插值：求过已知有限个数据点的近似函数。  
拟合：已知有限个数据点，求近似函数，可不过已知数据点，只要求在某种意义  
下它在这些点上的总偏差最小。  
插值和拟合都是要根据一组数据构造一个函数作为近似，由于近似的要求不同，二者的数学方法上是完全不同的。而面对一个实际问题，究竟应该用插值还是拟合，有时容易确定，有时则并不明显。

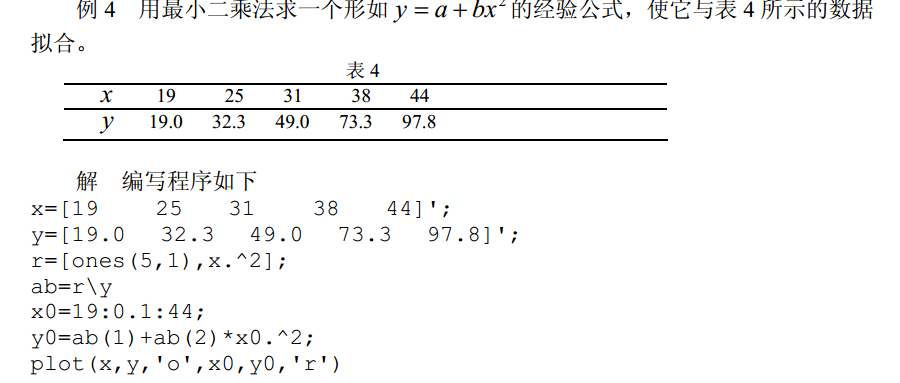
常见的插值：拉格朗日多项式插值、牛顿插值、分段线性插值、 Hermite 插值和三次样条插值。

拟合：拟合的实现分为MATLAB和excel实现。

MATLAB的实现就是polyfit函数：主要是多项式拟合。

更复杂的函数拟合，使用的是最小二乘法，或者其他方法。但是需要经验公式：

例如：



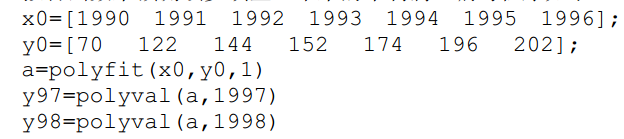
此代码比较简单，大家自己看书就能立刻看明白。

关于拟合：拟合可以用excel，也可以用MATLAB，关于excel的用法。大家自己探索，提示：添加趋势线。

关于matlab，需要了解一些函数：

Polyfit polyval 其余参考MATLAB汇总中的MATLAB常用函数参考。

Polyfit是多项式拟合：



需要输入x,y的数据，x和y个数一致，然后polyfit（x,y,n）n表示需要拟合的次数。Polyval一般套用在polyfit后，用法看上图。

插值：插值是相对拟合略微麻烦一点点：

插值的函数 interp2，这个大家经常见，关于interp2的用法网上介绍的很多。这里有一个需要注意的事项就是：以下为例：

x0=[1200:400:4000];

y0=[1200:400:3600];

z0=[1130 1250 1280 1230 1040 900 500 700;

1320 1450 1420 1400 1300 700 900 850;

1390 1500 1500 1400 900 1100 1060 950;

1500 1200 1100 1350 1450 1200 1150 1010;

1500 1200 1100 1550 1600 1550 1180 1070;

1500 1550 1600 1550 1600 1600 1600 1550;

1480 1500 1550 1510 1430 1300 1200 980];

[xi,yi]=meshgrid(1200:10:4000,1200:10:3600);%网格化处理，这一步很多人会忘掉，往往会输入为：x1=1200:10:4000;y1=1200:10:3600的形式注意

zil=interp2(x0,y0,z0,xi,yi,'linear');%使用linear进行插值，可以替换，更多的可以在MATLAB中使用help interp2

mesh(xi,yi,zil)

---------------------------------------------------------------------下面我们简单介绍meshgrid intrerp griddate

**重要，需要了解，尤其第四条（最后一页）interp2与griddata的区别**

**1. meshgrid**

meshgrid用于从数组a和b产生网格。生成的网格矩阵A和B大小是相同的。它也可以是更高维的。

[A,B]=Meshgrid(a,b)

生成size(b)Xsize(a)大小的矩阵A和B。它相当于a从一行重复增加到size(b)行，把b转置成一列再重复增加到size(a)列。因此命令等效于：

A=ones(size(b))\*a;

B=b'\*ones(size(a))

如下所示：

>> a=[1:2]

a =

1 2

>> b=[3:5]

b =

3 4 5

>> [A,B]=meshgrid(a,b)

A =

1 2

1 2

1 2

B =

3 3

4 4

5 5

>> [B,A]=meshgrid(b,a)

B =

3 4 5

3 4 5

A =

1 1 1

2 2 2

**2. interp**

interp1——一维数据插值函数

一维数据插值。该函数对数据点之间计算内插值，它找出一元函数f(x)在中间点的数值，其中函数表达式由所给数据决定。

yi=interp1(x,Y,xi)：返回插值向量yi，每一元素对应于参量xi，同时由向量X与Y的内插值决定。参量x 指定数据Y的点。若Y为一矩阵，则按Y的每列计算。yi是阶数为length(xi)\*size(Y,2)的输出矩阵。

yi=interp1(Y,xi)：假定x=1:N，其中N为向量Y的长度，或者为矩阵Y的行数。

yi=interp1(x,Y,xi,method)：用指定的算法计算插值。nearest为最近邻点插值，直接完成计算；linear为线性插值（默认方式），直接完成计算；spline为三次样条函数插值。

yi=interp1(x,Y,xi,method,'extrap')：对于超出x范围的xi中的分量将执行特殊的外插值法extrap。

yi=interp1(x,Y,xi,method,extrapval)：确定超出x范围的xi中的分量的外插值extrapval，其值通常取NaN或0。

interp2函数——二维数据内插值

完成二维的数据插值。

ZI=interp2(X,Y,Z,XI,YI)：返回矩阵ZI，其元素包含对应于参量XI与YI（可以是向量、或同型矩阵）的元素。用户可以输入行向量和列向量Xi与Yi，此时，输出向量Zi与矩阵meshgrid(xi,yi)是同型的。同时取决于由输入矩阵X、Y与Z确定的二维函数Z=f(X,Y)。

