样条插值实验

林元莘 521070910077

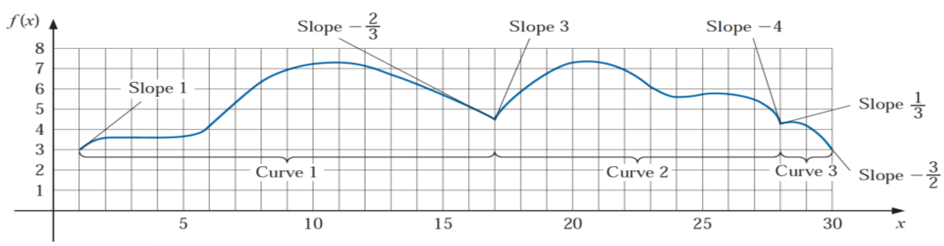
一、实验目的

1、利用三次样条插值和贝塞尔曲线作图

2、熟悉三次样条插值程序以及贝塞尔曲线程序

二、实验内容

1、根据表一中提供的数据用三次样条插值画出图一曲线

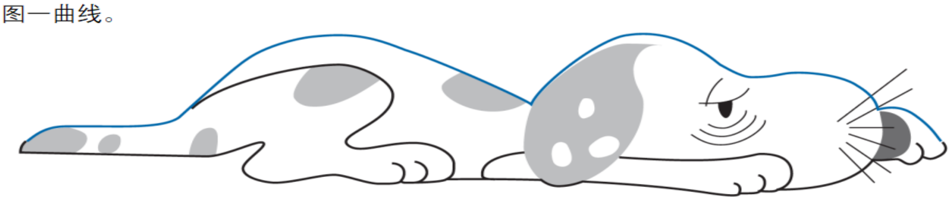


图一

Curve 1				Curve 2				Curve 3			
i	x_i	$f(x_i)$	$f'(x_i)$	i	x_i	$f(x_i)$	$f'(x_i)$	i	x_i	$f(x_i)$	$f'(x_i)$
0	1	3.0	1.0	0	17	4.5	3.0	0	27.7	4.1	0.33
1	2	3.7		1	20	7.0		1	28	4.3	
2	5	3.9		2	23	6.1		2	29	4.1	
3	6	4.2		3	24	5.6		3	30	3.0	−1.5
4	7	5.7		4	25	5.8					
5	8	6.6		5	27	5.2					
6	10	7.1		6	27.7	4.1	−4.0				
7	13	6.7									
8	17	4.5	−0.67								

表一

2、用三次样条插值得到的导数值控制贝塞尔曲线两个端点的斜率，用贝塞尔曲线画出图一曲线。



图二

3、通过在图上进行鼠标点击的方式用贝塞尔曲线画出上图轮廓，并用得到的数据进行三次样条插值画出类似轮廓。

三、实验原理

1、钳制三次样条插值

由于钳制三次样条中，人为定义了$S_1'(x_1) = v_1, S_{n-1}'(x_n) = v_n$。我们把额外的条件可以写作

$$2\delta_1 c_1 + \delta_1 c_2 = 3(\frac{\Delta_1}{\delta_1} - v_1)$$

$$\delta_{n-1} c_{n-1} + 2\delta_{n-1} c_n = 3(v_n - \frac{\Delta_{n-1}}{\delta_{n-1}})$$

对于原需求解的$Ac=r$,仅需$A(1,1:2)=[2\delta_1,\delta_1], A(n,n-1:n)=[\delta_{n-1},2\delta_{n-1}]$;

$r(1)=3(\frac{\Delta_1}{\delta_1}-v_1), r(n)=3(v_n-\frac{\Delta_{n-1}}{\delta_{n-1}})$。表现在程序中则为

```
function coeff=splinecoeff(x,y,v)%计算钳制三次样条系数
%输入：数据点的 x,y 向量
%     以及两个斜率的向量 v
%输出：系数矩阵
n=length(x);
v1=v(1);
vn=v(2);
A=zeros(n,n);%矩阵 A 是 n*n
r=zeros(n,1);
for i=1:n-1%定义 delta
    dx(i)=x(i+1)-x(i);dy(i)=y(i+1)-y(i);
end
for i=2:n-1%加载 A 矩阵
    A(i,i-1:i+1)=[dx(i-1),2*(dx(i-1)+dx(i)),dx(i)];
    r(i)=3*(dy(i)/dx(i)-dy(i-1)/dx(i-1));%右端端点
end
%设置端点条件
A(1,1:2)=[2*dx(1) dx(1)];r(1)=3*(dy(1)/dx(1)-v1);%钳制条件
A(n,n-1:n)=[dx(n-1),2*dx(n-1)];r(n)=3*(vn-dy(n-1)/dx(n-1));
coeff=zeros(n,3);
coeff(:,2)=A\r;%求解系数 c
for i=1:n-1%求解 b 和 d
    coeff(i,3)=(coeff(i+1,2)-coeff(i,2))/(3*dx(i));
    coeff(i,1)=dy(i)/dx(i)-dx(i)*(2*coeff(i,2)+coeff(i+1,2))/3;
end
coeff=coeff(1:n-1,1:3);
end

function [x1,y1]=splineplot(x,y,v,k)%画出钳制三次样条插值函数图像
%输入：数据点的 x,y 向量，每段画出的点数 k
%     以及两个斜率的向量 v
%输出：在画出点上的 x1,y1 样条值
n=length(x);
coeff=splinecoeff(x,y,v);
x1=[];y1=[];
for i=1:n-1
    xs=linspace(x(i),x(i+1),k+1);
    dx=xs-x(i);
    ys=coeff(i,3)*dx;%使用嵌套算法求值
    ys=(ys+coeff(i,2)).*dx;
    ys=(ys+coeff(i,1)).*dx+y(i);
    x1=[x1;xs(1:k)'];y1=[y1;ys(1:k)'];
end
x1=[x1;x(end)];y1=[y1;y(end)];
plot(x,y,'o',x1,y1)
end
```

