须在划分“域”后才能使用。
  - 数据不能存放于构架，只能存放在域中。
  - 构架的域可以存放任何类型、任何大小的数组。
  - 不同构架的同名域中存放的内容可不同。

一、变量与函数

1、变量

MATLAB中变量的命名规则

(1)变量名必须是不含空格的单个词；

(2)变量名区分大小写；

(3) 变量名必须以字母打头，之后可以是任意字母、数字或下划线，变量名中不允许使用标点符号.

(4) 不区分变量与常量.

特殊变量表

特殊变量	取　　值
ans	用于结果的缺省变量名
pi	圆周率
eps	计算机的最小数，当和 **1** 相加就产生一个比 **1** 大的数
inf	无穷大，如 **1/0**
NaN	不定量，如 **0/0**
i，j	**i=j=**$\sqrt{-1}$
nargin	所用函数的输入变量数目
nargout	所用函数的输出变量数目
realmin	最小可用正实数
realmax	最大可用正实数

2、数学运算符号及标点符号

+	加法运算	适用于两个数或两个矩阵或矩阵与向量的运算；不同符号用法稍有差异。
—	减法运算	
*	乘法运算	
.*	点乘运算	
/	除法运算	
./	点除运算	
^	乘幂运算	
.^	点乘幂运算	
\	反斜杠表示左除.	

（1）MATLAB的每条命令后，若为逗号或无标点符号， 则显示命令的结果；若命令后为分号，则禁止显示结果.

（2）“%”后面所有文字为注释.

（3）“...”表示续行.

3. 常用数学函数

函　数	名　称	函　数	名　称
sin(x)	正弦函数	asin(x)	反正弦函数
cos(x)	余弦函数	acos(x)	反余弦函数
tan(x)	正切函数	atan(x)	反正切函数
abs(x)	绝对值	max(x)	最大值
min(x)	最小值	sum(x)	元素的总和
sqrt(x)	开平方	exp(x)	以e为底的指数
log(x)	自然对数	$\log_{10}(x)$	以10为底的对数
floor(x)	取整	log2(x)	以2为底的对数
sign(x)	符号函数	fix(x)	取整

注意：这些函数可以直接以向量或矩阵作为参数，输出为对应向量或矩阵。

4、M文件

M文件有两种形式：脚本文件(Script File)和函数文件(Function File)。这两种文件的扩展名，均为“.m”——Matlab的可执行文件。

4.1 M脚本文件

- 对于一些比较简单的问题，在指令窗中直接输入指令计算。
- 对于复杂计算，采用脚本文件（Script file）最为合适。
- MATLAB只是按文件所写的指令执行。
- M脚本文件的特点是：
  - 脚本文件的构成比较简单，只是一串按用户意图排列而成的(包括控制流向指令在内的)MATLAB指令集合。

