

## 利用 Matlab 提取图片中的数据(修订版)

从事科研或者工程的人员在文档撰写过程中，常需要将文献中的曲线与自己的结果对比，为获取原始数据，一种常用的办法是手动描点，即将原始曲线放大然后打印出来，选取一定数量的点，读出其横纵坐标，然后重绘。对于较为平坦的曲线，这种方法当然可行，但当曲线数量增加，曲线变化复杂，这种方法工作量可想而知。前段时间由于原始数据丢失，仅剩几十幅图片，本人尝试过手动描点，经历几个小时奋战，实在无法继续，索性转向 matlab，借助其强大的数据处理能力，编写了两个 GUI 的小软件 image2data、data\_poly 提取数据，如今大功告成，遂于大家分享。

2010-12-26

[yc97463240@126.com](mailto:yc97463240@126.com)

2010 年编写软件至今，已有一年，平均每个星期都有网友向我索要软件或提出问题，与你们的交流很愉快，为答谢大家对我的鼓励，我决定：

**1、对本教程作出修改，针对大家容易出现错误的地方作出批注，并在一个月内对程序进行改进。**

**2、录制视频教程，更加直接的引导大家使用。**

最后申明，本教程完全免费，可在百度、豆丁和其他文档分享类网站在线阅读，教程会发布到优酷上，如有需要，直接 email 索取(一般当天回复)。

2012-03-10

[yc97463240@126.com](mailto:yc97463240@126.com)

本文分三部分：1、数据提取实验演示；2、软件编写要点；3、附录。关心使用操作的朋友请看第一部分，有兴趣钻研代码的朋友请看第二部分，本软件以功能实现为主，界面和操作方面略有不适，敬请谅解。

## 1、数据提取演示实验

**原始数据来源：**安华高科技数据手册（HSMP-38XX and HSMP-48XX Series），如图 1 所示。

**目标曲线：**提取 1MHz 频率下的 PIN 二极管电容与反偏电压之间关系曲线。

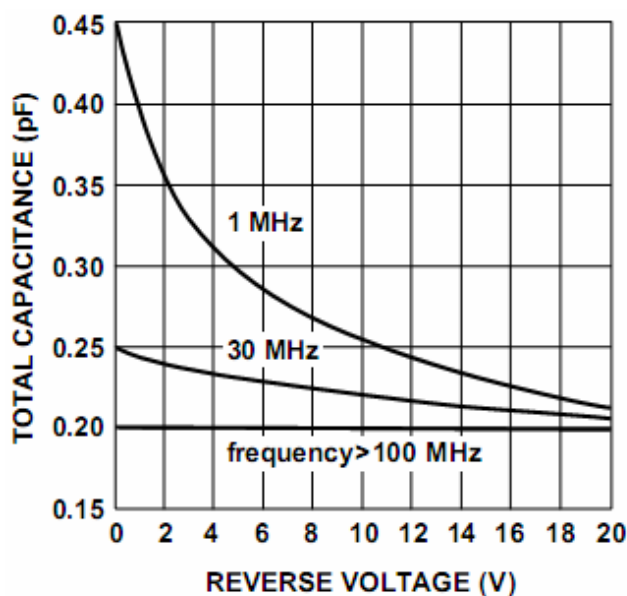


Figure 1. RF Capacitance vs. Reverse Bias, HSMP-3810 Series.

