Mathworks Tech-Note 1508

曲线拟合向导

第 1 节： 简介

MATLAB 即有内建的解决很多通常遇到的曲线拟合问题的能力，又具有附加这方面的产品。本技术手册描述了几种拟合给定数据集的曲线的方法，另外，本手册还解释了加权曲线拟合、针对复数集的曲线拟合以及其他一些相关问题的拟合技巧。在介绍各种曲线拟合方法中，采用了典型例子的结合介绍。

第 2 节： MathWorks 产品的曲线拟合特色

MATLAB 有可以用于曲线拟合的内建函数。MathWorks 公式也提供了很多工具箱可以用于曲线拟合。这些方法可以用来做线性或者非线性曲线拟合。MATLAB 也有一个开放的工具箱－－曲线拟合工具箱（Curve Fitting Toolbox），她可以用于参数拟合，也可以用于非参数拟合。本节将介绍曲线拟合工具箱与其他工具箱、以及各种 MATLAB 可以用于曲线拟合的内建函数的详细特征。

a. 曲线拟合工具箱

曲线拟合工具箱是专门为数据集合进行曲线拟合而设计的。这个工具箱集成了用 MATLAB 建立的图形用户界面（GUIs）和 M 文件函数。

- 利用工具箱的库方程（例如线性，二次，高阶多项式等）或者是用户自定义方程（局限于用户的想象力）可以进行参数拟合。当你想找出回归系数以及他们背后的物理意义的时候就可以采用参数拟合。
- 通过采用平滑样条或者其他各种插值方法，你就可以进行非参数拟合。当回归系数不具有物理意义并且不在意他们的时候，就采用非参数拟合方法。

曲线拟合工具箱提供了如下功能：

- 数据回归，譬如 截面（？sectioning）与平滑；
- 标准线性最小二乘拟合，非线性最小二乘拟合，加权最小二乘拟合，约束二乘（constrained least squares）拟合 以及 稳健（robust）拟合；
- 根据诸如 R^2 以及 误差平方和（SSE）确定的拟合性能的统计特征。

请查阅曲线拟合工具箱提供的 demos。

b. MATLAB 内建函数与具有曲线拟合能力的其他工具箱

除了曲线拟合工具箱，MATALB 与其他工具箱也提供了些可以用于解决线性和非线性曲线拟合的功能。本节列举并解释了其中几个。

c. 利用 MATLAB 内建函数进行线性曲线拟合

函数	描 述
polyfit	用多项式进行数据拟合。polyfit（X,Y,N）对数据 X,Y 拟合 N 阶多项式系数，P(X(I))~=Y(I)， 在最小二乘意义上。
\	反斜线或者矩阵阵左除。如果 A 是一个方阵，A\B 基本上与 inv（A）*B 一致的，是采用的不同计算方式而已。
polyval	在给定点计算多项式的值
corrcoef	计算两个向量的相关系数。它可以与 polyfit 和 polyval 函数一起用来在实际数据和拟合输出之间计算 R^2 相关系数

下面给出一个利用 corref 计算 R^2 值的例子：

```
load census
[p,s]=polyfit(cdate,pop,2);
Output=polyval(p,cdate);
Corrolation=corroef(cate,Output);
```

cdate 与它自身很好的相关，同样的 Output 也与它自身很好相关。反对角线上元素是

cdate 与 Output 之间的相关性。这个值非常接近于 1，因此实际数据与拟合结果能否较好的吻合。因此，这个拟合是“好”的拟合。（应该是这样判断的么？我怎么觉得应该通过 pop 与 Output 的相关性来判断拟合的好坏的呢？）

利用反斜线操作符与 polyfit 函数进行回归与曲线拟合的更多的例子请参照 MATLAB 文档中的 Regression and Curve Fitting 一节。

附加例子：

数据集：

```
t = [0 .3 .8 1.1 1.6 2.3]';
y = [0.5 0.82 1.14 1.25 1.35 1.40]';
plot(t,y,'o'), grid on
```

方法 1：多项式回归

基于图形，数据可能通过二次多项式建模如下：

$$y = a0 + a1*t + a2*t$$

其中未知系数 a0，a1，a2 可以通过最小二乘（通过最小化通过模型计算出来的数据的偏差的平方和）拟合计算。三个未知数 6 个方程如下：

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{bmatrix} = \begin{bmatrix} 1 & t_1 & t_1^2 \\ 1 & t_2 & t_2^2 \\ 1 & t_3 & t_3^2 \\ 1 & t_4 & t_4^2 \\ 1 & t_5 & t_5^2 \\ 1 & t_6 & t_6^2 \end{bmatrix} \times \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix}$$

用 6x3 的矩阵表示：

```
X = [ones(size(t))    t    t.^2];
```

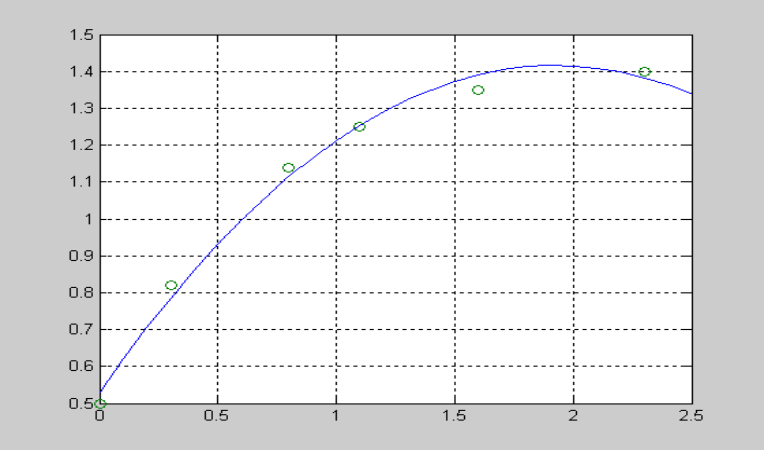
则结果通过反斜线操作符得到：

```
a=X\y
a =
   0.5318
   0.9191
  -0.2387
```