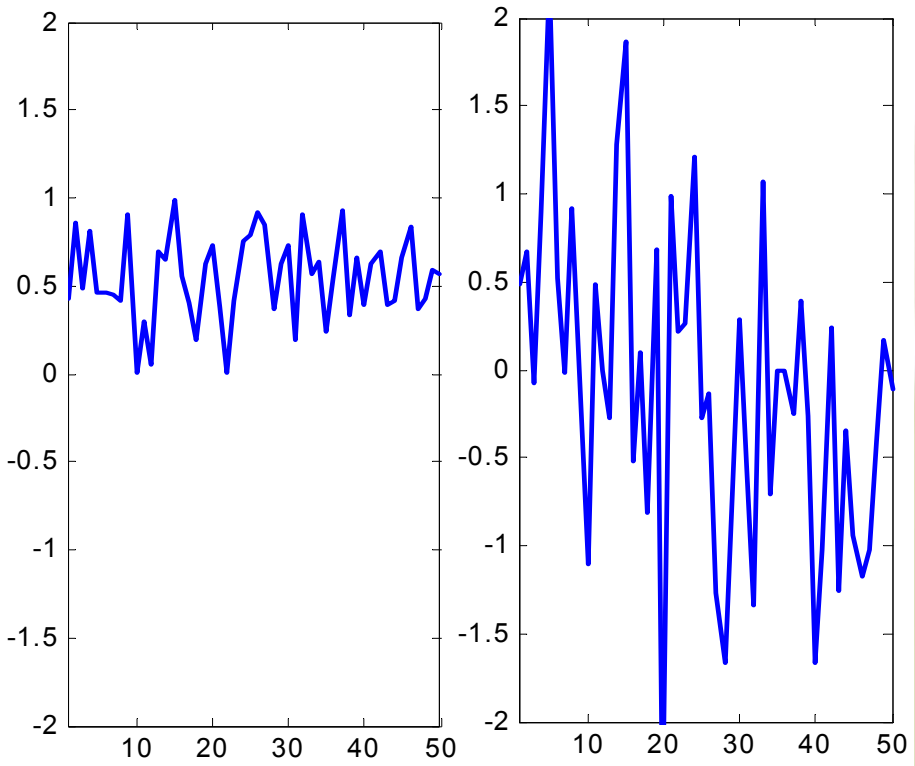
M文件脚本文件建立方法：

1. 在Matlab中,点: File->New->M-file, 生成编辑窗口
2. 在编辑窗口中输入复杂的计算命令集合；
3. 点：File->Save，存盘，文件名任意——易记忆的.

例、随机产生50个随机数，并用图形显示结果(比较均匀分布与正态分布)

```
a=rand(1,50);
subplot(1,2,1);
plot(a);

b=randn(1,50);
subplot(1,2,2);
plot(b);
```



4.2 M函数文件

- 与脚本文件不同，函数文件犹如一个“黑箱”，把一些数据送进并经加工处理，再把结果送出来。
- **MATLAB提供的函数指令大部分都是由函数文件定义的。**
- **M函数文件的特点是：**
  - 从形式上看，与脚本文件不同，函数文件的第一行总是以 **“function”**引导的“函数申明行”。一般结构如下

```
function   因变量名=函数名（自变量名）
%注释行
函数体.....
return
function  因变量名=函数名（自变量名）←子函数
%(只在上函数中使用)
%注释内容
函数体.....
return
.....
```

4.3 M函数文件的一般结构

- 典型 M函数文件的结构如下：
  - 函数申明行：位于函数文件的首行，以关键字 **function** 开头，函数名以及函数的输入输出宗量都在这一行被定义。
  - 第一注释行：紧随函数申明行之后以%开头第一注释行。该行供**lookfor**关键词查询和 **help**在线帮助使用 。
  - 在线帮助文本区 ：第一注释行及其之后的连续以%开头的所有注释行构成整个在线帮助文本。
  - 编写和修改记录：与在线帮助文本区相隔一个“空”行，也以%开头，标志编写及修改该M文件的作者和日期等 。
  - 函数体：为清晰起见，它与前面的注释以“空”行相隔。

M文件建立方法：

1. 在Matlab中点:File->New->M-file
2. 在编辑窗口中输入程序内容
3. 点：File->Save存盘，文件名必须函数名一致。

例：定义函数 $f(x_1,x_2)=100(x_2-x_1^2)^2+(1-x_1)^2$

1.建立M文件：fun.m

```
function   f=fun(x)
f=100*(x(2)-x(1)^2)^2+(1-x(1))^2;
return
```

2. 可以直接使用函数fun.m

例如：计算 f(1,2), 只需在Matlab命令窗口键入命令：

x=[1 2]；fun(x)

4.4 函数调用和参数传递

- 在MATLAB中，调用函数的常用形式是：

[输出参数1,输出参数2,…] = 函数名(输入参数1,输入参数2, …)

- 函数调用可以嵌套，一个函数可以调用别的函数，甚至调用它自己 (递归调用)。

- MATLAB允许使用比 “标称数目 ”较少的输入输出变量，实现对函数的调用 。

二. 矩阵(含数组):

1. 矩阵的建立与表示法:

在命令窗口中输入：**A=[1,2,3;4,5,6;7,8,9]**

可以得到：

```
A =
     1     2     3
     4     5     6
     7     8     9
```

若要显示整行或整列，则可以用(:)冒号来表示。冒(:)代表矩阵中行(ROWS)或列(COLUMNS)的全部。例如执行命令：**A(:,2)**,就会显示第2列的全部，结果为：

```
ans =  2
       5
       8
```

2.矩阵的四则运算符号：

加 “+” 减 “—” 乘 “*” 乘方“^”

除 “/” 共轭转置 “'” 非共轭转置 “.'”