2、贝塞尔曲线

给定端点$(x_1,y_1),(x_4,y_4)$，控制点$(x_2,y_2),(x_3,y_3)$，设

$$b_x = 3(x_2 - x_1); b_y = 3(y_2 - y_1)$$
$$c_x = 3(x_3 - x_2) - b_x; c_y = 3(y_3 - y_2) - b_y$$
$$d_x = x_4 - x_1 - b_x - c_x; d_y = y_4 - y_1 - b_x - c_y$$

定义在$0 \le t \le 1$的贝塞尔曲线如下：

$$x(t) = x_1 + b_x t + c_x t^2 + d_x t^3$$
$$y(t) = y_1 + b_y t + c_y t^2 + d_y t^3$$

由其性质可知，$x'(0) = 3(x_2 - x_1); y'(0) = 3(y2 - y1); x'(1) = 3(x_4 - x_3); y'(1) = 3(y_4 - y_3)$

则可得到：$y_x|_{x=x_1} = \dfrac{y_2 - y_1}{x_2 - x_1}, y_x|_{x=x_4} = \dfrac{y_4 - y_3}{x_4 - x_3}$，即切线斜率可以通过端点与控制点求出。相关程序如下

```
function [X,Y,k]=bezierdraw%通过鼠标点击画出相关贝塞尔曲线并获取所点坐标与样条点处导数值
%在 MATLAB 的图形窗口中点击得到第一个点
%然后再点击三次得到两个控制点和另外一个样条点
%然后继续以每次三次点击得到三个点，给曲线增加更多段
%按下回车终止程序
plot([0 30],[0,0],'k',[0 0],[0 8],'k');hold on
axis on;%设置坐标轴
set(gca,'fontsize',8,'xminortick','on');
set(gca,'XTick',0:1:30);
grid on;%设置坐标轴网格
t=0:.02:1;
[x,y]=ginput(1);%进行一次鼠标点击
X(1)=x;
Y(1)=y;%保存第一次首端点
k=[];
while(0 == 0)
  [xnew,ynew] = ginput(3);  %另外三次点击
  if length(xnew) < 3
    break                  %按下回车键，终止
  end
  x=[x;xnew];y=[y;ynew];%画出样条点和控制点 pts
  k1=(y(2)-y(1))./(x(2)-x(1));
  k2=((y(4)-y(3))./(x(4)-x(3)));
  k=[k;k1;k2];%保存每次曲线首末端点导数值
  plot([x(1) x(2)],[y(1) y(2)],'r:',x(2),y(2),'rs');
  plot([x(3) x(4)],[y(3) y(4)],'r:',x(3),y(3),'rs');
  plot(x(1),y(1),'bo',x(4),y(4),'bo');
  bx=3*(x(2)-x(1)); by=3*(y(2)-y(1)); %样条方程
  cx=3*(x(3)-x(2))-bx;cy=3*(y(3)-y(2))-by;
  dx=x(4)-x(1)-bx-cx;dy=y(4)-y(1)-by-cy;
  xp=x(1)+t.*(bx+t.*(cx+t*dx));    %霍纳方法
  yp=y(1)+t.*(by+t.*(cy+t*dy));
  plot(xp,yp,'b')              %画出样条曲线
  x=x(4);y=y(4); %将上段最后一点转化为第一点，继续
  X=[X;x];Y=[Y;y];%保存每次曲线末端点
end
hold off
end

function bezierplot(x,y,k)%通过端点、控制点、导数值，画出某条贝塞尔曲线
%输入：首末数据点 x,y 向量，首末数据点处导数值 k
```

