浙江大学实验报告

专业：混合班
姓名：徐圣泽
学号：3190102721
日期：2021.4.4

课程名称：科学计算　　指导老师：赖俊　　实验名称：样条函数实验

第三次作业

问题

（1）编写不同三次样条函数

根据以下条件编写五种不同的三次样条函数$I_h(x)$，函数的输入为插值点坐标和函数值，函数的输出为$I_h(x)$的系数。

1.自然三次样条：边界条件为$S_1''(x_0)=0, S_n''(x_n)=0$。

2.曲率调整三次样条：边界条件为$S_1''(x_0)=a=y_0'', S_n''(x_n)=b=y_n''$。

3.钳制三次样条：边界条件为$S_1'(x_0)=a=y_1', S_n(x_n)=b=y_n'$。

4.抛物线端点的三次样条：边界条件为$S_1''(x_0)=S_1''(x_1), S_n''(x_{n-1})=S_n''(x_n)$。

5.非扭结三次样条：边界条件为$S_1'''(x_1)=S_1'''(x_2), S_{n-1}'''(x_{n-1})=S_n'''(x_{n-1})$。

（2）绘制误差图及求误差值

利用编写程序测试函数$y(x)=\frac{1}{1+25x^2}$在区间$[0,1]$上的不同三次样条插值函数，其中插值点分为两种：①$x_i=\frac{i}{n}$，②$x_i=(\frac{i}{n})^2$，其中$i=0,1,\ldots,n$，$n=20$。画出$y(x)-I_h(x)$的误差（取100等距点），并给出$x=0.03$和$x=0.97$的误差值。

（3）结论与探讨

根据题设条件和编写程序得到的结果，探讨哪种边界条件比较合理。

数学原理

设在区间$[a,b]$上取$n+1$个节点，给定节点上的函数值$f(x_i)=y_i$，现在构造三次样条插值函数$s(x)$满足下列条件：

(1)$s(x_i)=y_i$.
(2)在每个小区间$[x_i,x_{i+1}]$上是一个不高于三次的多项式.
(3)$s(x)\in C_{[a,b]}^2$.

假设在区间$[a,b]$上三次样条插值函数$s(x)$存在，并用m_i来表示$s(x)$在点x_i处的微商值，由于曲线通过点(x_i,y_i)，并且在每一个小区间$[x_i,x_{i+1}]$上满足条件：

$$
\begin{gathered}
s(x_i)=y_i, s(x_{i+1})=y_{i+1} \\
s'(x_i)=m_i, s'(x_{i+1})=m_{i+1}
\end{gathered}
$$

故根据$Hermite$插值公式写出小区间$[x_i,x_i+1]$上的三次样条插值函数$s(x)$的计算公式：

$$
\begin{aligned}
s(x)=&(1+2\frac{x-x_i}{x_{i+1}-x_i})(\frac{x-x_{i+1}}{x_i-x_{i+1}})^2y_i+(1+2\frac{x-x_{i+1}}{x_i-x_{i+1}})(\frac{x-x_i}{x_{i+1}-x_i})^2y_{i+1} \\
&+(x-x_i)(\frac{x-x_{i+1}}{x_i-x_{i+1}})^2m_i+(x-x_{i+1})(\frac{x-x_i}{x_{i+1}-x_i})^2m_{i+1}
\end{aligned}
$$

需要设法求出节点x_i处的微商值m_i，利用函数$s(x)$在节点x_i上二阶微商连续的性质，对x求微商并令$h_i=x_{i+1}-x_i$，不难得到：

$$
\begin{gathered}
s''(x)=(\frac{6}{h_i^2}-\frac{12}{h_i^3}(x_{i+1}-x))y_i+(\frac{6}{h_i^2}-\frac{12}{h_i^3}(x-x_i))y_{i+1}+ \\
(\frac{2}{h_i}-\frac{6}{h_i^2}(x_{i+1}-x))m_i-(\frac{2}{h_i}-\frac{6}{h_i^2}(x-x_i))m_{i+1}
\end{gathered}
$$

因此可以得到区间$[x_i,x_i+1]$上点x_i的右微商和左微商：

$$
\begin{gathered}
s''(x_i^+)=-\frac{6}{h_i^2}y_i+\frac{6}{h_i^2}y_{i+1}-\frac{4}{h_i}m_i-\frac{2}{h_i}m_{i+1} \\
s''(x_i^-)=\frac{6}{h_{i-1}^2}y_{i-1}-\frac{6}{hi-1^2}y_i+\frac{2}{h_{i-1}}m_{i-1}+\frac{4}{h_{i-1}}m_i
\end{gathered}
$$