特殊符号“.”——“.*”, “./”, “.^” 左除“\”

例如：b=[1+2i;3+4i]

```
b =
   1.0000 + 2.0000i
   3.0000 + 4.0000i
b'
ans =
   1.0000 - 2.0000i   3.0000 - 4.0000i
b.'
ans =
   1.0000 + 2.0000i   3.0000 + 4.0000i
```

运算非常灵活

3. 其他特殊矩阵的生成方法：

1) eye (m,n)或eye (m) 产生m*n 或 m*m的单位矩阵。例如：

eye (3,4)与eye (3)分别产生如下矩阵：

1	0	0	0
0	1	0	0
0	0	1	0

1	0	0
0	1	0
0	0	1

2) zeros (m,n) 或 zeros (m) 产生m*n 或m*m 的零矩阵。例如： zeros (3,4) 与zeros (3) 分别产生如下矩阵：

0	0	0	0
0	0	0	0
0	0	0	0

0	0	0
0	0	0
0	0	0

3） ones (m,n) 或ones (m) 产生m*n或m*m的全部元素为1的矩阵。例如：ones (3,4)与ones(3)分别产生如下矩阵：

1	1	1	1
1	1	1	1
1	1	1	1

1	1	1
1	1	1
1	1	1

4.常用矩阵函数：

1） d=eig (A) ----返回矩阵A的特征值所组成的列向量；

[v,d]=eig (A)----返回分别由矩阵A的特征向量和特征值（以其为主对角线元素，其余元素为零）的两个矩阵。

例如执行命令：**d=eig (A)** 结果为：

```
d =
    16.1168
    -1.1168
    -0.0000
```

例如执行命令：**[v,d]=eig (A)** 结果为：

```
v =                                    d =
   0.2320    0.7858    0.4082          16.1168         0         0
   0.5253    0.0868   -0.8165           0        -1.1168         0
   0.8187   -0.6123    0.4082           0         0        -0.0000
```

其中**v (:,i)** 为**d (i,i)**所对应的特征向量。

2） **det (A)** 计算行列式**A**的值。例如**det (A)** 结果为：

ans = 0

3） **inv (A)** 求矩阵**A**的逆。例如:**inv (A)** 结果为：

Warning: Matrix is close to singular or badly scaled.Results may be inaccurate. RCOND = 2.055969e-018.

ans = 1.0e+016 *

-0.4504	0.9007	-0.4504
0.9007	-1.8014	0.9007
-0.4504	0.9007	-0.4504

4） orth (A) 返回对应于A的正交化矩阵。例如：orth (A) 结果为：

ans =

0.2148　0.8872

0.5206　0.2496

0.8263　-0.3879

5） poly (A) 若A为一矩阵，则返回A的特征多项式。例如：poly (A) 结果为：

ans =

1.0000　-15.0000　-18.0000　-0.0000

若A为一向量，则返回以A的元素为根的特征多项式。例如：

r=[1,2,3]; p= poly (r) 结果为：

p =

1　-6　11　-6

6) rank (A) 计算矩阵A的秩。例如：r=rank (A) 结果为：r = 2

5. 矩阵分解：

1） [q,r]=qr (A) 将矩阵A做正交化分解，使得A=q*r。q为单位矩阵 (unitary matrix),其范数（norm）为1。r为对角化的上三角矩阵。例如：

```
[q,r]=qr(A)
q =
   -0.1231    0.9045    0.4082
   -0.4924    0.3015   -0.8165
   -0.8616   -0.3015    0.4082
r =
   -8.1240   -9.6011  -11.0782
         0    0.9045    1.8091
         0         0   -0.0000
norm(q)
ans =      1.0000
```

2) [L,U]=lu (A) 将矩阵A做对角线分解，使得A=L*U，L为下三角矩阵（lower triangular matrix），U为上三角矩阵(upper triangular matrix)。

例如：

```
[L,U]=lu(A)
L =
    0.1429    1.0000         0
    0.5714    0.5000    1.0000
    1.0000         0         0
U =
    7.0000    8.0000    9.0000
         0    0.8571    1.7143
         0         0    0.0000
```

6. 关系运算符

MATLAB有用于比较矩阵的六个关系运算符，也可以对矩阵与一个标量进行比较，即矩阵中的每个元素与标量进行比较。关系运算符如下：

<	小于
<=	小于等于
>	大于
>=	大于等于
==	等于
~=	不等于

关系运算符比较对应的元素，产生一个仅包含1和0的具有相同维数的矩阵。其元素是：

1	比较结果是真
0	比较结果是假

在一个表达式中，算术运算符优先级最高，其次是关系运算符，最低级别是逻辑运算符。圆括号可以改变其顺序。

可用于同维矩阵比较及矩阵与单个元素比较。

7. 逻辑运算符

在MATLAB中有四种逻辑运算符：

&	与
|	或
~	非
xor	异或

逻辑运算符的运算优先级最低。在一个表达式中，关系运算符和算术运算符的运算级别
要高于逻辑运算符。

xor和or之间的差别在于：表达式中至少有一个是真，那么or是真； xor是表达式中有一个是真但不能两者均为真时才为真。运算符&和|比较两个相同维数的矩阵，如同前一节一样，它也能使一个标量与一个矩阵进行比较。逻辑运算符是按元素比较的。零元素表示逻辑值假，任何其他值的元素表示逻辑值真。其结果是一个包含1和0的矩阵。