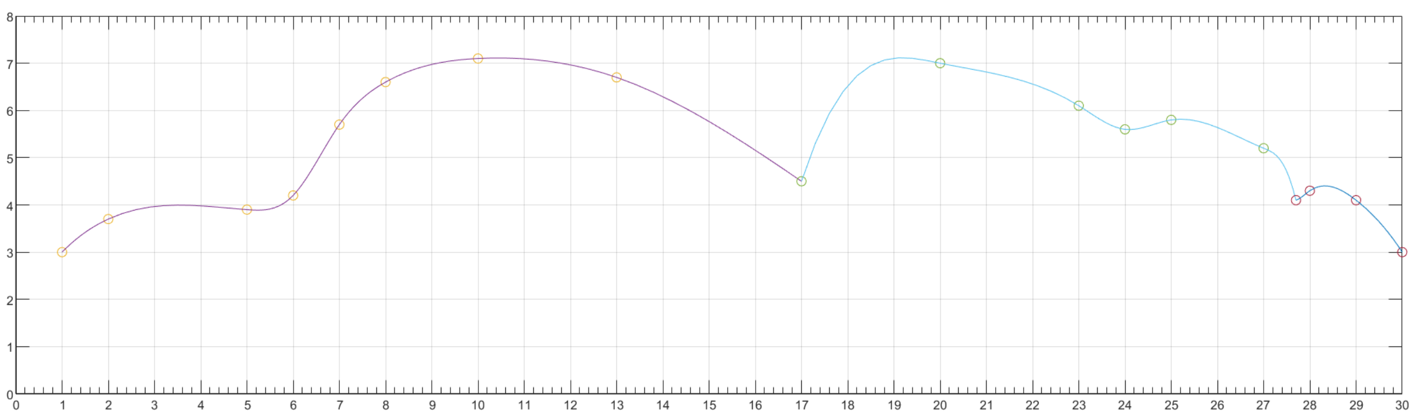
```
%输出：画出该对点间对应的贝塞尔曲线
  t=0:0.01:1;
  r=0.6;
  x=[x(1),x(1)+r./sqrt(k(1)^2+1),x(2)-r./sqrt(k(2)^2+1),x(2)];y=[y(1),y(1)+k(1)*r./sqrt(k(1)^2+1),y(2)-k(2)*r.
  %为了控制平滑程度，通过调试参数，让控制点与样条点距离保持 0.6 较好
  plot([x(1) x(2)],[y(1) y(2)],'r:',x(2),y(2),'rs');hold on
  plot([x(3) x(4)],[y(3) y(4)],'r:',x(3),y(3),'rs');hold on
  plot(x(1),y(1),'bo',x(4),y(4),'bo');hold on%画出样条点以及控制点
  bx=3*(x(2)-x(1)); by=3*(y(2)-y(1)); %样条方程
  cx=3*(x(3)-x(2))-bx;cy=3*(y(3)-y(2))-by;
  dx=x(4)-x(1)-bx-cx;dy=y(4)-y(1)-by-cy;
  xp=x(1)+t.*(bx+t.*(cx+t*dx));    % 霍纳方法
  yp=y(1)+t.*(by+t.*(cy+t*dy));
  plot(xp,yp,'b');hold on             %画出样条曲线
end
```

四、程序实现以及问题求解

1、第一题代入数据直接通过函数生成即可

```
format long
x1=[1,2,5,6,7,8,10,13,17];
y1=[3,3.7,3.9,4.2,5.7,6.6,7.1,6.7,4.5];
v11=[1,-0.67];
x2=[17,20,23,24,25,27,27.7];
y2=[4.5,7.0,6.1,5.6,5.8,5.2,4.1];
v22=[3,-4];
x3=[27.7,28,29,30];
y3=[4.1,4.3,4.1,3.0];
v33=[0.33,-1.5];   %输入点横纵坐标以及端点斜率
plot([0 30],[0,0],'k',[0 0],[0 8],'k');hold on
axis on;%设置坐标轴
set(gca,'fontsize',8,'xminortick','on');
set(gca,'XTick',0:1:30);
set(gcf,'unit','normalized','position',[0,0,0.9,0.3]);
grid on;%设置坐标轴网格
splineplot(x1,y1,v11,10);hold on
splineplot(x2,y2,v22,10);hold on
splineplot(x3,y3,v33,10);%画出钳制三次样条

function coeff=splinecoeff(x,y,v)%计算钳制三次样条系数
%输入：数据点的 x,y 向量
%     以及两个斜率的向量 v
%输出：系数矩阵
n=length(x);
v1=v(1);
vn=v(2);
A=zeros(n,n);%矩阵 A 是 n*n
r=zeros(n,1);
for i=1:n-1%定义 delta
    dx(i)=x(i+1)-x(i);dy(i)=y(i+1)-y(i);
end
for i=2:n-1%加载 A 矩阵
    A(i,i-1:i+1)=[dx(i-1),2*(dx(i-1)+dx(i)),dx(i)];
    r(i)=3*(dy(i)/dx(i)-dy(i-1)/dx(i-1));%右端端点
```

```
end
%设置端点条件
A(1,1:2)=[2*dx(1) dx(1)];r(1)=3*(dy(1)/dx(1)-v1);%钳制条件
A(n,n-1:n)=[dx(n-1),2*dx(n-1)];r(n)=3*(vn-dy(n-1)/dx(n-1));
coeff=zeros(n,3);
coeff(:,2)=A\r;%求解系数 c
for i=1:n-1%求解 b 和 d
    coeff(i,3)=(coeff(i+1,2)-coeff(i,2))/(3*dx(i));
    coeff(i,1)=dy(i)/dx(i)-dx(i)*(2*coeff(i,2)+coeff(i+1,2))/3;
end
coeff=coeff(1:n-1,1:3);
end

function [x1,y1]=splineplot(x,y,v,k)
%输入：数据点的 x,y 向量，每段画出的点数 k
%      以及两个斜率的向量 v
%输出：在画出点上的 x1,y1 样条值
n=length(x);
coeff=splinecoeff(x,y,v);
x1=[];y1=[];
for i=1:n-1
    xs=linspace(x(i),x(i+1),k+1);
    dx=xs-x(i);
    ys=coeff(i,3)*dx;%使用嵌套算法求值
    ys=(ys+coeff(i,2)).*dx;
    ys=(ys+coeff(i,1)).*dx+y(i);
    x1=[x1;xs(1:k)'];y1=[y1;ys(1:k)'];
end
x1=[x1;x(end)];y1=[y1;y(end)];
plot(x,y,'o',x1,y1)
end
```