因此二阶多项式模型为： $y = 0.5318+0.9191*t-0.2387*t^2$

计算模型在均匀空间的值，并将原来的值画在同一个图形上：

```
T=(0:.1:2.5)';
Y=[ones(size(T))  T  T.^2)]*a;
plot(T,Y,'-t,y,'o'),grid on
```

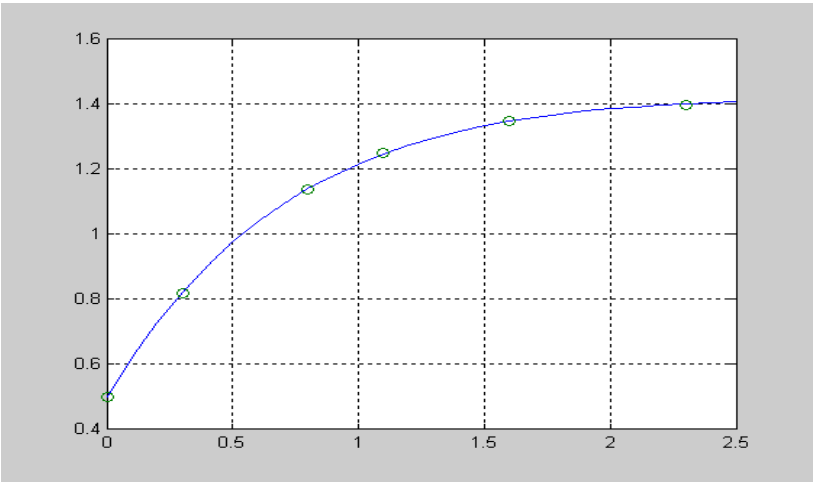


方法 2： 线性参数回归

建立模型：

$$y = a_0 + a_1 e^{-t} + a_2 t e^{-t}$$

```
X=[ones(size(t)) exp(-t) t.*exp(-t)];
a = X\y;
T=(0:.1:2.5)';
Y=[ones(size(T)) exp(-T)  T.exp(-T)]*a;
plot(T,Y,'-',t,y,'o'),grid on
```



方法 3： 多元回归

如果 y 是一个包含多个独立变量的函数，表示变量间的相邻关系的矩阵方程可以通过附加数据进行扩展。

假设我们测量参数 x1、x2 的少数几个值的输出 y，观测值如下：

```
x1 = [.2 .5 .6 .8 1.0 1.1]';
x2 = [.1 .3 .4 .9 1.1 1.4]';
y  = [.17 .26 .28 .23 .27 .24]';
```

本数据的一个多元模型是： $y = a_0 + a_1 x_1 + a_2 x_2$

多元回归解决的是通过最小二乘拟合求未知系数 a0，a1，a2。通过构造回归矩阵 X 构造和解决同步方程，依然采用反斜线操作符。

```
X=[ones(size(x1)) x1 x2];
a=X\y;
```

为了评价模型，求取绝对误差的最大值：

```
Y = X *a;
MaxErr = max(abs(Y-y));
```

实例分析：曲线拟合（本节来自 matlab 的在线帮助文档）

本节提供了以实际数据分析形式的在 MATLAB 中的数据分析基本能力的概貌。下面的例子是以收集的人口普查数据为基础，采用 MATLAB 函数对数据进行实验拟合：

- 多项式拟合
- 残差分析
- 指数拟合
- 误差界限

1. 导入数据

load census （**census.mat** 包含了美国 **1790** 年到 **1990** 年的人口数据）

其中包括两个变量：**cdate** 与 **pop**

cdate 是从 **1790** 以 **10** 递增到 **1990** 的一个列向量，是年份数；

pop 是 cdate 中年份相应的人口数据向量

2. 多项式拟合

首先我们想通过简单的多项式对普查数据进行拟合。利用 MATLAB 中的两个函数进行处理：polyfit 与 polyval 。

polyfit 函数是在给定阶次多项式上对数据进行最小二乘意思上的最优拟合。假设采用 4 阶多项式，拟合过程为：

```
>> p=polyfit(cdate,pop,4)
   Warning: Polynomial is badly conditioned. Remove repeated data points
   or try centering and scaling as described in HELP POLYFIT.
p = 1.0e+005 *
    0.0000   -0.0000    0.0000   -0.0126    6.0020
```

警告的产生是因为 polyfit 函数用很大的值 cdate 作为基本数据，用他来产生范德蒙矩阵（Vandermonde matrix），具体细节的可以在 polyfit 的 m 文件中看到。cdate 的展开导致尺度标准问题，一个解决办法就是标准化 cdate 数据。

预处理：标准化数据

标准化处理是为了提高后期数值计算精度而进行的尺度变化处理。一个处理办法是：

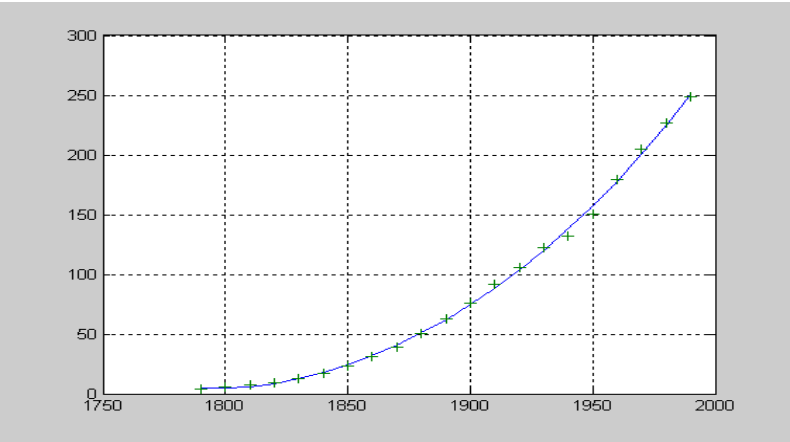
```
sdate=(cdate-mean(cdate))./std(cdate);
```