利用左微商等于右微商，可整理得到：

$$
\begin{cases}
\alpha_i=\dfrac{h_{i-1}}{h_{i-1}+h_i} \\
\beta_i=3(\dfrac{1-\alpha_i}{h_{i-1}}(y_i-y_{i-1})+\dfrac{\alpha_i}{h_i}(y_{i+1}-y_i))
\end{cases}
$$

同时得到方程组：

$$(1-\alpha_i)m_{i-1}+2m_i+\alpha_i m_{i+1}=\beta_i$$

这是关于$n+1$个未知量的$n-1$个线性方程组，因此为了得到特定的一个解，还需要边界条件。下面每个小题都提供了一种边界条件的情形，补充了两个附加条件。

注：在下面的解题过程中，代码中出现的下标可能和本部分数学原理中出现的下标不同。

代码及运行结果

（1）自然三次样条

（i）函数部分

我们将函数分为几个模块编写，便于在每个问题求解时只需对函数进行微调程序即可正常运行，接下来以第一小题为例进行说明。

首先函数名的定义，这是为了更方面区分各个小题之间调用的不同函数：

```
function M = sort1spline(x,y,s0,sn)%x,y为变量序列参数，s0,sn为边界参数
```

定义变量：

```
n=length(x)-1;
M=zeros(1,n+1);m=zeros(n+1,1);
h=zeros(1,n);
a=zeros(1,n); b=zeros(1,n);
A=zeros(n+1,n+1);B=zeros(n+1,1);
```

根据公式对各元素进行计算：

```
for i=1:n
    h(i)=x(i+1)-x(i);
    if i==1
        a(i)=0;
        b(i)=0;
    else
        a(i)=1.0*h(i-1)/(h(i-1)+h(i));
        b(i)=3*1.0*((1-a(i))/h(i-1)*(y(i)-y(i-1))+a(i)*1.0/h(i)*(y(i+1)-y(i)));
    end
end
```

对系数矩阵和各列向量的元素进行定义：

```
A(1,1)=2;A(1,2)=1;B(1,1)=3/h(1)*(y(2)-y(1))-h(1)/2*s0;%方程组中第一个式子
for i=2:n%方程组中第2~n个式子
    A(i,i-1)=1-a(i);A(i,i)=2;A(i,i+1)=a(i);
    B(i,1)=b(i);
end
A(n+1,n)=1;A(n+1,n+1)=2;B(n+1,1)=3/h(n)*(y(n+1)-y(n))+h(n)/2*sn;%方程组中最后一个式
子
```

最后得到目的列向量，并转置得到函数值：

```
m=A\B;M=m';
```

注：将上述部分顺次拼接即得到本题求解所需函数完整代码。

（ii）绘图及求误差部分

首先定义本题求解中需要用到的变量，其中有部分为传入函数的参数：

```
x1=0:0.05:1;
y1=1*1.0./(1+25*x1.^2);
x2=x1.^2;
y2=1*1.0./(1+25*x2.^2);
x=0:0.01:1;
y=1*1.0./(1+25*x.^2);
n1=length(x1);
N=length(x);
caculatedvalue1=zeros(1,n1);
caculatedvalue2=zeros(1,n1);
```

本题中所需用到的边界值参数：

```
s0=0;sn=0;%(下面一小题只需将此部分改动一下即可：s0=-50;sn=925/4394;)
```

根据题目要求，绘制两条误差曲线：

```
M1=sort1spline(x1,y1,s0,sn);
for i=1:N
    for j=1:n1-1
        if (x(i)>=x1(j)) && (x(i)<=x1(j+1))
            caculatedvalue1(i)=(1+2*1.0*(x(i)-x1(j))/(x1(j+1)-x1(j)))*(x(i)-
x1(j+1))*(x(i)-x1(j+1))/((x1(j)-x1(j+1))^2)*y1(j)+(1+2*1.0*(x(i)-
x1(j+1))/(x1(j)-x1(j+1)))*(x(i)-x1(j))*(x(i)-x1(j))/((x1(j)-x1(j+1))^2)*y1(j+1)+
(x(i)-x1(j))*1.0*(x(i)-x1(j+1))*(x(i)-x1(j+1))/((x1(j)-x1(j+1))^2)*M1(j)+(x(i)-
x1(j+1))*1.0*(x(i)-x1(j))*(x(i)-x1(j))/((x1(j)-x1(j+1))^2)*M1(j+1);
        end
    end
end
error1=y-caculatedvalue1;
subplot(2,1,1);
plot(x,error1,'r');

M2=sort1spline(x2,y2,s0,sn);
for i=1:N
    for j=1:n1-1
        if (x(i)>=x2(j)) && (x(i)<=x2(j+1))
            caculatedvalue2(i)=(1+2*1.0*(x(i)-x2(j))/(x2(j+1)-x2(j)))*(x(i)-
x2(j+1))*(x(i)-x2(j+1))/((x2(j)-x2(j+1))^2)*y2(j)+(1+2*1.0*(x(i)-
x2(j+1))/(x2(j)-x2(j+1)))*(x(i)-x2(j))*(x(i)-x2(j))/((x2(j)-x2(j+1))^2)*y2(j+1)+
(x(i)-x2(j))*1.0*(x(i)-x2(j+1))*(x(i)-x2(j+1))/((x2(j)-x2(j+1))^2)*M2(j)+(x(i)-
x2(j+1))*1.0*(x(i)-x2(j))*(x(i)-x2(j))/((x2(j)-x2(j+1))^2)*M2(j+1);
        end
    end
end
error2=y-caculatedvalue2;
subplot(2,1,2);
plot(x,error2,'r');
```