2、用样条插值得到的导数值控制贝塞尔曲线导数值，用贝塞尔曲线画图一。

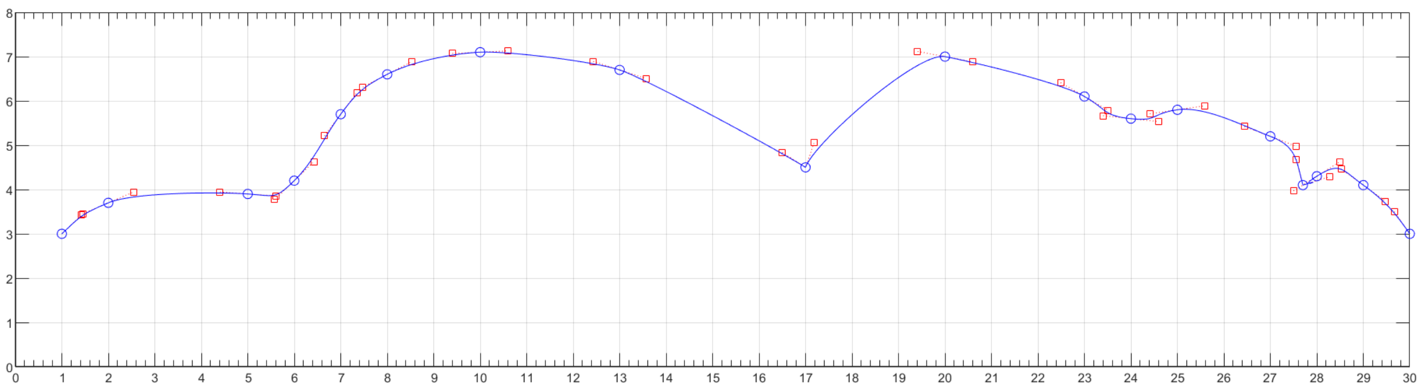
利用我们得到的贝塞尔曲线的导数公式$y_x|_{x=x_1}=\dfrac{y_2-y_1}{x_2-x_1}, y_x|_{x=x_4}=\dfrac{y_4-y_3}{x_4-x_3}$，只要给定控制点与样条点距离，便可得到控制点横纵坐标，进而求解该段贝塞尔曲线。

通过尝试，得到控制点与样条点距离大约为 0.6 时，图像看起来较为平滑，无过于突出的点。

```
format long
x1=[1,2,5,6,7,8,10,13,17];
```

```
y1=[3,3.7,3.9,4.2,5.7,6.6,7.1,6.7,4.5];
x2=[17,20,23,24,25,27,27.7];
y2=[4.5,7.0,6.1,5.6,5.8,5.2,4.1];
x3=[27.7,28,29,30];
y3=[4.1,4.3,4.1,3.0];
v11=[1,-0.67];
v22=[3,-4];
v33=[0.33,-1.5];%输入点与导数值
A1=splinecoeff(x1,y1,v11);%获取三次样条系数矩阵
k1=[(A1(:,1))',v11(2)];%获取每点上的样条导数值
plot([0 30],[0,0],'k',[0 0],[0 8],'k');hold on
axis on;%设置坐标轴
set(gca,'fontsize',8,'xminortick','on');
set(gca,'XTick',0:1:30);
set(gcf,'unit','normalized','position',[0,0,0.9,0.3]);
grid on;%设置坐标轴网格
for i=1:(length(k1)-1)%将点进行分对画图
    k=[k1(i),k1(i+1)];%首末导数值
    x=[x1(i),x1(i+1)];%首末坐标值
    y=[y1(i),y1(i+1)];
    bezierplot(x,y,k);hold on%画出该对点的贝塞尔曲线
end
%对其他两组数据进行相同操作
A2=splinecoeff(x2,y2,v22);
k2=[(A2(:,1))',v22(2)];
for i=1:(length(k2)-1)
    k=[k2(i),k2(i+1)];
    x=[x2(i),x2(i+1)];
    y=[y2(i),y2(i+1)];
    bezierplot(x,y,k);hold on
end
A3=splinecoeff(x3,y3,v33);
k3=[(A3(:,1))',v33(2)];
for i=1:(length(k3)-1)
    k=[k3(i),k3(i+1)];
    x=[x3(i),x3(i+1)];
    y=[y3(i),y3(i+1)];
    bezierplot(x,y,k);hold on
end

function bezierplot(x,y,k)%画出贝塞尔曲线
%输入：首末数据点 x,y 向量，首末数据点处导数值 k
%输出：画出该对点间对应的贝塞尔曲线
  t=0:0.01:1;
  r=0.6;
  x=[x(1),x(1)+r./sqrt(k(1)^2+1),x(2)-r./sqrt(k(2)^2+1),x(2)];y=[y(1),y(1)+k(1)*r./sqrt(k(1)^2+1),y(2)-k(2)*r.
  %为了控制平滑程度，让控制点与样条点距离保持 0.6
  plot([x(1) x(2)],[y(1) y(2)],'r:',x(2),y(2),'rs');hold on
  plot([x(3) x(4)],[y(3) y(4)],'r:',x(3),y(3),'rs');hold on
  plot(x(1),y(1),'bo',x(4),y(4),'bo');hold on%画出样条点以及控制点
  bx=3*(x(2)-x(1)); by=3*(y(2)-y(1)); %样条方程
  cx=3*(x(3)-x(2))-bx;cy=3*(y(3)-y(2))-by;
  dx=x(4)-x(1)-bx-cx;dy=y(4)-y(1)-by-cy;
  xp=x(1)+t.*(bx+t.*(cx+t*dx));    % 霍纳方法
  yp=y(1)+t.*(by+t.*(cy+t*dy));
  plot(xp,yp,'b');hold on             %画出样条曲线
end
```

```
function coeff=splinecoeff(x,y,v)%计算样条系数
%输入：数据点的 x,y 向量
%      以及两个斜率的向量 v
%输出：系数矩阵
n=length(x);
v1=v(1);
vn=v(2);
A=zeros(n,n);%矩阵 A 是 n*n
r=zeros(n,1);
for i=1:n-1%定义 delta
    dx(i)=x(i+1)-x(i);dy(i)=y(i+1)-y(i);
end
for i=2:n-1%加载 A 矩阵
    A(i,i-1:i+1)=[dx(i-1),2*(dx(i-1)+dx(i)),dx(i)];
    r(i)=3*(dy(i)/dx(i)-dy(i-1)/dx(i-1));%右端端点
end
%设置端点条件
A(1,1:2)=[2*dx(1) dx(1)];r(1)=3*(dy(1)/dx(1)-v1);%钳制条件
A(n,n-1:n)=[dx(n-1),2*dx(n-1)];r(n)=3*(vn-dy(n-1)/dx(n-1));
coeff=zeros(n,3);
coeff(:,2)=A\r;%求解系数 c
for i=1:n-1%求解 b 和 d
    coeff(i,3)=(coeff(i+1,2)-coeff(i,2))/(3*dx(i));
    coeff(i,1)=dy(i)/dx(i)-dx(i)*(2*coeff(i,2)+coeff(i+1,2))/3;
end
coeff=coeff(1:n-1,1:3);
end
```