然后再以标准化后数据进行 4 阶多项式拟合：

```
>> p=polyfit(sdate,pop,4)
p = 0.7047    0.9210   23.4706   73.8598   62.2285
```

通过图形我们来观察其拟合的好坏：

```
>> pop4=polyval(p,sdate);
>> plot(cdate,pop4,'-',cdate,pop,'+'), grid on
```



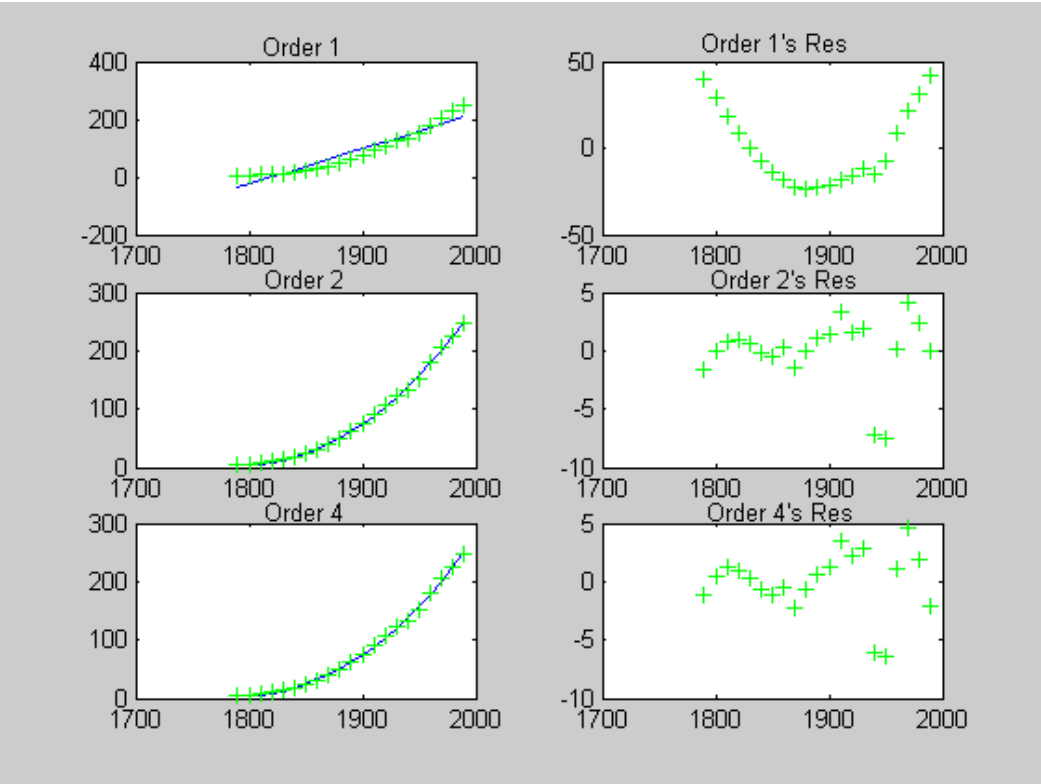
另外一个规范化数据的方法就是通过结果与单位的知识进行转换。如，对于本数据集，选择 1790 作为 0 年也可以得到较为满意的解。

3. 残差分析

一个评价拟合好坏的测度就是残差——观测值与预测值的差异。对不同的拟合，利用残差进行比较。从拟合图形和残差上，我们显而易见，采用标准化数据比简单的多项式可能对数据有更好的拟合。

利用如下命令，分别对数据进行 1 阶，2 阶，4 阶的拟合，作图，并比较其残差：

```
>> load census
>> sdate=(cdate-mean(cdate))./std(cdate);
>> p1=polyfit(sdate,pop,1);
>> pop1=polyval(p1,sdate);
>> plot(cdate,pop1,'b-',cdate,pop,'g+');
>> res1=pop-pop1;
>> figure,plot(cdate,res1,'g+');
>> p=polyfit(sdate,pop,2);
>> pop2=polyval(p,sdate);
>> figure,plot(cdate,pop2,'b-',cdate,pop,'g+')
>> res2=pop-pop2;
>> figure,plot(cdate,res2,'g+');
>> p=polyfit(sdate,pop,4);
>> pop4=polyval(p,sdate);
>> figure,plot(cdate,pop4,'b-',cdate,pop,'g+')
>> res4=pop-pop4;
>> figure,plot(cdate,res4,'g+');
>> max(abs(res1))
ans =      41.3987
>> max(abs(res2))
ans =      7.5361
>> max(abs(res4))
ans =       6.3455
```



4. 指数拟合

看前面的人口图形，发现人口数据曲线有些与指数曲线相似。利用这一点，我们试着对人口数据值的对数进行拟合，依然采用前面的标准化方法。

```
>> logp1=polyfit(sdate,log10(pop),1);
>> logpred1=10.^polyval(logp1,sdate);
>> semilogy(cdate,logpred1,'-',cdate,pop,'+');    grid on
>> logp2=polyfit(sdate,log10(pop),2);
>> logpred2=10.^polyval(logp2,sdate);
>> figure,semilogy(cdate,logpred2,'-',cdate,pop,'+');grid on
>> logres2=log10(pop)-polyval(logp2,sdate);
>> figure,plot(cdate,logres2,'+');
>> r=pop-10.^(polyval(logp2,sdate));
>> figure ,plot(cdate,r,'+')
```