8.逻辑函数

在MATLAB中有几个逻辑函数。在以下定义的函数中，假设A是一个$m \times n$的矩阵， x是一个向量。

在一些计算中，很重要的一点是要在给定的矩阵中以一定的特征定位。例如，在部分选主元的高斯消去法中，必须在工作列中寻找最大的项。MATLAB命令find可以用于这种情况。

find(x) 返回一个x中包含非零元素的下标的向量。如果所有的元素都是零，那么返回一个空矩阵，即[ ]。

find(A) 返回一个长的列向量，表示A中包含非零元素的下标向量。

[u, v]=find(A) 返回向量u和v，它们包含A中的非零元素的下标，即A中元素(u_k, v_k)为非零。

[u,v,b] = find(A) 返回包含A中非零元素的下标向量u和v以及一个包含对应非零元素的向量。A中元素(u_k, v_k)为非零并且能在b_k中找到。

命令find可以与关系运算符合用，这样使命令更有用。例如:

index = find(x>0.5)

8.逻辑函数(续)

any(x) 如果x中的有一个元素为非零值，那么返回1；否则，返回0。

any(A) 对A进行列运算，根据相应列是否包含非零元素，返回一个带1和0的行向量。

all(x) 如果所有的元素都是非零值，返回1；否则，返回0。

all(A) 对A进行列操作，根据相应列是否所有元素都为非零值，返回带1和0的一个行向量。

例：all (x < = 5)、all (all (A == A′))

isnan(A) 返回一个维数与A相同的矩阵，在这个矩阵中，对应A中有‘N a N’处为1，其他地方为0。

isinf(A) 返回一个维数与A相同的矩阵，在这个矩阵中，对应A中有‘inf’处为1，其他地方为0。

isempty(A) 如果A是一个空矩阵，返回1；否则返回0。

isequal(A, B) 如果A和B是相同的，即有相同的维数和相同的内容，则返回1。

isreal(A) 如果A是一个不带虚部的实矩阵，则返回1；否则，返回零。

isfinite(A) 返回一个与A维数相同的矩阵。在这个矩阵中， A中元素是有限的，则对应元素为1；否则，为零。

二、图形功能——2维

1. 2维图形生产 ——函数plot

例子：1) y=[0 0.58 0.70 0.95 0.83 0.25];plot(y)

2) x=0:2*pi/30:2*pi; y=sin(x); plot(x,y)

多条曲线：

```
x=0:2*pi/30:2*pi;
y1=sin(x); y2=cos(x);
plot(x,y1,x,y2)
```

命令hold on；hold off

函数linspase

线型和颜色： plot(x,y1,'.+',x,y2,'-*')

标记：xlabel, ylabel, title, text, gtext('sinx')

坐标系控制：axis

多幅图形：subplot(m,n,p)——画面分成m×n区域的第p个区域中画图；

画函数图象：

fplot('sin(x)./x',[-20 20 -0.4 1.2]), gtext('sin(x)/x')

二、图形功能——3维

2. 3维图形——mesh、surf、plot3()等

例子：1)做函数图形 $z=\dfrac{\sin\sqrt{x^2+y^2}}{\sqrt{x^2+y^2}}$

```
x=-10:0.3:10;y=x;
[X,Y]=meshgrid(x,y);
Z=sqrt(X.^2+Y.^2);
Z=sin(Z)./Z;
mesh(X,Y,Z); (或surf(X,Y,Z))
```

2) 空间曲线——螺旋线

```
t=0:pi/50:10*pi;  plot3(sin(t),cos(t),t)
```

3) 等高线 contour

对1),

```
hold on;contour3(X,Y,Z,10，’r’)
```

三、多项式：

多项式是用向量形式来表示，从最右边算起，第一个为0阶系数，第二个为1阶系数，依次类推。例如一个元三次多项式：**4x^3+3x^2+2x+1**用向量**[4 3 2 1]**来表示。

1.多项式的运算：

1) 乘：**conv**指令执行多项式的相乘运算，指令格式为：**z=conv (x,y)** 例如：

```
x=[1 3 5]; y=[2 4 6];
z=conv(x,y)
z =
     2    10    28    38    30
```

如果要对两个以上的多项式进行相乘，可以重复使用**conv**指令，例如：（**x,y**同上）

```
conv(conv(x,y),x)
ans =
     2    16    68   172   284   280   150
```