3、用鼠标点出图上轮廓画出贝塞尔曲线，再用得到数据画出三次样条插值

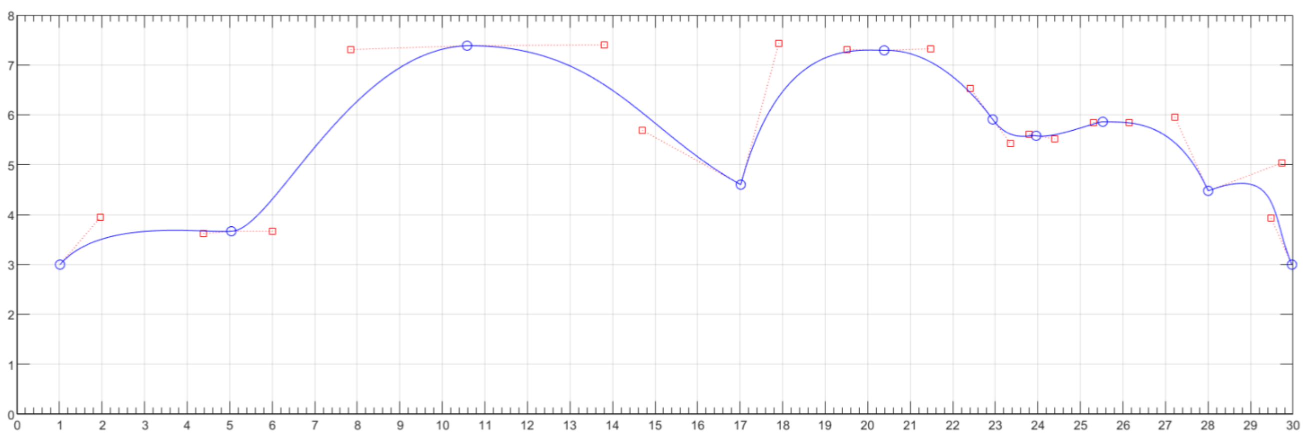
利用我们得到的贝塞尔曲线的导数公式 $y_x|_{x=x_1}=\dfrac{y_2-y_1}{x_2-x_1}, y_x|_{x=x_4}=\dfrac{y_4-y_3}{x_4-x_3}$，可以通过控制点与样条点坐标，推出所需的钳制导数值。