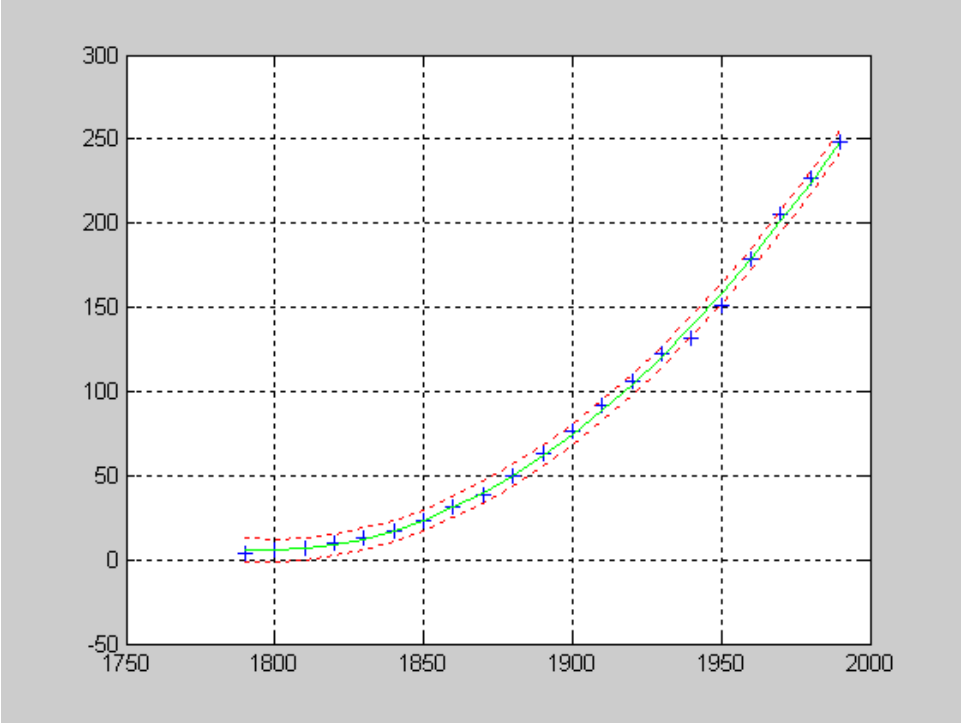
可以看出，残差更加随机性强一些。或许，残差随着人口数的增加而在数量级方面增长很快，但是总体而言，对数模型提供了更精确的拟合。

5. 误差界限

误差界限对于判别你的拟合是否是个合理的模型很有用。你可以通过从 polyfit 选择任意两个输出参数作为 polyfit 的两个输入参数。

本例子采用普查例子断乎机和前面讲的规范化方法，利用 polyfit 和 polyval 得到二阶多项式模型的误差界限。对年度值进行规范化，本例子采用＋－2△间隔，相应的其置信度的 **95**％。

```
>> load census
>> sdate=(cdate-mean(cdate))./std(cdate);
>> [p2,S2]=polyfit(sdate,pop,2);
>> [pop2,del2]=polyval(p2,sdate,S2);
>> plot(cdate,pop,'+',cdate,pop2,'g-',cdate,pop2+2*del2,'r:',cdate,pop2-2*del2,'r:'),grid on
```



优化工具箱的利用（接 1）

函 数	描 述
LSQLIN	有约束线性最小二乘优化
LSQNONNEG	非负约束线性最小二乘优化问题

当有约束问题存在的时候，应该采用上面的方法代替 Polyfit 与反斜线（\）。具体例子请参

阅优化工具箱文档中的相应利用这两个函数的例子。

d. 非线性曲线拟合

利用 MATLAB 的内建函数

函数名 描 述

FMINBND 只解决单变量固定区域的最小值问题

FMINSEARCH 多变量无约束非线性最小化问题（Nelder-Mead 方法）。

下面给出一个小例子展示一下如何利用 FMINSEARCH

1. 首先生成数据

```
>> t=0:.1:10;
>> t=t(:);
>> Data=40*exp(-.5*t)+rand(size(t));  % 将数据加上随机噪声
```

2. 写一个 m 文件，以曲线参数作为输入，以拟合误差作为输出

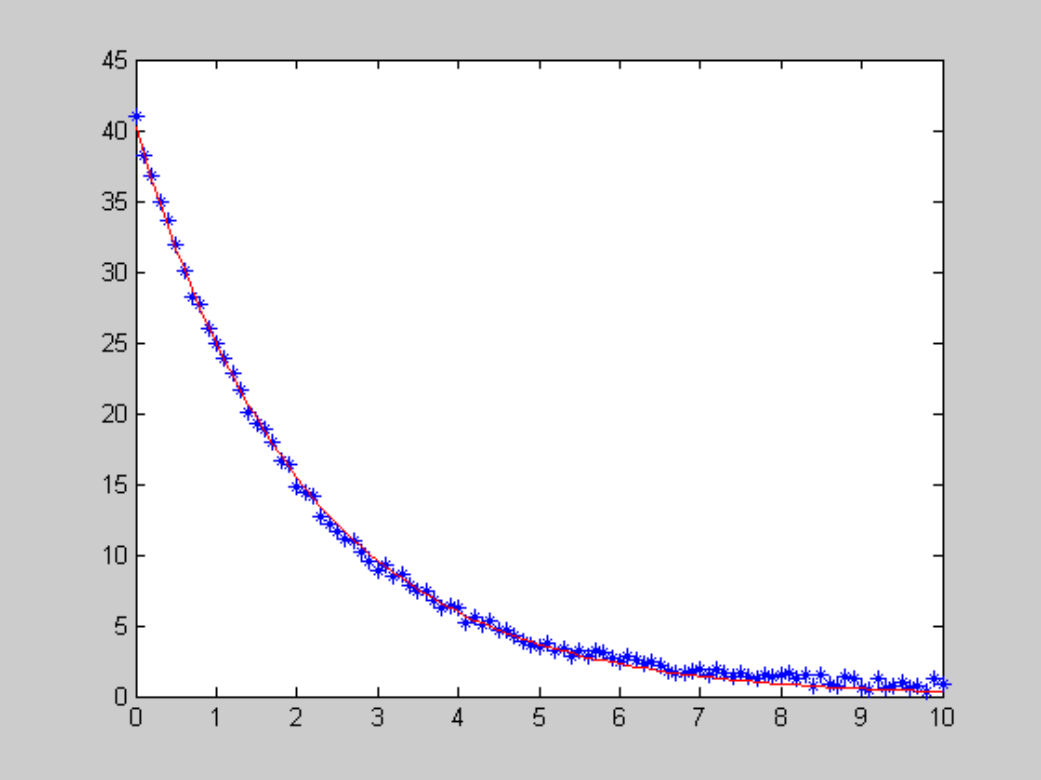
```
function sse=myfit(params,Input,Actural_Output)
A=params(1);
lamda=params(2);
Fitted_Curve=A.*exp(-lamda*Input);
Error_Vector=Fitted_Curve-Actural_Output;
% 当曲线拟合的时候，一个典型的质量评价标准就是误差平方和
sse=sum(Error_Vector.^2);
%当然，也可以将 sse 写作：sse=Error_Vector(:)'*Error_Vector(:);
```