图 1 包含目标曲线的原始图像

**实验步骤：**

### Step1: 制作 jpg 图片

方法：利用 PPT 的图片另存功能制作 jpg 图片，本方法具有普遍性。

首先从 pdf 中 copy 图 1，即上述包含数据坐标的曲线，然后 paste 到 ppt 的空白页面，最后调整图片大小，另存为 figure1.jpg，如图 2 所示。

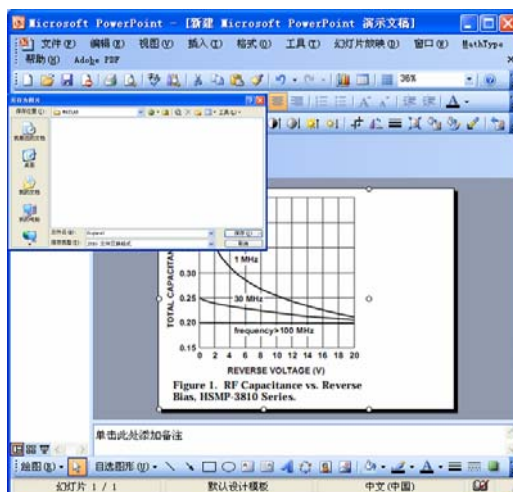


图 2 利用 ppt 制作 jpg 格式图像

### Step2: 导入图片, 填写相关参数

首先运行 `image2data.m` 或 `image2data.fig` (程序采用 Matlab R2008b 编写), 填写坐标轴的最大最小值, 然后在

`import` 按钮下的文本框输入文件名 `figure1`(默认 `jpg` 后缀), 最后点解 `import` 按钮导入图片, 如图 3 所示。

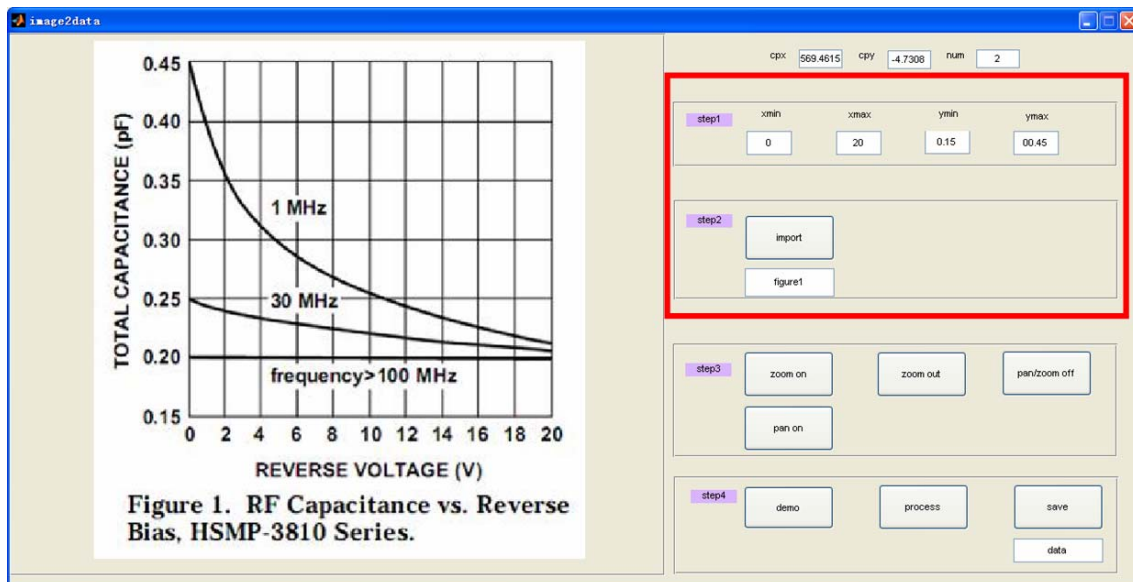


图 3 导入图像并输入相关参数

#### 特别关注:

- 1、程序版本, 如果采用 Matlab R2008b 以外的版本可能出现不兼容的问题, 如果运行出现问题请换用 R2008b。
- 2、文件路径, Matlab 每次启动后会自动进入启动路径(startup directory), 而程序编写时采用相对路径, 所以请将 `jpg` 文件和程序 `copy` 到当前路径对应的文件夹(current directory), 或者更改当前路径为 `jpg` 文件和程序所在的文件夹, 否则会出现未知错误。