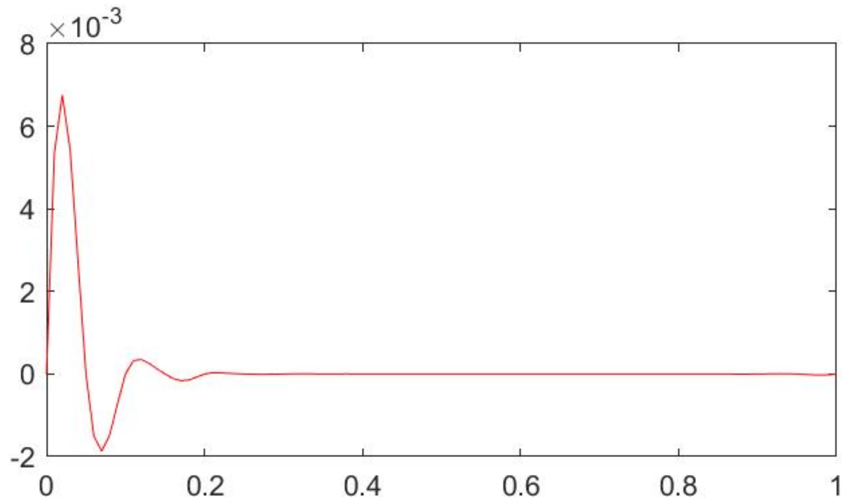
根据题目要求，求出$x=0.03$和$x=0.97$处的误差值：

```
disp(error1(4));
disp(error1(98));
disp(error2(4));
disp(error2(98));
```