3. 调用 FMINSEARCH

```
>> Strarting=rand(1,2);
>> options=optimset('Display','iter');
>> Estimates=fiminsearch(@myfit,Strarting,options,t,Data);
>> plot(t,Data,'*');
>> hold on
>> plot(t,Estimates(1)*exp(-Estimates(2)*t),'r');
```

Estimates 将是一个包含了对原数据集进行估计的参数值的向量。

附图见后面：



FMINSEARCH 通常能够用来解决不连续情况，特别是如果他们不出现在解的附近的时候。它得到的通常也是局部解。FMINSEARCH 只能够最小化实数值（也就是说，解的域必须只能包括实数，函数的输出只能够为实数值）。当感兴趣的是复数变量的域的时候，他们必须被分割为实部与虚部。

MATLAB 的 FIGURE 窗口：最基本的拟合界面与数据统计工具

MATLAB 通过基本的拟合界面也支持基本曲线拟合。利用这个界面，你可以快速地在简单易用的环境中实现许多基本的曲线拟合。这个界面可以实现以下功能：

a. 通过比样条插值（spline interpolant）、hermite 插值、或者是高达 10 阶的多项式插值实现数据的拟合；

b. 对给定数据同时实现多样插值的绘制；

c. 绘制残差图；

d. 检查拟合结果的残差的数值；

e. 通过内插值或者外推插值评价一个拟合结果；

f. 对拟合结果和残差的模进行图形绘制；

g. 将拟合结果保存入 MATLAB 工作空间。

开发你的拟合应用的时候，你可以通过基本拟合（Basic Fitting）界面，也可以通过命

令行函数，也可以同时作用。你可以通过基本拟合界面只能够实现 2－D 数据的拟合。然而，如果你用 subplot 绘制多个数据集，只要有至少一个数据集是 2D 的，那么就可以用基本拟合界面。

可以通过如下步骤激活基本拟合界面：

1. 绘制数据；

2. 从 figure 窗口的 **Tools** 菜单条下面选择 **Basic Fitting** 菜单项；

有关 Basic Fitting 界面的更多信息，请查阅 MATLAB 帮助文档的相应部分。

注意：对于 HP，IBM 以及 SGI 平台，MATLAB6.0（R12.0）以及 MATLAB6.1(R12.1)的基本拟合界面不受支持。

数据统计界面可以用来对图形中的每个数据集进行统计量的计算。可以通过如下步骤将数据统计界面激活：

1. 制数据；

2. 从 figure 窗口的 **Tools** 菜单条下面选择 **Data Statistics** 菜单项；

优化工具箱函数

LSQNONLIN 解决非线性最小二乘法问题，包括非线性数据拟合问题

LSQCURVEFIT 解决非线性数据拟合问题

下面给出利用这两个函数的例子：

LSQNONLIN：利用这个函数最小化连续函数只能够找到句柄解。下面的例子说明利用 LSQNONLIN 函数用下面的函数进行拟合：

f = A + B **exp(C***x)+D*exp(E*x)

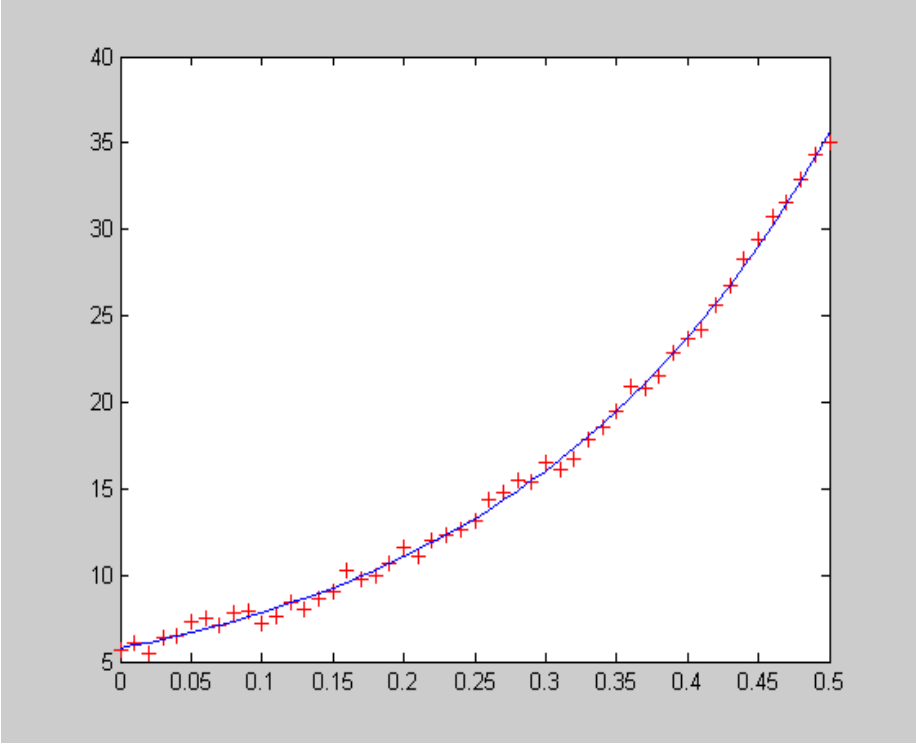
对数据集 x 与 y 进行拟合，其中 y 是在给定 x 的情况下的期望输出。为了解决这个问题，先建立下面的名为 fit_simp.m 的函数，它利用数据 x 与 y，将他们作为优化输入参数传递给 LSQNONLIN。利用给定的数据 x 计算 f 的值，再与原始数据 y 进行比较。经验值与实际计算出的值之间的差异作为输出值返回。LSQNOLIN 函数就是最小化这些差的平方和。

```
function    diff = fit_simp(x,X,Y)
% 此函数被 LSQNONLIN 调用
% x 是包含等式系数的向量
% X 与 Y 是作为操作数传递给 lsnonlin
A = x(1);
B = x(2);
C = x(3);
```

```
D = x(4);
E = x(5);
diff = A + B.*exp(C.*X)+D.*exp(E.*X)-Y;
```