ZI=interp2(Z,XI,YI)：默认地，X=1:n、Y=1:m，其中[m,n]=size(Z)。再按第一种情形进行计算。

ZI=interp2(Z,n)：作n次递归计算，在Z的每两个元素之间插入它们的二维插值，这样，Z的阶数将不断增加。interp2(Z)等价于interp2(z,1)。

ZI=interp2(X,Y,Z,XI,YI,method)：用指定的算法method计算二维插值。linear为双线性插值算法（默认算法），nearest为最临近插值，spline为三次样条插值，cubic为双三次插值。

interp3函数——三维数据插值

完成三维数据插值。

VI=interp3(X,Y,Z,V,XI,YI,ZI)：求出由参量X,Y,Z决定的三元函数V=V(X,Y,Z)在点（XI,YI,ZI）的值。参量XI,YI,ZI是同型阵列或向量。若向量参量XI,YI,ZI是不同长度、不同方向（行或列）的向量，这时输出参量VI与Y1,Y2,Y3为同型矩阵。Y1,Y2,Y3为用函数meshgrid(XI,YI,ZI)生成的同型阵列。若插值点(XI,YI,ZI)中有位于点（X,Y,Z）之外的点，则相应地返回特殊变量值NaN。

VI=interp3(V,XI,YI,ZI)：默认地，X=1:N，Y=1:M，Z=1:P，其中，[M,N,P]=size(V)，再按上面的情形计算。

VI=interp3(V,n)：作n次递归计算，在V的每两个元素之间插入它们的三维插值。这样，V的阶数将不断增加。interp3(V)等价于interp3(V,1)。

VI=interp3(...,method)：用指定的算法method做插值计算。linear为线性插值（默认算法），cubic为三次插值，spline为三次样条插值，nearest为最邻近插值。

interpn函数——n维数据插值

完成n维数据插值。

VI=interpn(X1,X2,...,Xn,V,Y1,Y2,..,Yn)：返回由参量X1,X2,..,Xn,V确定的n元函数V=V(X1,X2,..,Xn)在点（Y1,Y2,...,Yn）处的插值。参量Y1,Y2,...,Yn是同型的矩阵或向量。若Y1,Y2,...,Yn是向量，则可以是不同长度，不同方向（行或列）的向量。

VI=interpn(V,Y1,Y2,...,Yn)：默认地，X1=1:size(V,1),X2=1:size(V,2),...,Xn=1:size(V,n)，再按上面的情形计算。

VI=interpn(V,ntimes)：作ntimes递归计算，在V的每两个元素之间插入它们的n维插值。这样，V的阶数将不断增加。interpn(V)等价于interpn(V,1)。

**3. griddata**

功能 数据格点

格式

(1)ZI = griddata(x,y,z,XI,YI)

用二元函数z=f(x,y)的曲面拟合有不规则的数据向量x,y,z。griddata 将返回曲面z 在点（XI,YI）处的插值。曲面总是经过这些数据点（x,y,z）的。输入参量（XI,YI）通常是规则的格点（像用命令meshgrid 生成的一样）。XI 可以是一行向量，这时XI 指定一有常数列向量的矩阵。类似地，YI 可以是一列向量，它指定一有常数行向量的矩阵。

(2)[XI,YI,ZI] = griddata(x,y,z,xi,yi)

返回的矩阵ZI 含义同上，同时，返回的矩阵XI,YI 是由行向量xi 与列向量yi 用命令meshgrid 生成的。

(3)[XI,YI,ZI] = griddata(.......,method)

用指定的算法method 计算：

‘linear’：基于三角形的线性插值（缺省算法）；

‘cubic’： 基于三角形的三次插值；

‘nearest’：最邻近插值法；

‘v4’：MATLAB 4 中的griddata 算法。

**4. matlab二维插值--interp2与griddata**

二者均是常用的二维差值方法，两者的区别是，interp2的插值数据必须是矩形域，即已知数据点（x,y)组成规则的矩阵，或称之为栅格，可使用meshgid生成。而griddata函数的已知数据点（X，Y）不要求规则排列，特别是对试验中随机没有规律采取的数据进行插值具有很好的效果。griddata(X,Y,XI,YI,'v4') v4是一种插值算法，没有具体的名字，原文称为“MATLAB 4 griddata method”，是一种很圆滑的差值算法，效果不错。X和Y提供的已知数据点，XI和YI是需要插值的数据点，一般使用meshgrid生成，当然也可以其他数据，但是那样绘图的时候就比较麻烦，不能使用mesh等，只能使用trimesh。

示例如下：

a=[

3 3 1.5300

3 27 0.4210

5 17 0.5980

9 9 0.5900

13 25 0.4470

15 15 1

17 5 0.3830

21 21 0.3100

25 13 0.2830

27 3 0.2820

27 27 0.1200

];

x=a(:,1);

y=a(:,2);

z=a(:,3);

xtemp=linspace(min(x),max(x),100);

ytemp=linspace(min(y),max(y),100);

[X,Y]=meshgrid(xtemp,ytemp);

Z=griddata(x,y,z,X,Y,'v4');

surf(X,Y,Z)

shading interp