注：将上述部分顺次拼接即得到本题所需绘图及误差求解完整代码。

(iii）运行结果

第一种插值点得到的曲线图：

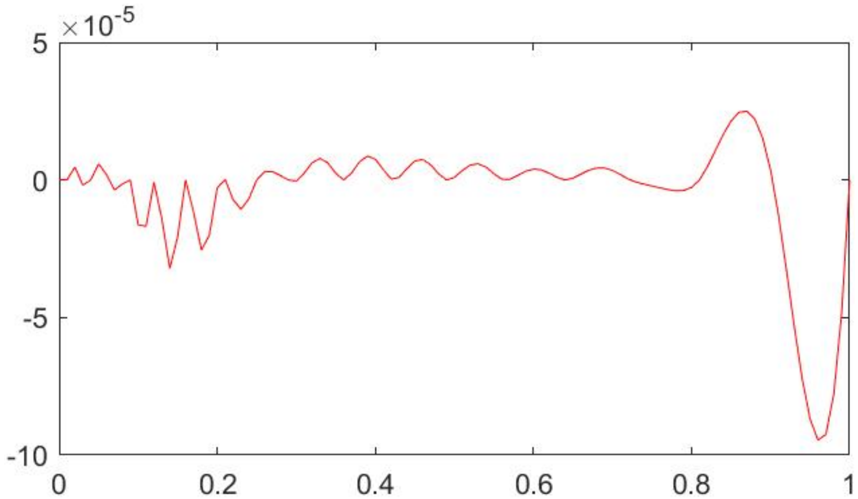


固定点处的误差值：

$x=0.03$处：0.0054

$x=0.97$处：-2.0134×10^{-5}

第二种插值点得到的曲线图：



固定点处的误差值：

$x=0.03$处：-1.9130×10^{-6}

$x=0.97$处：-9.2497×10^{-5}

（2）曲率调整三次样条

（i）函数部分

本部分函数与第一小题中相同。

（ii）绘图及求误差部分

这种情况和第（1）种情况的原理是相同的，仅仅改变了传入函数的参数值，在上述代码的解释中也已经提及。

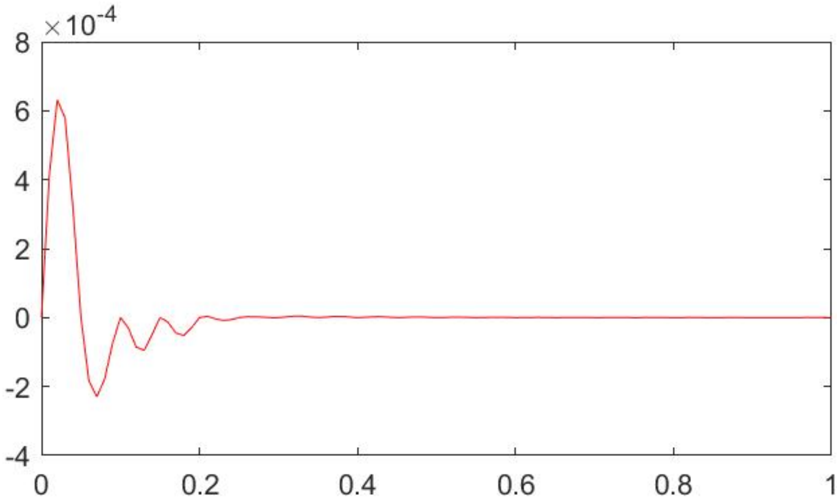
只需将 `s0=0;sn=0;` 此部分的参数值改为题干值即可。

根据题干要求，求得$y''(x) = \frac{-50(1+25x^2)+5000x}{(1+25x^2)^3}$，代入$x = 0$和$x = 1$，解得所需参数值分别为$s_0 = -50$和$s_n = \frac{925}{4394}$。

代入即可得本题解。

（iii）运行结果

第一种插值点得到的曲线图：

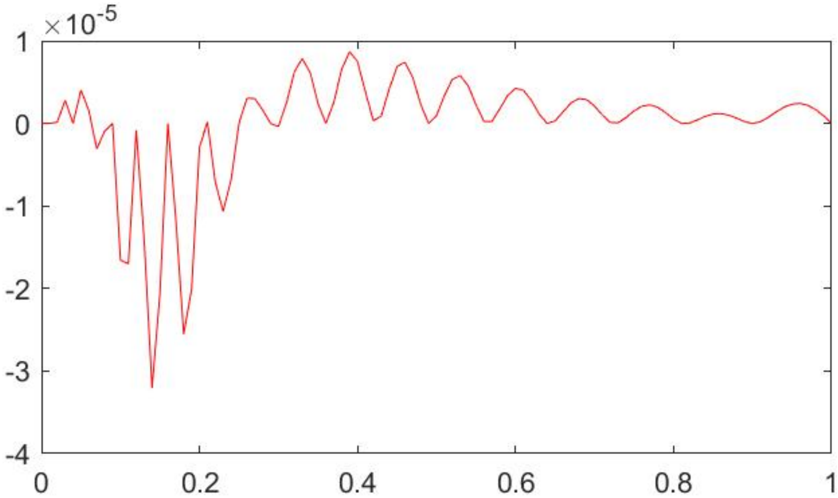


固定点处的误差值：

$x = 0.03$处：5.7810×10^{-4}

$x = 0.97$处：1.3275×10^{-7}

第二种插值点得到的曲线图：



固定点处的误差值：

$x=0.03$处：2.7970×10^{-6}

$x=0.97$处：2.1875×10^{-6}

（3）钳制三次样条

（i）函数部分

因为在上面我们已经对函数进行了模块化的编写，因此只需要改变其中一些部分便可继续调用。

首先定义新的函数名：

```
function M = sort2spline(x,y,s0,sn)
```

中间部分几乎一样，不在此赘述，主要是系数矩阵和各列向量的元素的定义改动较大：

```
A(1,1)=1;B(1,1)=s0;%方程组中第一个式子
for i=2:n%方程组中第2~n个式子
    A(i,i-1)=1-a(i);A(i,i)=2;A(i,i+1)=a(i);
    B(i,1)=b(i);
end
A(n+1,n+1)=1;B(n+1,1)=sn;%方程组中最后一个式子
```