下面的脚本是利用上面定义的 fit_simp.m 函数的一个例子：

```
% 定义你打算拟合的数据集合
>> X=0:.01:.5;
>> Y=2.0.*exp(5.0.*X)+3.0.*exp(2.5.*X)+1.5.*rand(size(X));
% 初始化方程系数
>> X0=[1 1 1 1 1]';
% 设置用中等模式（memdium-scale）算法
>> options=optimset('Largescale','off');
 % 通过调用 LSQNONLIN 重现计算新的系数
>> x=lsqnonlin(@fit_simp,X0,[],[],options,X,Y);
% 调用 LSQNONLIN 结果输出表明拟合是成功的
Optimization terminated successfully:
 Gradient in the search direction less than tolFun
 Gradient less than 10*(tolFun+tolX)
% 绘制原始数据与新的计算的数据
>> Y_new=x(1)+x(2).*exp(x(3).*X)+x(4).*exp(x(5).*X);
>> plot(X,Y,'+r',X,Y_new,'b');
```



注意：LSQNONLIN 只可以处理实数变量。在处理包括复数变量的实例的拟合的时候，数据集应该被切分成实数与虚数部分。下面给出一个例子演示如何对复数参数进行最小二乘拟合。

为了拟合复数变量，你需要将复数分解为实数部分与虚数部分，然后把他们传递到函数中去，这个函数被 LEASTSQ 作为单个输入调用。首先，将复数分解为实部与虚部两个向量。其次，将这两个向量理解成诸如第一部分是实部、第二部分是虚部。在 MATLAB 函数中，重新装配复数数据，并用你想拟合的复数方程计算。将输出向量分解实部与虚部，将这两部分连接为一个单一的输出向量传递回 LEASTSQ。下面，给出一个例子演示如何根据两个复数指数拟合实数 X 与 Y。

建立方程：

```
function zero = fit2(x,X,Y)
% 根据输入 x 重建复数输入
cmpx = x(1:4)+i.*x(5:8);
% 利用复数计算函数
zerocomp = cmpx(1).*exp(cmpx(2).*X) + cmpx(3).* exp(cmpx(4).*X)-Y;
% 将结果转换成一个列向量
% 其中第一部分是实部，第二部分是虚部
numx = length(X); % 实部长度
zero=real(zerocomp); %实部
zero(numx+1:2*numx)=imag(zerocomp); % 虚部
```

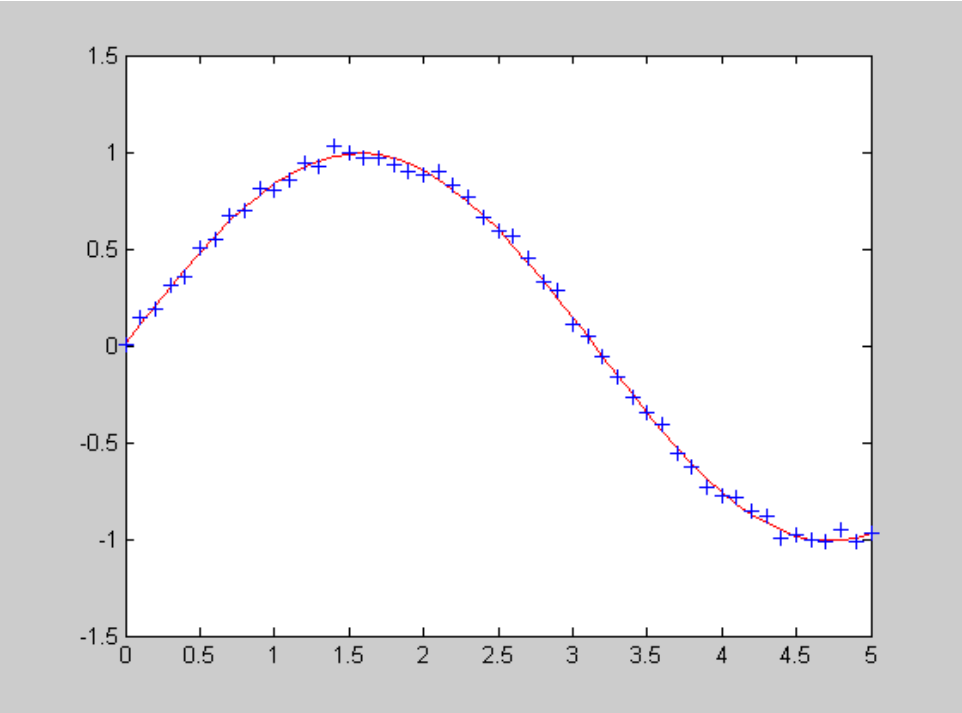
为了评价计算这个函数，需要 X 与 Y 数据集。LSQNONLIN 将根据它拟合出下面方程中的参数 a，b，c 与 d：

```
Y = a*exp(b*X)+c*exp(d*X);
```

其中，a，b，c 与 d 是复数变量。

```
>> X=0:.1:5;
>> Y=sin(X);
>> Y=Y+.1*rand(size(Y))-.05;
>> cmpx0=[1 i 2 2*i];
>> x0(1:4)=real(cmpx0);
>> x0(5:8)=imag(cmpx0);
>> x=leastsq(@fit2,x0,[],[],X,Y);
>> cmpx=x(1:4)+i.*x(5:8);
>> Y1=real(cmpx(1).*exp(cmpx(2).*X)+cmpx(3).*exp(cmpx(4).*X));
```

```
>> plot(X,Y1,'r');
>> hold on
>> plot(X,Y,'+');
```