### Step3: 坐标轴标定(关键)

状态栏介绍: `cpx` 显示当前取点横坐标, `cpy` 显示当前取点纵坐标, 两者均为相对值, `num` 显示当前鼠标取点总数目 (注意, 初始点数为 2, 然后存处 4 个坐标轴标定坐标, 剩余用来存储曲线坐标)。

按钮功能介绍: `zoom on` 按钮 → 图像放大, `zoom out` 按钮 → 恢复初始显示大小, `pan on` 按钮 → 鼠标拖动图像, `pan on/off` 按钮 → 退出放大或者拖动操作模式, `空格键` → 取点操作, `delete` 键 → 删除最近一次取点数据。

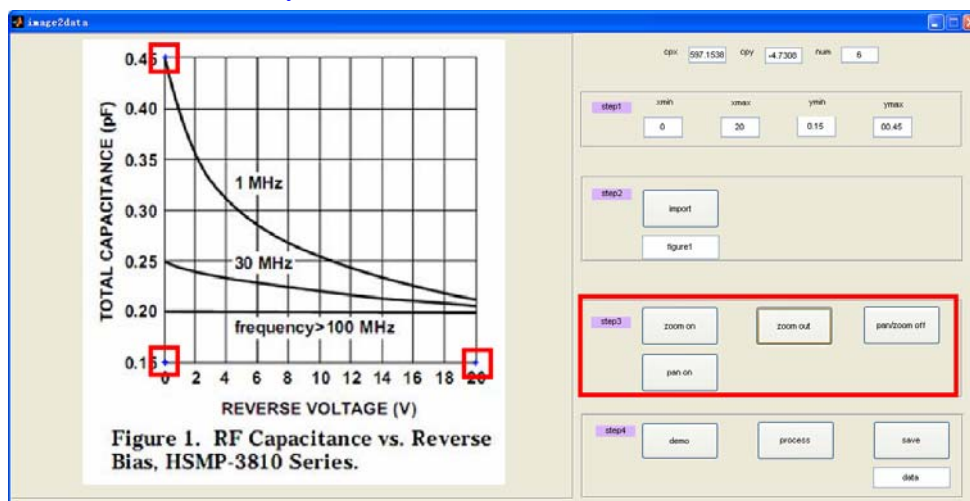


图 4 坐标轴定标

坐标标定：本次需要标定四个坐标值，顺序依次为 Ymax→Ymin→Xmin→Xmax。操作时首先按下 **zoom out** 按钮恢复原始图像显示，然后利用 **zoom on** 按钮放大图像、**pan on** 按钮平移图像到合适位置后，按下 **pan on/off** 按钮退出放大或者拖动操作模式，最后按下 **空格键** 取点即可。如果操作失误，按下 **delete** 键可删除最近一次取点数据。

### 特别关注：

- 1、每次放大或者拖动操作完毕后，必须按下 **pan on/off** 按钮，才能用 **空格键** 进行取点操作。
- 2、按下 **zoom on** 按钮后，还可以用鼠标中间的滚轮放大缩小图片，十分方便。
- 3、操作完毕，确认 **num** 显示为 6 后方可进行曲线描点。

### Step4: 曲线描点(关键)

按照上述操作反复进行图像放大、拖动、取点操作，取点时状态栏的 **cpX** 和 **cpY** 会显示当前相对坐标（注意，这个坐标图像坐标，y 轴方向向下，后面坐标变换需要考虑）。根据需要确定取点数目，图 5 给出了描点完毕后的曲线，可以看出取点基本代表了曲线的全部信息。