2) 分解：与1）相反，用deconv指令，其指令格式为：

[z,r]=deconv (x,y)——表示x除以 y商为z，余数为r。

例如：[z,r]=deconv(z,x)

```
z =
     2     4     6
r =
     0     0     0     0     0
```

3) 求根：roots指令用于求多项式的根。例如：

```
fx=[1 3 2];
rootoffx=roots(fx)
rootoffx =
    -2
    -1
```

有如roos([1 0 1])

4） **polyval (p,x)** 计算多项式**p**在**x**处的值，其中**x**可以是数或向量或矩阵。

例如：

```
p = [1  -6  11  -6];
x=1;  p1=polyval (p,x)  结果为：   p1 =     0
x=[1,2,3];  p2=polyval (p,x)  结果为：
p2 =     0     0     0
x=A;  p3=polyval (p,x)   结果为：
p3 =
      0     0     0
      6    24    60
    120   210   336
```

5） **polyder (p)** 求**p**的微分多项式。 例如：

```
p=[1 -6 11 -6];
dp=polyder(p)
dp =
   3   -12    11
```

6） [r,p,k]=residue (a,b) 求a/b的部分因式分解。若多项式a,b都没有重根,则:

$$\frac{b(s)}{a(s)} = \frac{r_1}{s-p_1} + \frac{r_2}{s-p_2} + \ldots + \frac{r_n}{s-p_n} + k(s)$$

例如用求x/(x^2+3x+2)的部分因式分解:

```
x=[1 0];  y=[1 3 2];
[r,p,k]=residue(x,y)
 r =
    2
   -1
 p =
   -2
   -1
 k =
   [ ]
```

又例:

```
x = [ 5 3 -2 7];
y = [-4 0 8 3];
[r, p, k] = residue(x,y);
```

当输入三个参数 r,p,k 时，该函数又会生成原来的函数。例如:

```
[x,y]=residue(r,p,k)
   x =
     1     0
   y =
     1     3     2
```

三、符号变量、符号运算——符号运算工具箱

函数**sym**用于生成符号变量和符号表达式，如：

x=sym('x'); a=sym('alpha')

分别创建变量**x, alpha**

f=sym('a*x^2+b*x+c')创建变量表达式**f**，但要注意此式并没有自动创建变量**a, b, c, x**。可以用函数**syms**对多个变量同时定义，如：

syms a b c x

函数**sym**也可以用来表示确定的函数，如

f=sym ('f(x)')

函数**f (x)**。

常见符号计算：

1. 微分：**diff**是求微分最常用的函数。其输入参数既可以是函数表达式，也可以是符号矩阵。**Diff (f, x, n)**表示对**f**关于**x**求**n**阶导数。例如：

1) 下面程序段将生成表达式**sin (ax),** 并分别对其中的**x**和**a**求导。

```
syms  a  x
f=sin(a*x);
df=diff(f,x)
 df =
      cos(a*x)*a
dfa=diff(f,a)
 dfa =
      cos(a*x)*x
```

2) 若输入参数为矩阵，将对矩阵中的每个元素求导.

```
syms  a  x
A=[-sin(a*x),sin(a*x);cos(a*x),cos(a*x)]
A =
   [ -sin(a*x),  sin(a*x)]
   [ cos(a*x),  cos(a*x)]
dy=diff(A,x)
dy =
   [ -cos(a*x)*a, cos(a*x)*a]
   [ -sin(a*x)*a, -sin(a*x)*a]
```

3）、可用函数jacobian来计算Jacobi矩阵。

```
syms r l f
x=r*cos(l)*cos(f);
y=r*cos(l)*sin(f);
z=r*sin(l);
J=jacobian([x;y;z],[r l f])
```

```
J =
 [ cos(l)*cos(f), -r*sin(l)*cos(f), -r*cos(l)*sin(f)]
 [ cos(l)*sin(f), -r*sin(l)*sin(f),  r*cos(l)*cos(f)]
 [     sin(l),           r*cos(l),          0       ]
```

2. 积分：用函数int来求符号表达式的积分。命令格式为：

int (f, r, x0, x1)其中f为所要积分的表达式，r为积分变量，若为定积分，则x0,x1为积分上下限。例：

```
syms x;
f=exp(-x^2)
f =
exp(-x^2)
int(f,x)
ans =
        1/2*pi^(1/2)*erf(x)
int(f,x,-inf,inf)
ans =
        pi^(1/2)
```