LSQCURVEFIT：利用此函数可以在最小二乘意义上解决非线性曲线拟合（数据拟合）问题。也就是说，给定输入数据 xdata，以及观测的输出数据 ydata，找到系数 x，使得函数 F(x，xdata)能够最好的拟合向量值。LSQCURVEFIT 利用与 LSQNONLIN 相同的算法。它的目的在于专门为数据拟合问题提供一个接口。

在拟合的时候，2 维、3 维或者 N 维参数拟合是没有什么差别的。下面给出一个 3 维参数拟合的例子。待拟合函数是：

```
z = a1*y.*x..^2+a2*sin(x)+a3*y.^3;
```

建立的 myfun.m 的函数如下：

```
function F = myfun(a, data);
x = data(1,:);
y = data(2,:);
F= a(1)*y.*x.^2+a(2)*sin(x)+a(3)*y.^3;
```

下面的脚本展示了这么利用上面的函数：

```
>> xdata= [3.6 7.7 9.3 4.1 8.6 2.8 1.3 7.9 10.0 5.4];
 >> ydata= [16.5 150.6 263.1 24.7 208.5 9.9 2.7 163.9 325.0 54.3];
 >> zdata= [95.09 23.11 60.63 48.59 89.12 76.97 45.68 1.84 82.17 44.47];
 >> data=[xdata; ydata];
 >> a0= [10, 10, 10]; %  初识揣测
 >> [a, resnorm] = lsqcurvefit(@myfun,a0,data,zdata)
      Maximum number of function evaluations exceeded;
       increase options.MaxFunEvals
      a = 0.0088    -34.2886      -0.0000
      resnorm = 2.2636e+004
      >> format long
      >> a
      a =    0.00881645527493 -34.28862491919983    -0.00000655131499
      >> option=optimset('MaxFunEvals',800);
      >> [a, resnorm] = lsqcurvefit(@myfun,a0, data, zdata, [], [], option)
      Optimization terminated successfully:
       Relative function value changing by less than OPTIONS.TolFun
      a =    0.00740965259653 -20.21201417111138    -0.00000502014964
      resnorm = 2.195886958305428e+004
```

统计工具箱函数

函数名	描 述
nlinfit（非线性回归）	采用 Gauss-Newton 法进行非线性最小二乘数据拟合
lscov（线性回归）	根据已知协方差矩阵进行最小二乘估计
regress	多元线性回归
regstats	回归诊断
ridge	脊回归（？ Ridge regress）
rstool	多维响应表面可视化（RSV）
stepwise	交互式逐步回归

具体例子请参阅相应文档

第3节 加权曲线回归方法

当拟合数据包含随机变量时，通常在针对误差有两个主要假设：

- 误差只在响应数据中存在，而不在预测数据中存在；
- 误差是满足零均值常数方差正态分布的数据变量。

第二条假设方差为常数严重影响野点的出现。因为野点原来真实模式数据，他们可能引起真实拟合的潜在错误。为了改善拟合，你可以领用附加的尺度因子（也就是权值）来提高拟合的质量。将野点的权值设置得比较小，因此可以获得较好的拟合。

你可以利用下面的三个工具箱来实施加权拟合。

a. Curve Fitting Toolbox

此工具箱对线性与非线性数据有通用的最小二乘拟合能力。权值是操作数设置中的一部分，你可以通过 **fitoptions** 进行设置。在曲线拟合工具箱中，权值可以是与数据相联系的一个权向量。

b. Statistics Toolbox

统计工具箱中的 **robustfit** 函数具有加权自动回归的能力。robustfit 采用 robust 回归对数据进行拟合，其输出为回归系数。它也可以进行线性加权回归。调用语法结构是： ROBUSTFIT(X,Y,'WFUN',TUNE,'CONST')

c. Optimization Toolbox

你也可以利用优化工具箱的最小二乘求解器-----LSQNONLIN 和 LSQCURVEFIT 函数实施加权最小二乘拟合。为了利用 LSQNONLIN 和 LSQCURVEFIT 实现加权最小二乘拟合，你需要针对你的拟合数据建立起一个方程。你可以在 Solution 27840 中找到相应的例子。

附： Solution 27840

如何利用优化工具箱的 LSQNONLIN 函数实现加权最小二乘拟合

例子：

1. 制造准备拟合的样本数据。在本例子中，数据是叠加了噪声的负指数模型：

```
>> xdata=0:.01:1;
>> ydata=exp(-3*xdata)+randn(size(xdata))*.05;
```

2. 在图形中查看数据，即用 plot 画出 xdata 与 ydata，用蓝色的点表示：

```
>> plot(xdata,ydata,'b.');
```

3. 建立一个函数作为 LSQNONLIN 的输入，该函数包含你打算拟合的数据的方程。

建立名为 Mycurve.m 的 m 文件:

```
function err_weithted = mycurv(parameter, real_x,real_y)
% 想拟合的方程是: exp(parameter*xdata)
% 其中,parameter 是未知的
% fit = exp(parameter*real_x);
fit = exp(parameter*real_x);
err = fit -real_y;
% 对误差进行加权
% 在本例子中,,我们只对最后一项进行加权,并设置其他的项为 0(这将
%使得例子根据清楚)
weight = [zeros(1,length(real_x)-1) 1]; % 误差向量的权重
err_weithted = err.*weight;
```