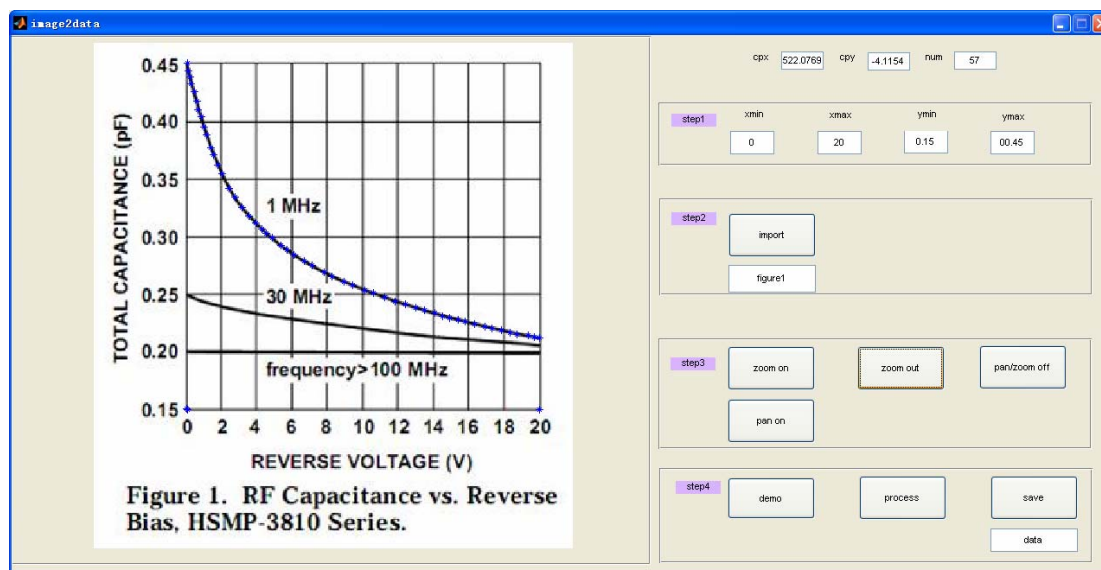


图 5 曲线描点

### 特别关注：

- 1、每次放大或者拖动操作完毕后，必须按下 **pan on/off** 按钮，否则直接按下 **空格键** 会弹出 **matlab** 命令框。

2、曲线取点请按照先后顺序依次进行，程序中未对所有数据点重新排序。

### Step5: 数据处理及存储

首先按下 `demo` 按钮，描点后的曲线会显示在图像当中，如图 6 所示；然后按下 `process` 按钮，程序自动进行坐标转换(相对坐标→绝对坐标)；最后在 `save` 按钮下的文本框中对数据文本命名 `data`(不需要后缀，默认 txt 文本)，程序会将数据存入 `data.txt` 文件中。