$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$$

3. 级数求和：函数用于对符号表达式求和。

例：

```
syms k;
s1=symsum(1/k^2,1,inf)
s1 =
    1/6*pi^2
s2=symsum(1/k,1,inf)
s2=
      inf
```

4. 极限：用函数limit来求表达式的极限。

函数limit的常用调用格式：

数学表达式	命令格式
$\lim_{x\to 0} f(x)$	Limit (f),或limit (f , x)
$\lim_{x\to a} f(x)$	Limit (f , x , a) , 或 limit (f , a
$\lim_{x\to a^-} f(x)$	Limit (f , x , a , ‘left’)
$\lim_{x\to a^+} f(x)$	Limit (f , x , a , ‘right’)

```
syms x;f=exp(-x^2);
limit(f,x,inf)
ans=
     0
```

5.化简：

1）、**collect (f)** 将表达式中相同次幂的项合并，也可以再输入一个参数指定以哪个变量的幂次合并。

2）、**expand (f)** 将表达式展开。

3）、**horner (f)** 将表达式转换为嵌套格式。——减少运算次数

4）、**factor (f)** 将表达式分解因式，并且分解后的多项式的所有系数都为有理数。

5）、**simplify (f)** 利用函数规则对表达式进行化简。

设有解析函数$f(x)=x^2\sin^2 x$，利用 **MATLAB** 的符号运算工具箱可以对该函数进行解析推导，得出诸如高阶导数、积分、**Taylor** 幂级数展开等。

```
clear
clc
syms x; f='x^2*(sin(x))^2';
diff(f); f1=simple(ans)
diff(f,x,2); f2=simple(ans)
diff(f,x,3); f3=simple(ans)
diff(f,x,4); f4=simple(ans)
int(f4,x)
taylor(x^2*(sin(x))^2,15,x)
```

MATLAB的程序设计

脚本文件和函数文件

M文件有两种形式：脚本文件(Script File)和函数文件(Function File)。这两种文件的扩展名，均为“ . m”。

1.1 M脚本文件

- 对于一些比较简单的问题，在指令窗中直接输入指令计算。
- 对于复杂计算，采用脚本文件（Script file）最为合适。
- MATLAB只是按文件所写的指令执行。
- M脚本文件的特点是：
  - 脚本文件的构成比较简单，只是一串按用户意图排列而成的(包括控制流向指令在内的)MATLAB指令集合。
  - 脚本文件运行后，所产生的所有变量都驻留在 MATLAB 基本工作空间(Base workspace)中。只要用户不使用清除指令(clear)， MATLAB指令窗不关闭，这些变量将一直保存在基本工作空间中。

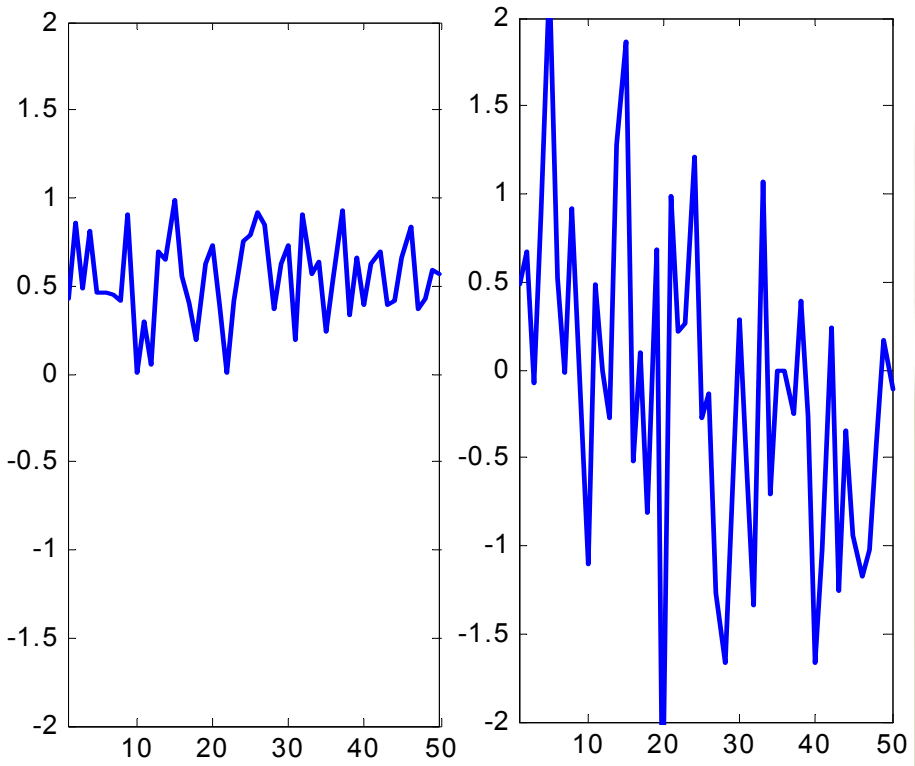
M文件脚本文件建立方法：

1. 在Matlab中,点: File->New->M-file, 生成编辑窗口
2. 在编辑窗口中输入复杂的计算命令集合；
3. 点：File->Save，存盘，文件名任意——易记忆的.