在接下来的几个小题中，也主要都是通过改动此部分得到新的函数。

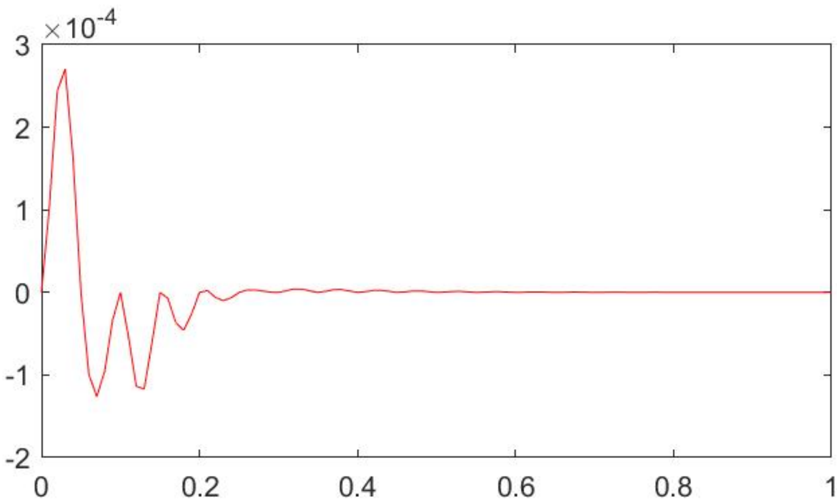
（ii）绘图及求误差部分

此部分并未进行较大改动，只需根据题干条件改动参数值，同时调用新的`sort2spline`函数。

由$y'(x)=-\frac{50x}{(1+25x^2)^2}$，得到$s_0=0$和$s_n=-25/338$。

（iii）运行结果

第一种插值点得到的曲线图：

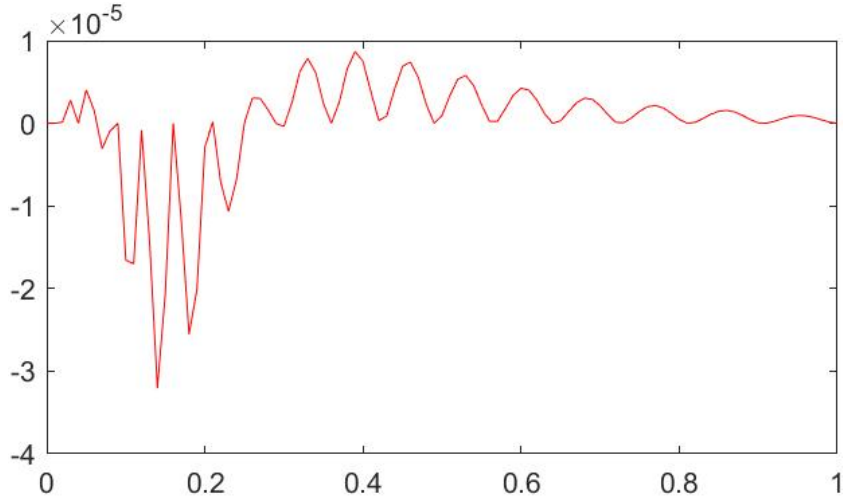


固定点处的误差值：

$x=0.03$处：2.7069×10^{-4}

$x=0.97$处：5.4926×10^{-8}

第二种插值点得到的曲线图：



固定点处的误差值：

$x=0.03$处：2.7965×10^{-6}

$x=0.97$处：7.0694×10^{-7}

（4）抛物线端点的三次样条

（i）函数部分

定义新的函数，此时只需传入两个参数。

```
function M = sort3spline(x,y)
```

改动系数矩阵和列向量元素定义部分：

```
A(1,1)=1;A(1,2)=1;B(1,1)=2/h(1)*(y(2)-y(1));%方程组中第一个式子
for i=2:n%方程组中第2~n个式子
    A(i,i-1)=1-a(i);A(i,i)=2;A(i,i+1)=a(i);
    B(i,1)=b(i);
end
A(n+1,n)=1;A(n+1,n+1)=1;B(n+1,1)=2/h(n)*(y(n+1)-y(n));%方程组中最后一个式子
```