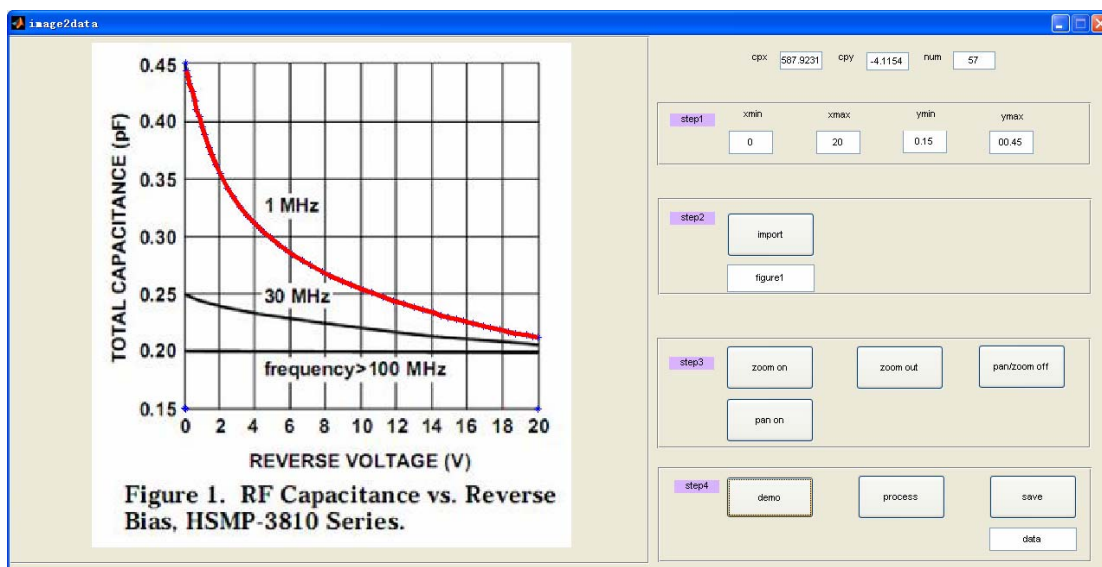


图 6 拟合曲线效果

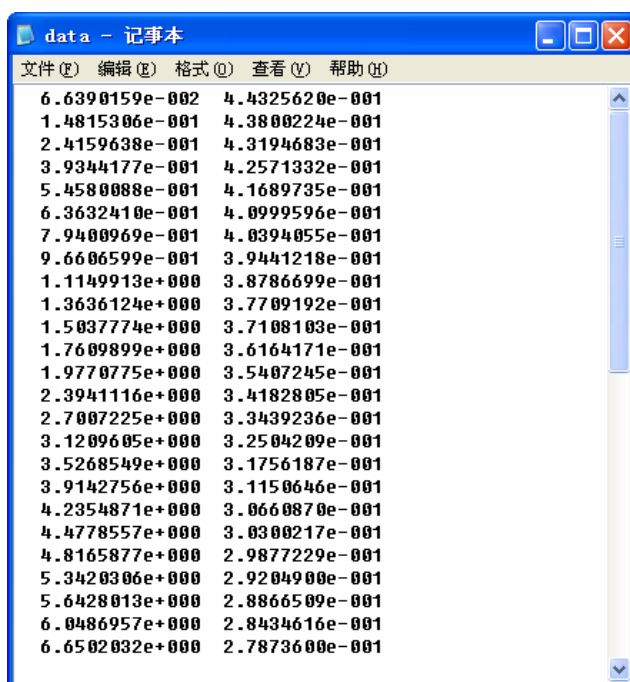


图 7 数据记录文本

### Step6: 数据后处理

由于 `data.txt` 文本中数据是手动选取的，分布不够均匀，直接使用不太方便，所以我们需要通过数据拟合，重新采样得到等间距的数据。运行 `data_poly.m` 或者 `data_poly.fig`，得到图 8 所示的界面，该程序内置多项式拟合，其中 `nth` 代表多项式拟合的阶数（一般 6 就够用了），`num` 代表重新采样的数据个数，`xmin`、`ymin`、`xmax`、`ymax` 表示需要输出数据的横纵坐标范围，程序默认输入文本为 `data.txt`，输出