例、随机产生50个随机数，并用图形显示结果(比较均匀分布与正态分布)

```
a=rand(1,50);
subplot(1,2,1);
plot(a);

b=randn(1,50);
subplot(1,2,2);
plot(b);
```



1 脚本文件和函数文件（续1）

1.2 M函数文件

- 与脚本文件不同，函数文件犹如一个“黑箱”，把一些数据送进并经加工处理，再把结果送出来。
- **MATLAB**提供的函数指令大部分都是由函数文件定义的。
- **M**函数文件的特点是：
  - 从形式上看，与脚本文件不同，函数文件的第一行总是以 “**function**”引导的“函数申明行”。
  - 从运行上看，与脚本文件运行不同，每当函数文件运行，**MATLAB**就会专门为它开辟一个临时工作空间，称为函数工作空间（**Function workspace**）。当执行文件最后一条指令时，就结束该函数文件的运行，同时该临时函数空间及其所有的中间变量就立即被清除。
  - **MATLAB**允许使用比 “标称数目 ”较少的输入输出变量，实现对函数的调用。

1.3 M文件的一般结构

- 由于从结构上看，脚本文件只是比函数文件少一个“函数申明行”，所以只须描述清楚函数文件的结构 。
- 典型 M函数文件的结构如下 ：
  - 函数申明行：位于函数文件的首行，以关键字 **function** 开头，函数名以及函数的输入输出宗量都在这一行被定义。
  - 第一注释行：紧随函数申明行之后以%开头第一注释行。该行供**lookfor**关键词查询和 **help**在线帮助使用 。
  - 在线帮助文本区 ：第一注释行及其之后的连续以%开头的所有注释行构成整个在线帮助文本。
  - 编写和修改记录：与在线帮助文本区相隔一个“空”行，也以%开头，标志编写及修改该M文件的作者和日期等 。
  - 函数体：为清晰起见，它与前面的注释以“空”行相隔。

M文件建立方法：

1. 在Matlab中点:File->New->M-file
2. 在编辑窗口中输入程序内容
3. 点：File->Save存盘，文件名必须函数名一致。

例：定义函数 $f(x_1,x_2)=100(x_2-x_1^2)^2+(1-x_1)^2$

1.建立M文件：fun.m

```
function    f=fun(x)
f=100*(x(2)-x(1)^2)^2+(1-x(1))^2;
```

2. 可以直接使用函数fun.m

例如：计算 f(1,2), 只需在Matlab命令窗口键入命令：
x=[1 2]；fun(x)

2 函数调用和参数传递

2.1 局部变量和全局变量

- 局部(Local)变量：它存在于函数空间内部的中间变量，产生于该函数的运行过程中，其影响范围也仅限于该函数本身 。

- 全局(Global)变量：通过 global 指令，MATLAB也允许几个不同的函数空间以及基本工作空间共享同一个变量，这种被共享的变量称为全局变量。

2.2 函数调用

- 在MATLAB中，调用函数的常用形式是：

[输出参数1,输出参数2,…] = 函数名(输入参数1,输入参数2, …)

- 函数调用可以嵌套，一个函数可以调用别的函数，甚至调用它自己 (递归调用)。

2.3 参数传递

- **MATLAB**在函数调用上有一个与众不同之处：函数所传递的参数具有可调性。
- 传递参数数目的可调性来源于如下两个**MATLAB**永久变量：
  - 函数体内的 **nargin** 给出调用该函数时的输入参数数目。
  - 函数体内的 **nargout** 给出调用该函数时的输出参数数目。
- 只要在函数文件中包括这两个变量，就可以知道该函数文件调用时的输入参数和输出参数数目。