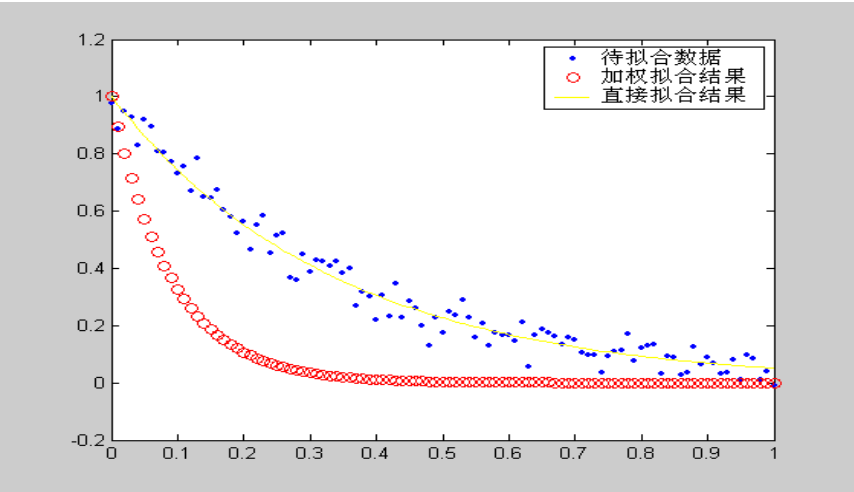
4. 调用优化程序,返回 parameter_hat 参数

```
>> parameter_hat = lsqnonlin(@mycurv,3,[],[],[],xdata,ydata)
  Optimization terminated successfully:
   First-order optimality less than OPTIONS.TolFun, and no
   negative/zero curvature detected
   parameter_hat = -11.18559914772636
```

parameter_hat 返回利用 m 文件 Mycurv 的指定方程的产生值。

5. 获取了拟合的方程式，将 parameter_hat 带入指数方程，并在同一张图上画出。

```
>> fitted = exp(parameter_hat*xdata); hold on
>>plot(xdata,fitted,'ro');
 >>parameter_hat2 = lsqnonlin(@mycurv,3,[],[],[],xdata,ydata) %不加权拟合
 >>fitted2 = exp(parameter_hat2*xdata);
 >>plot(xdata,fitted2,'y-');
 >>legend('待拟合数据','加权拟合结果','直接拟合结果');
```



比较拟合结果可以看出不加权时，曲线从所有数据点的中间穿过去（如黄色的线所示）。加权的时候，拟合曲线只通过最后一个点（因为只有该点权值非 0）。

第 4 节：利用 Curve Fitting Toolbox 改善拟合结果

很多因素对曲线拟合造成影响。下面列出各种提示有助于你提高拟合质量：

- 模型的选择：或者从 **MATLAB** 的模型库、或者用户自定义的模型的选择，是最主要的一个因素，试着用各种不同的模型对你的数据进行拟合比较；
- 数据预处理：在拟合前对数据进行预处理也很有用。这包括：
  - ▲ 对响应数据进行变换
  - ▲ 剔除 **Infs**，**NaNs**，以及野点

  更多的细节请查阅 **Curve Fitting Toolbox** 的文档。
- 合理的拟合应该具有处理出现奇异而使得预测趋于无穷大的时候的能力。

  具体细节请查阅 **Curve Fitting Toolbox** 的文档
- 如果你提供越多的系数的估计信息，你的拟合越容易收敛。下面的提示有利于你加快拟合的速度，提高拟合的精度：
  - ▲ 在开始拟合的时候进行一些智能的揣测。如果你认为系数可能是某些值，那么就用那些值作为起始值
  - ▲ 如果缺少起始值的信息，试着用大量不同的起始值
  - ▲ 尽量重复训练参数。例如，如果你知道参数必须是正的，那么设置它的下限是 **0** 使得循环拟合过程。
  - ▲ 调整各种拟合操作数，例如：
    - **a.** 采用不同的算法
    - **b.** 增加迭代与函数评价的次数
    - **c.** 减小容许误差
  - ▲ 将数据分解为几个子集，对不同的子集采用不同的曲线拟合
- 复杂的问题最好通过进化的方式解决，即一个问题的少量独立变量先解决。低阶问题的解通常通过近似映射作为高阶问题解的起始点。例如，如果你的模型最好被描述为：

  ```
  y = c + a * exp(b*x) +d * sin(f*x)
  ```

  那么它经常最好每次拟合一个项，通常从最重要的项开始。你可以先拟合：

  ```
  y = c1 + a1 * exp(b1*x)
  ```

  然后利用拟合出的结果系数作为上式中的 **a**，**b**，**c** 的起始值进行完整的拟合。

第 5 节 ： 其他相近方法

或许 MATLAB 2002 年 2 月的 News and Notes 杂志中有关曲线拟合的文章：Atmospheric Carbon Di-Oxide Modeling and the Curve Fitting Toolbox 或许对你有用。

附：一个用 fminsearch 进行园的拟合的例子：

```
function [xc,yc,r,fval,exitflag,output] = tlscirc(x,y)
% TLS circle fit
options=optimset('TolFun',1e-10);
[x0,y0]=circfit(x,y); % center estimate
[z,fval,exitflag,output]=fminsearch(@circobj,[x0,y0],options,x,y);
[f,r]=circobj(z,x,y); xc=z(1); yc=z(2);

function [f,r] = circobj(z,x,y)
% TLS circle fit objective f and radius r
n=length(x);
X=x-z(1); Y=y-z(2); X2=X'*X; Y2=Y'*Y;
r=sum(sqrt(X.^2+Y.^2))/n; f=X2+Y2-n*r^2;

The circfit function is used above as an initial estimator. The
objective values of tlscirc and circfit differ by nearly 14% in the
example below.

x=[1;2;3;5;7;9]; y=[7;6;7;8;7;5];
plot(x,y,'bo'), hold on

[xc,yc,r,f]=tlscirc(x,y)
% [xc,yc,r,f]=[4.7398, 2.9835, 4.7142, 1.2276]
rectangle('Position', [xc-r,yc-r,2*r,2*r], 'Curvature', [1,1],'EdgeColor','b')

[xc,yc,r] = circfit(x,y)
X=x-xc; Y=y-yc; f=sum((sqrt(X.^2+Y.^2)-r).^2)
% [xc,yc,r,f]=[4.7423, 3.8351, 4.1088, 1.3983]
rectangle('Position', [xc-r,yc-r,2*r,2*r], 'Curvature', [1,1],'EdgeColor','r')

hold off, axis equal
```

Genial @ USTC

2004-4-17