文本为 `ndata.txt`，数据拟合结果如下图所示。

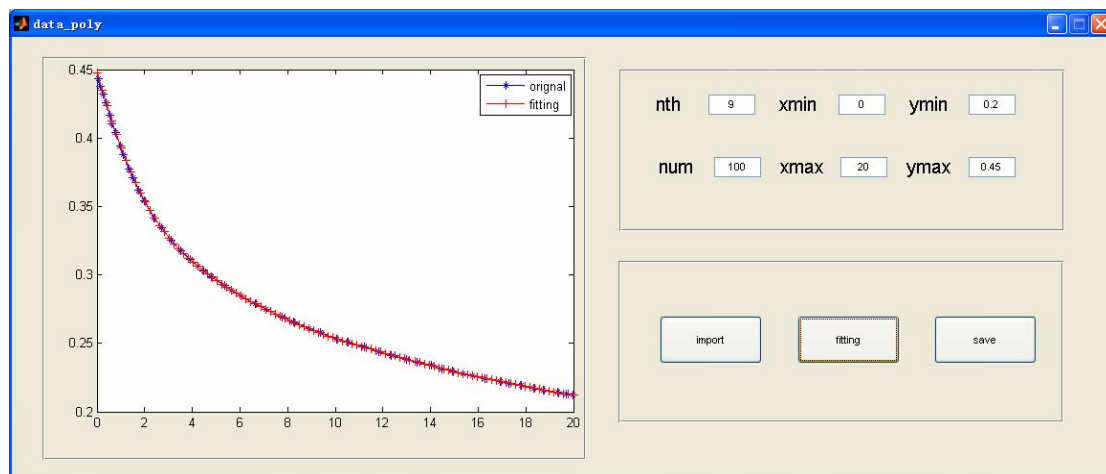


图 8 数据拟合再采样结果

可能大家会问，为什么两个程序不合在一起呢？原因在于数据拟合包含的种类很多，最基本的为多项式拟合，其他的有指数、三角函数等等，针对不同的曲线，所采取的拟合函数不尽相同，matlab 中的 `cftool` 工具箱包含了很多的拟合函数，当 `data_poly` 拟合效果不好时，大家可以试试 `cftool` 工具。

至此，数据提取过程完毕，只要大家按照步骤细心操作，提取曲线的速度和获得数据的精度应该能让你满意。

## 2、软件编写要点

这两个小软件从构思到完成大概用了两天，功能的完成绝大部分归功于 matlab 丰富的函数库和方便的 GUI，其中用到的主要函数主要有：`imread`, `imshow`, `findobj`, `num2str`, `strcat`, `load`, `save`, `get`, `set`。列表如下，更详细的介绍请参阅 `matlab help` 文档。

表 1 主要函数列表

Index	Function	Description
1	<code>findobj</code>	Locate graphics objects with specific properties
2	<code>get</code>	Query object properties
3	<code>imread</code>	Read image from graphics file
4	<code>imshow</code>	Display image
5	<code>load</code>	Load workspace variables from disk
6	<code>num2str</code>	Convert number to string
7	<code>save</code>	Save workspace variables to disk
8	<code>set</code>	Set object properties
9	<code>strcat</code>	Concatenate strings horizontally

除了上述函数的掌握之外，还需要对 matlab 的 GUI 数据结构和函数响应有一定的理解，在此就不多讲了（多看 `matlab help` 相关例程就明白了）。

编写程序之前，首先心中要有一个框架，做些什么，怎么做，顺序如何等。本软件的结构如下图所示：

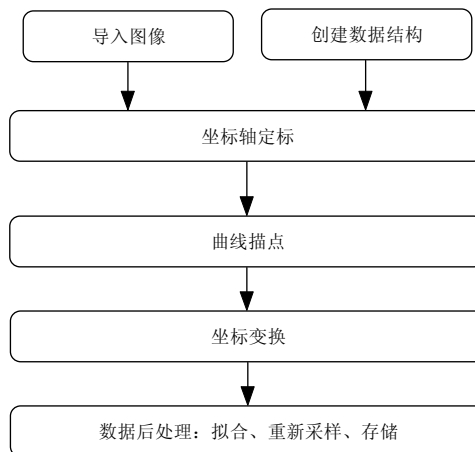


图9 程序流程示意

**导入图像程序段:**

```

h_temp=findobj('tag','edit8');
str_img=strcat(get(h_temp,'String'),'.jpg'); %get fullname of picture
image_temp=imread(str_img);
imshow(image_temp); %read and show
  
```

**创建数据结构程序段:**

```

handles.cpx=0;
handles.cpy=0;% gloable variables for current position of mouse
handles.idata=zeros(2,2);% the key variable for obtaining curve data
guidata(hObject, handles);
  