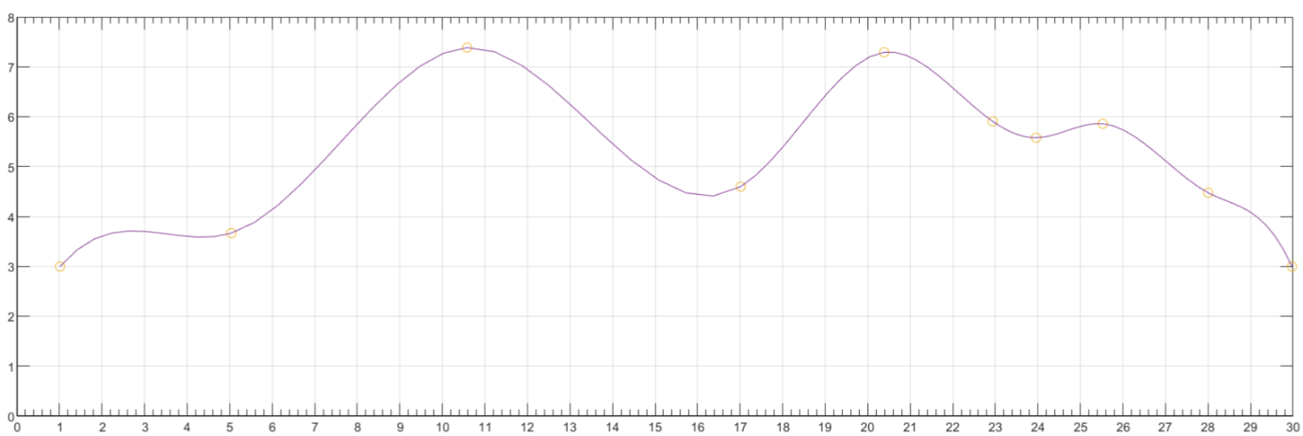
则在原贝塞尔曲线点击画图的程序中，添加获取样条点与导数值等信息，便于画出对应三次样条插值曲线 。

```
[A,B,v]=bezierdraw;%鼠标点击画出大致轮廓, 并获取相关点坐标以及导数值数据
v=[v(1),v(length(v))];%获取曲线首末点导数值
splineplot(A',B',v,10);%画出相应三次样条插值

function [X,Y,k]=bezierdraw%通过鼠标点击画出相关贝塞尔曲线并获取所点坐标与样条点处导数值
%在 MATLAB 的图形窗口中点击得到第一个点
%然后再点击三次得到两个控制点和另外一个样条点
```

```
%然后继续以每次三次点击得到三个点，给曲线增加更多段
%按下回车终止程序
plot([0 30],[0,0],'k',[0 0],[0 8],'k');hold on
axis on;%设置坐标轴
set(gca,'fontsize',8,'xminortick','on');
set(gca,'XTick',0:1:30);
grid on;%设置坐标轴网格
t=0:.02:1;
[x,y]=ginput(1);%进行一次鼠标点击
X(1)=x;
Y(1)=y;%保存第一次首端点
k=[];
while(0 == 0)
  [xnew,ynew] = ginput(3);  %另外三次点击
  if length(xnew) < 3
    break                  %按下回车键，终止
  end
  x=[x;xnew];y=[y;ynew];%画出样条点和控制点 pts
  k1=(y(2)-y(1))./(x(2)-x(1));
  k2=((y(4)-y(3))./(x(4)-x(3)));
  k=[k;k1;k2];%保存每次曲线首末端点导数值
  plot([x(1) x(2)],[y(1) y(2)],'r:',x(2),y(2),'rs');
  plot([x(3) x(4)],[y(3) y(4)],'r:',x(3),y(3),'rs');
  plot(x(1),y(1),'bo',x(4),y(4),'bo');
  bx=3*(x(2)-x(1)); by=3*(y(2)-y(1)); %样条方程
  cx=3*(x(3)-x(2))-bx;cy=3*(y(3)-y(2))-by;
  dx=x(4)-x(1)-bx-cx;dy=y(4)-y(1)-by-cy;
  xp=x(1)+t.*(bx+t.*(cx+t*dx));    %霍纳方法
  yp=y(1)+t.*(by+t.*(cy+t*dy));
  plot(xp,yp,'b')                %画出样条曲线
  x=x(4);y=y(4); %将上段最后一点转化为第一点，继续
  X=[X;x];Y=[Y;y];%保存每次曲线末端点
end
hold off
end
function coeff=splinecoeff(x,y,v)
n=length(x);
v1=v(1);
vn=v(2);
A=zeros(n,n);%矩阵 A 是 n*n
r=zeros(n,1);
for i=1:n-1%定义 deltas
    dx(i)=x(i+1)-x(i);dy(i)=y(i+1)-y(i);
end
for i=2:n-1%加载 A 矩阵
    A(i,i-1:i+1)=[dx(i-1),2*(dx(i-1)+dx(i)),dx(i)];
    r(i)=3*(dy(i)/dx(i)-dy(i-1)/dx(i-1));%右侧端点
end
A(1,1:2)=[2*dx(1) dx(1)];r(1)=3*(dy(1)/dx(1)-v1);%钳制
A(n,n-1:n)=[dx(n-1),2*dx(n-1)];r(n)=3*(vn-dy(n-1)/dx(n-1));
coeff=zeros(n,3);
coeff(:,2)=A\r;%求解系数 c
for i=1:n-1%求解 b 和 d
    coeff(i,3)=(coeff(i+1,2)-coeff(i,2))/(3*dx(i));
    coeff(i,1)=dy(i)/dx(i)-dx(i)*(2*coeff(i,2)+coeff(i+1,2))/3;
end
coeff=coeff(1:n-1,1:3);
```

```
end

function [x1,y1]=splineplot(x,y,v,k)
n=length(x);
coeff=splinecoeff(x,y,v);
x1=[];y1=[];
for i=1:n-1
    xs=linspace(x(i),x(i+1),k+1);
    dx=xs-x(i);
    ys=coeff(i,3)*dx;%使用嵌套乘法
    ys=(ys+coeff(i,2)).*dx;
    ys=(ys+coeff(i,1)).*dx+y(i);
    x1=[x1;xs(1:k)'];y1=[y1;ys(1:k)'];
end
x1=[x1;x(end)];y1=[y1;y(end)];
plot([0 30],[0,0],'k',[0 0],[0 8],'k');hold on
axis on;%设置坐标轴
set(gca,'fontsize',8,'xminortick','on');
set(gca,'XTick',0:1:30);
grid on;%设置坐标轴网格
plot(x,y,'o',x1,y1)
end
```