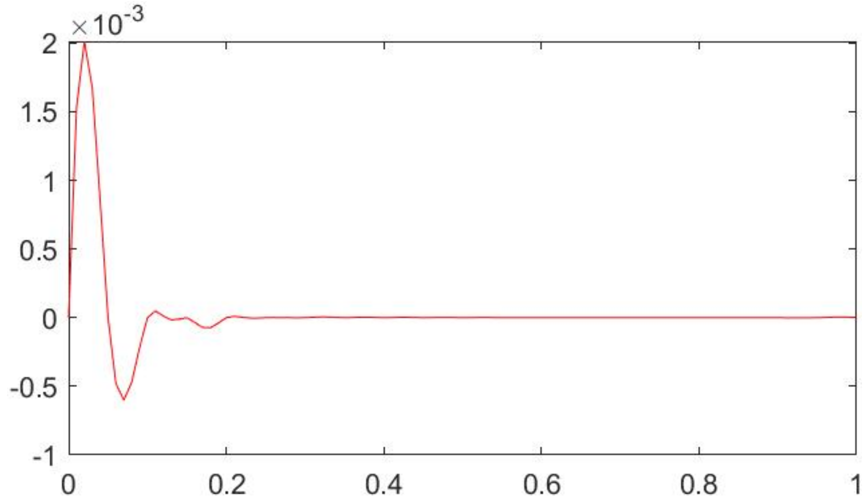
其余部分均保持不变。

（ii）绘图及求误差部分

将调用部分改为调用新定义的函数即可。

（iii）运行结果

第一种插值点得到的曲线图：

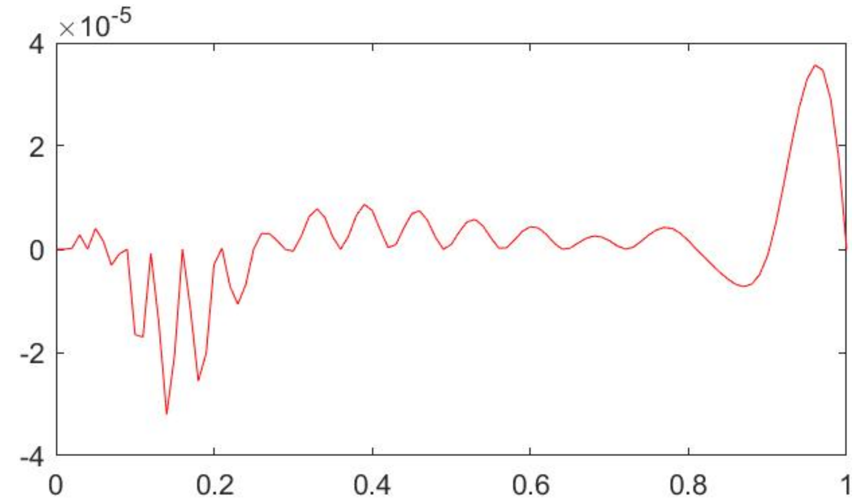


固定点处的误差值：

$x = 0.03$处：0.0017

$x = 0.97$处：3.4402×10^{-6}

第二种插值点得到的曲线图：



固定点处的误差值：

$x = 0.03$处：2.7960×10^{-6}

$x = 0.97$处：3.4689×10^{-5}

（5）非扭结三次样条

（i）函数部分

定义新的函数，依然只需要两个参数：

```
function M = notaknotspline(x,y)
```

改动系数矩阵和列向量元素定义部分：

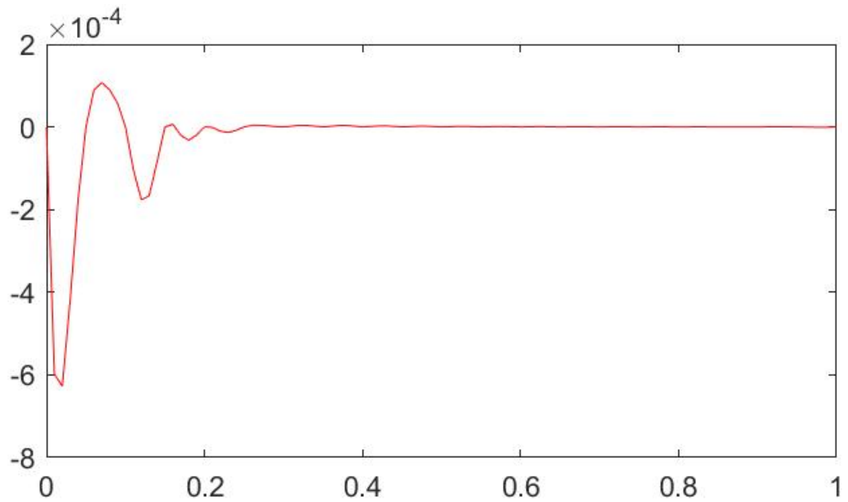
```
A(1,1)=1/(h(1)^2);A(1,2)=1/(h(1)^2)-1/(h(2)^2);A(1,3)=-1/(h(2)^2);
B(1,1)=-2/(h(1)^3)*y(1)+(2/(h(1)^3)+2/(h(2)^3))*y(2)-2/(h(2)^3)*y(3);%方程组中第一个式子
for i=2:n%方程组中第2~n个式子
    A(i,i-1)=1-a(i);A(i,i)=2;A(i,i+1)=a(i);
    B(i,1)=b(i);
end
A(n+1,n-1)=1/(h(n-1)^2);A(n+1,n)=1/(h(n-1)^2)-1/(h(n)^2);A(n+1,n+1)=-1/(h(n)^2);
B(n+1,1)=-2/(h(n-1)^3)*y(n-1)+(2/(h(n-1)^3)+2/(h(n)^3))*y(n)-2/(h(n)^3)*y(n+1);%方程组中最后一个式子
```