```

**坐标轴定标及坐标变换:**

```

%obtain input axis information
temp=str2num(get(findobj('tag','edit1'),'string'));
xmin=temp;
temp=str2num(get(findobj('tag','edit2'),'string'));
xmax=temp;
temp=str2num(get(findobj('tag','edit3'),'string'));
ymin=temp;
temp=str2num(get(findobj('tag','edit4'),'string'));
ymax=temp;
%save axis to global variable idata
idata(1:2,:)=[xmin,xmax;ymin,ymax];
idata=[idata;handles.idata(3:end,:)];

axis_p=idata(3:6,:);
data_p=idata(7:end,:); % class idata

xpmin=axis_p(3,1);
xpmax=axis_p(4,1);
ypmin=axis_p(1,2);
ypmax=axis_p(2,2);
  
```

```
%coordinate transform
data_new=data_p;
data_new(:,1)=data_p(:,1)-xpmin;
data_new(:,2)=ypmax-data_p(:,2);

data_new(:,1)=data_new(:,1)/(xpmax-xpmin);
data_new(:,2)=data_new(:,2)/(ypmax-ypmin);

data_new(:,1)=xmin+data_new(:,1)*(xmax-xmin);
data_new(:,2)=ymin+data_new(:,2)*(ymax-ymin);
%save results as a new variable
idata(7:end,:)=data_new;
handles.newidata=idata;
guidata(hObject,handles);
```

#### 数据后处理代码段:

```
%obtain input data
x=handles.data(:,1);
y=handles.data(:,2);
n=str2double(get(findobj('tag','edit1'),'string'));
np=str2double(get(findobj('tag','edit2'),'string'));
xmin=str2double(get(findobj('tag','edit3'),'string'));
xmax=str2double(get(findobj('tag','edit4'),'string'));
%data fitting and samlping
p=polyfit(x,y,n);
xnew=[xmin:(xmax-xmin)/(np-1):xmax]';
ynew=polyval(p,xnew);
%save results and plot curve
plot(x,y,'-*b',xnew,ynew,'-+r');
legend('original','fitting');
handles.ndata=[xnew,ynew];
guidata(hObject,handles);
```



### 3、附录

#### 3.1 描点数据列表

index	x	y	index	x	y
1	0.066390	0.443256	31	10.077438	0.253223
2	0.148153	0.438002	32	10.556335	0.250863
3	0.241596	0.431947	33	11.223789	0.247257
4	0.393442	0.425713	34	11.807809	0.243784
5	0.545801	0.416897	35	12.361811	0.241112
6	0.636324	0.409996	36	12.975033	0.238485
7	0.794010	0.403941	37	13.444189	0.236304
8	0.966066	0.394412	38	14.039891	0.233766
9	1.114991	0.387867	39	14.489587	0.231361
10	1.363612	0.377092	40	14.905877	0.229847
11	1.503777	0.371081	41	15.367253	0.227755
12	1.760990	0.361642	42	15.761467	0.226464
13	1.977078	0.354072	43	16.303302	0.224371
14	2.394112	0.341828	44	16.913604	0.221877
15	2.700723	0.334392	45	17.281537	0.220497
16	3.120961	0.325042	46	17.818457	0.218983
17	3.526855	0.317562	47	18.314875	0.216891
18	3.914276	0.311506	48	18.729530	0.215510
19	4.235487	0.306609	49	19.339940	0.214397
20	4.477856	0.303002	50	19.702032	0.212972
21	4.816588	0.298772	51	19.985282	0.212216
22	5.342031	0.292049			
23	5.642801	0.288665			
24	6.048696	0.284346			
25	6.650203	0.278736			
26	7.082379	0.274862			
27	7.784130	0.269030			
28	8.233826	0.265512			
29	8.896676	0.260792			
30	9.396013	0.257809			

#### 3.2 数据拟合结果

Linear model Poly9:

$$\text{fittedmodel1}(x) = p1*x^9 + p2*x^8 + p3*x^7 + p4*x^6 + p5*x^5 + p6*x^4 + p7*x^3 + p8*x^2 + p9*x + p10$$