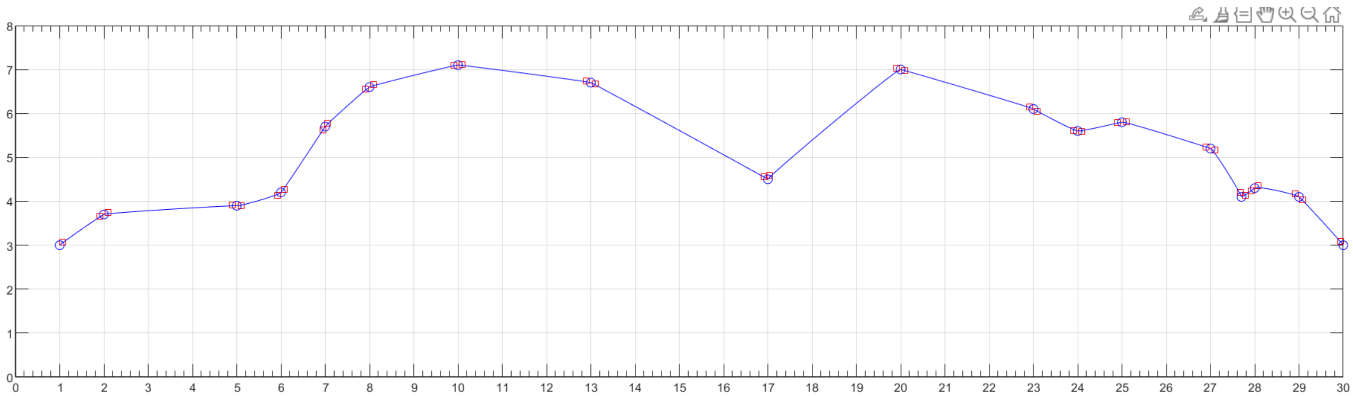


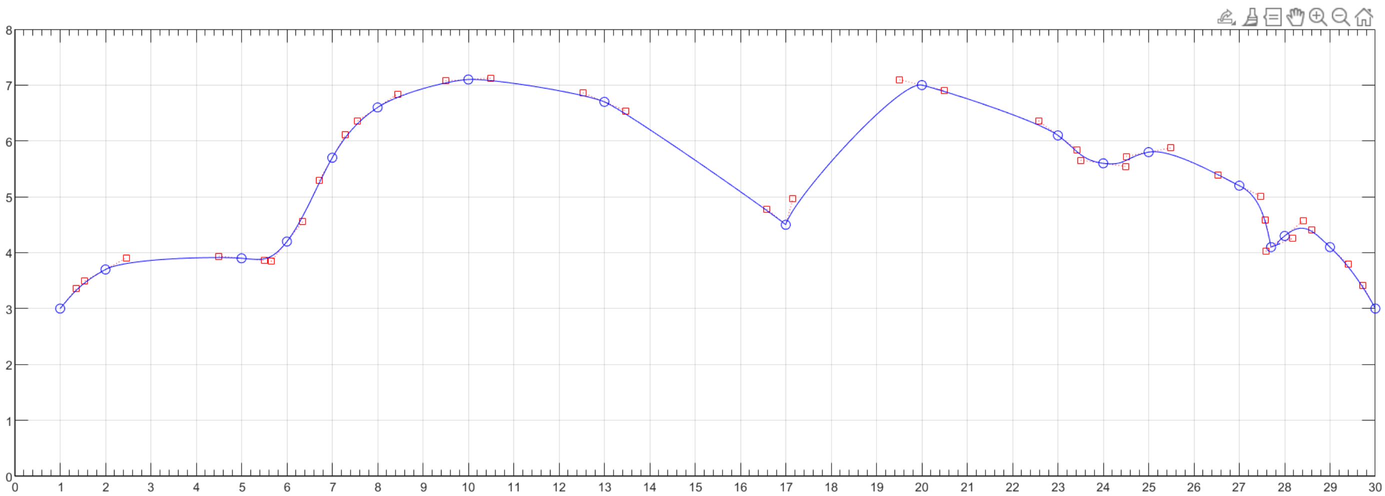
五、结果分析

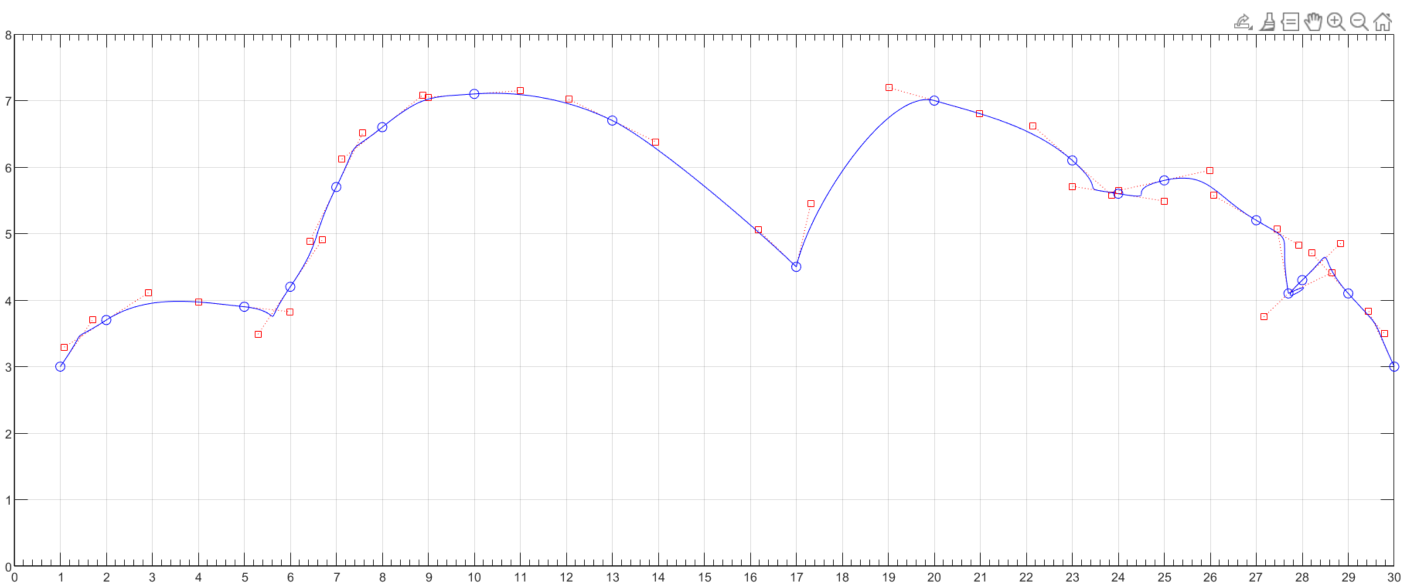
三次样条插值由于稳定性较高，拟合曲线较光滑，但是缺少更细致的细节。

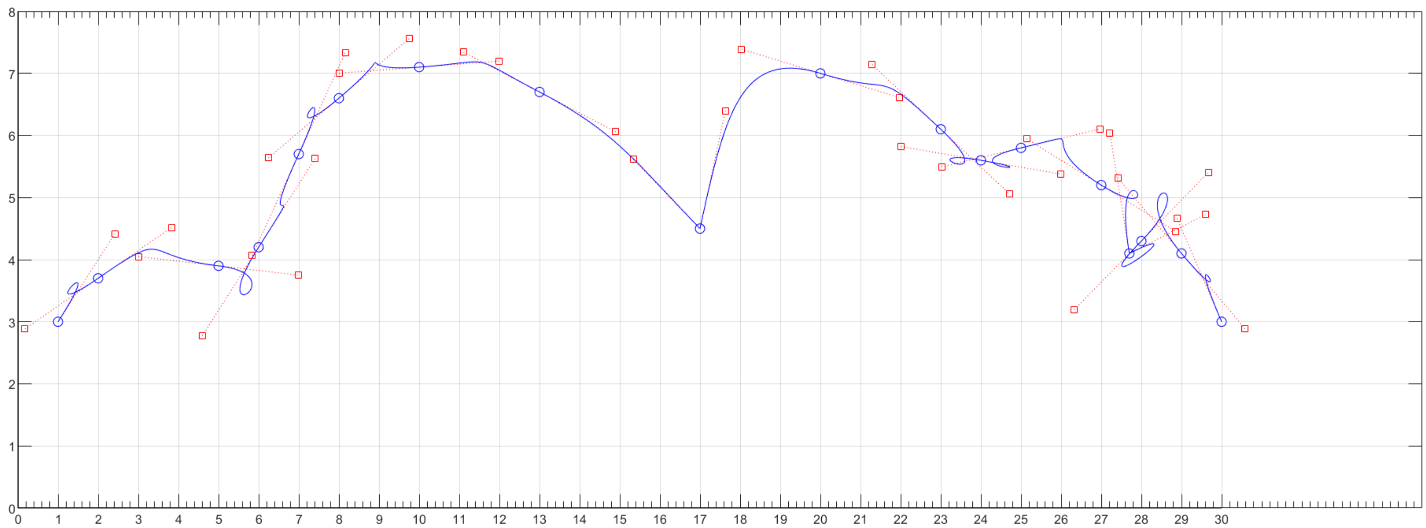
贝塞尔曲线对图像细节把握更精准，但控制点的不一带来的稳定性波动较大。在点击画图过程中，控制点的选取远近不同，造成的差异较大。而在第二题控制导数值画贝塞尔曲线时，控制点距样条点的远近差异带来的影响也较大。

例如上图控制 r 为 0.6，而当我进行调整时，r=0.1,0.5,1,2 的不同图像如下。









但笔者在试参数的时候发现，若 r 取端点值坐标差的一半，$r=\frac{x_4-x_1}{2}$，效果较好