(ii) 绘图及求误差部分

将调用部分改为调用新定义的函数即可。

(iii) 运行结果

第一种插值点得到的曲线图：

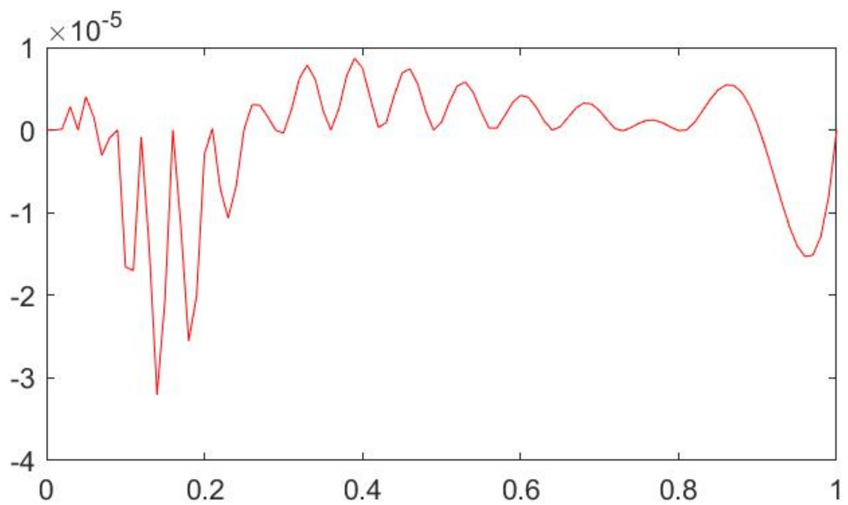


固定点处的误差值：

$x=0.03$处：-4.1996×10^{-4}

$x=0.97$处：-6.6442×10^{-7}

第二种插值点得到的曲线图：



固定点处的误差值：

$x=0.03$处：2.8121×10^{-6}

$x = 0.97$处：-1.5134×10^{-5}

思考与讨论

（1）结论

通过上面绘制的误差曲线图可以发现，两种插值点划分得到的误差结果差异还是比较明显的。一般地，根据第二种插值点划分方法得到的误差相对更小，因此第二种划分相对第一种来说更加合理准确。

根据上述的讨论我们可以看出，在本次实验中，根据非扭结三次样条函数得到的误差值相对较小。

（2）思考

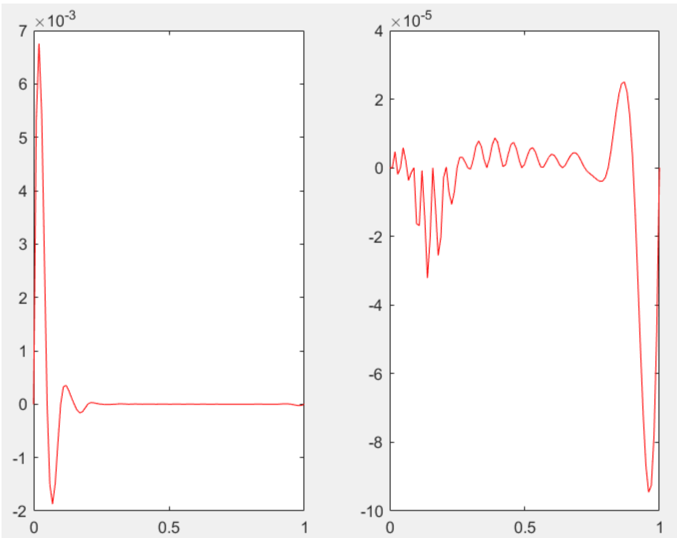
通过这次实验我们发现，这几种函数背后蕴含着不同的边界条件，但都可以求解得到最后的函数及误差值，原因是他们都能够为线性方程组提供两个方程，使得原方程组具有唯一特定解。