- 值得注意：**nargin**、 **nargout** 本身都是函数，不是变量，所以用户不能赋值，也不能显示。

- “变长度”输入输出变量：**varargin** 、 **varrgout**。具有接受“任意多输入”、返回“任意多输出”的能力。

2.3 MATLAB的程序结构和控制流

2.3.1程序流控制

- 循环结构： MATLAB提供两种循环方式。

  for—end 循环和while---end循环。

- 分支结构： if—[else]—end 。//if—elseif—[else]—end

- switch---case 结构。

```
for  x=array
{commands}
end
```

```
while  expression
{commands}
end
```

```
if  expression
    {commands1}
else
    {commands2}
end
```

```
if  (expression1)
    {commands1}
elseif  (expression2)
    {commands2}
elseif  (expression3)
    {commands3}
elseif ……
      ……
else
   {commands}
end
```

```
switch (a)
case  (a＝)1
    {commands1}
case  (a=)2
    {commands3}
case  3
      ……
otherwise
   {command n}
end
```

2．3 MATLAB的程序结构和控制流（续）

2．3．3 图形用户界面（GUI）编程

- 现代的主流应用程序已经从命令行的交互方式转变为以图形界面为主的交互方式，这主要是由于它给用户带来了操作和控制的方便与灵活性。（面向对象编程）
- **MATLAB**能够以比较简单的方式实现一系列的图形界面功能。通过对控件、菜单属性的设置和 **Callback** 的编写，就能够满足大多数用户的需求。
- 控件的 **Callback** 属性： **Callback** 属性的取值是字符串，可以是某个**M**文件名或一小段**MATLAB**语句。当用户激活控件对象（例如：在控件对象图标上单击鼠标左键）时，应用程序就运行 **Callback** 属性定义的子程序。
- 菜单的 **Callback** 属性： **Callback** 属性的取值是字符串，可以是某个**M**文件名或一小段**MATLAB**语句。当用户激活菜单对象时，若没有子菜单就运行 **Callback** 属性定义的子程序。若有，先运行 **Callback** 属性定义的子程序，再显示子菜单。

2.4 M文件的调试

- 编写 M文件时，错误（Bug）在所难免。错误有两种：语法（Syntax）错误和运行（Run-time）错误。

- 语法错误是指变量名、函数名的误写，标点符号的缺、漏等。对于这类错误，通常能在运行时发现，终止执行，并给出相应的错误原因以及所在行号。

- 运行错误是算法本身引起的，发生在运行过程中。相对语法错误而言，运行错误较难处理 。尤其是M函数文件，它一旦运行停止，其中间变量被删除一空，错误很难查找。

- 有两种调试方法：直接调试法和工具调试法。

2.4 M文件的调试（续1）

- 直接调试法：可以用下面方法发现某些运行错误。
  - 在M文件中，将某些语句后面的分号去掉， 迫使M文件输出一些中间计算结果，以便发现可能的错误。
  - 在适当的位置，添加显示某些关键变量值的语句（包括使用 **disp** 在内）。
  - 在原M脚本或函数文件的适当位置，增添指令 **keyboard** 。 **keyboard** 语句可以设置程序的断点 。
- 设置断点，单步执行来查找算法错误——**debug**;

2.5 编程实践

例1　设银行年利率为11.25%.将10000元钱存入银行，问多长时间会连本带利翻一番？

脚本文件实现方式(ex1.m)：

```
money=10000
years=0;
rot=1+11.25/100
while money<20000
    years=years+1;
    money=money*rot;
end
```

函数文件实现方式(ex2.m)：

```
function [years,
money]=ex2(money,year)
%
%
mony0=money;
rot=1+11.25/100;
If nargin>1
    for i=1:year
        money=money*rot;
    end
else
    year=0;
    while money<2*money0
        years=years+1;
        money=money*rot;
    end
end
```

2.5 编程实践

例2 计算分段函数的定积分

$$f(x)=\begin{cases} x^2+1 & x>1 \\ 2x & 0<x\le 1, \\ x^3 & x\le 0 \end{cases}$$

脚本文件计算定积分(ex3.m)：

```
a=-3;b=5;
M=1000;
detx=(b-a)/M;
s=0;
for i=1:M
   s=s+fun1(a+detx*i);
end
s=s*detx
```

编写函数文件 (fun1.m)：

```
function f=fun1(x)
%分段函数定义
%
if x>1
   f=x^2+1;
elseif x<=0
   f=x^3;
else
   f=2*x;
end
```

2.5 编程实践

例3　一球从100米高度自由落下,每次落地后反跳回原高度的一半,再落下. 求它在第10次落地时,共经过多少米?第10次反弹有多高?

脚本文件计算 (ex4.m):

```
a(1)=100;
b=a(1);
for i=2:11
    a(i)=a(i-1)/2;
    b=b+a(i);
end
aa=zeros(1,2*length(a));
aa(1:2:end)=a;
plot(aa)
```