Coefficients (with 95% confidence bounds):

$$p1 = 1.116e-010 \quad (5.79e-011, 1.653e-010)$$

$$p2 = -1.047e-008 \quad (-1.528e-008, -5.669e-009)$$

p3 = 4.105e-007 (2.304e-007, 5.905e-007)  
 p4 = -8.65e-006 (-1.231e-005, -4.986e-006)  
 p5 = 0.0001044 (6.051e-005, 0.0001483)  
 p6 = -0.0006919 (-0.001006, -0.0003775)  
 p7 = 0.001806 (0.0005058, 0.003107)  
 p8 = 0.005695 (0.002846, 0.008543)  
 p9 = -0.06124 (-0.06399, -0.05849)  
 p10 = 0.4474 (0.4466, 0.4482)

### 3.3、重新采样数据列表

index	x	y	index	x	y
1	0.000000	0.447368	51	10.101010	0.253098
2	0.202020	0.435242	52	10.303030	0.251962
3	0.404040	0.423655	53	10.505051	0.250856
4	0.606061	0.412661	54	10.707071	0.249775
5	0.808081	0.402291	55	10.909091	0.248718
6	1.010101	0.392561	56	11.111111	0.247681
7	1.212121	0.383475	57	11.313131	0.246660
8	1.414141	0.375021	58	11.515152	0.245653
9	1.616162	0.367182	59	11.717172	0.244658
10	1.818182	0.359932	60	11.919192	0.243671
11	2.020202	0.353240	61	12.121212	0.242691
12	2.222222	0.347073	62	12.323232	0.241715
13	2.424242	0.341392	63	12.525253	0.240744
14	2.626263	0.336160	64	12.727273	0.239776
15	2.828283	0.331337	65	12.929293	0.238810
16	3.030303	0.326887	66	13.131313	0.237846
17	3.232323	0.322771	67	13.333333	0.236886
18	3.434343	0.318955	68	13.535354	0.235930
19	3.636364	0.315405	69	13.737374	0.234980
20	3.838384	0.312088	70	13.939394	0.234036
21	4.040404	0.308978	71	14.141414	0.233101
22	4.242424	0.306048	72	14.343434	0.232177
23	4.444444	0.303273	73	14.545455	0.231266
24	4.646465	0.300635	74	14.747475	0.230370
25	4.848485	0.298113	75	14.949495	0.229491
26	5.050505	0.295693	76	15.151515	0.228630
27	5.252525	0.293362	77	15.353535	0.227789
28	5.454546	0.291107	78	15.555556	0.226970
29	5.656566	0.288921	79	15.757576	0.226173
30	5.858586	0.286796	80	15.959596	0.225397
31	6.060606	0.284727	81	16.161616	0.224642
32	6.262626	0.282708	82	16.363636	0.223907
33	6.464647	0.280738	83	16.565657	0.223189
34	6.666667	0.278815	84	16.767677	0.222486
35	6.868687	0.276937	85	16.969697	0.221795

36	7.070707	0.275104	86	17.171717	0.221111
37	7.272727	0.273316	87	17.373737	0.220432
38	7.474748	0.271574	88	17.575758	0.219752
39	7.676768	0.269878	89	17.777778	0.219069
40	7.878788	0.268228	90	17.979798	0.218379
41	8.080808	0.266625	91	18.181818	0.217681
42	8.282828	0.265070	92	18.383838	0.216974
43	8.484849	0.263562	93	18.585859	0.216264
44	8.686869	0.262101	94	18.787879	0.215555
45	8.888889	0.260687	95	18.989899	0.214861
46	9.090909	0.259318	96	19.191919	0.214198
47	9.292929	0.257993	97	19.393939	0.213591
48	9.494950	0.256711	98	19.595960	0.213076
49	9.696970	0.255470	99	19.797980	0.212698
50	9.898990	0.254266	100	20.000000	0.212516