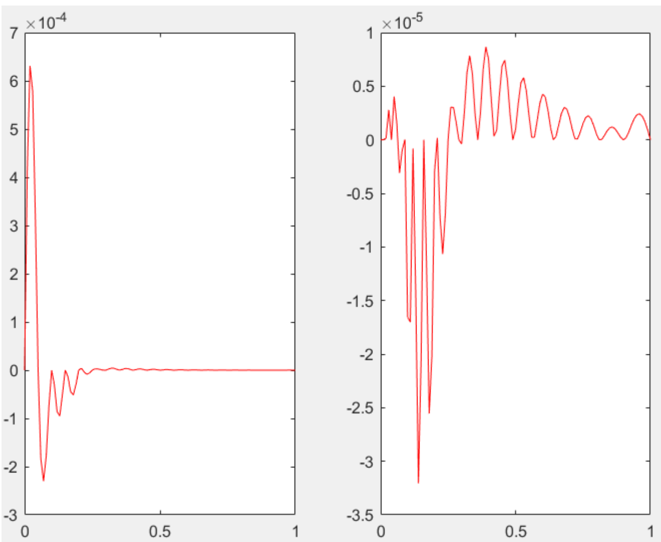
同时我们可以看到，每种边界条件的误差曲线图都各具特色，不同插值点划分也会带来不一样的结果，但是每种边界条件没有严格的优劣之分，他们都各有长短。

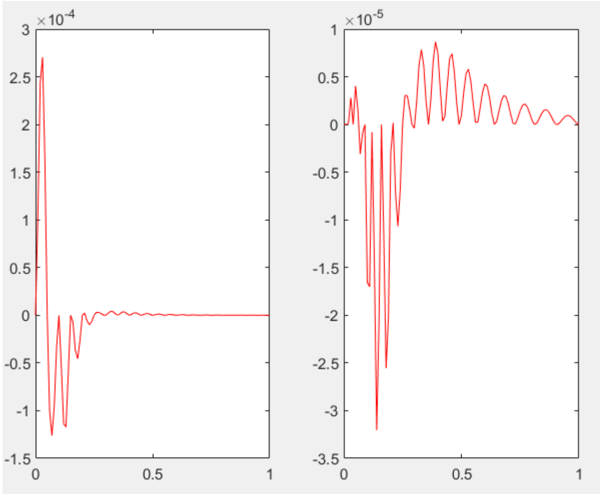
在实际解题的过程中，根据已有条件选择合适的边界条件和插值点划分方法，才能够合理地使误差最小，得到最合适的答案。

附录

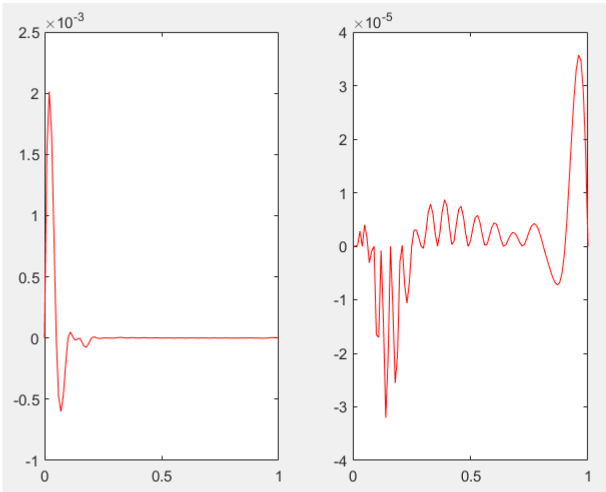
下面是五个小题两种划分方法的误差曲线的对比图，这样看更加直观明了。



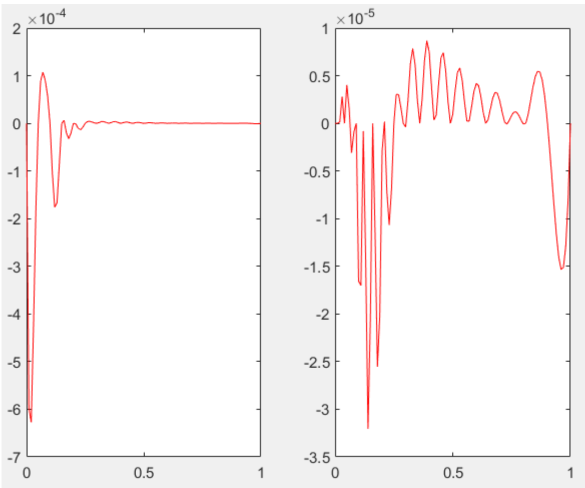




×10^-4
3
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
0
0.5
1
×10^-5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
-3
-3.5
0
0.5
1



×10^-3
2.5
2
1.5
1
0.5
0
-0.5
-1
0
0.5
1
×10^-5
4
3
2
1
0
-1
-2
-3
-4
0
0.5
1



×10^-4
2
1
0
-1
-2
-3
-4
-5
-6
-7
0
0.5
1
×10^-5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
-3
-3.5
0
0.5
1