

# 数值分析第四次上机练习实验报告

## ——函数逼近的插值方法

huangyf15

### 一、问题描述

设  $f(x) = \frac{1}{1+25x^2}$ ,  $x \in [-1,1]$ . 取  $x_j = -1 + \frac{2j}{n}$ ,  $j = 0,1,\dots,n$ . 取适当的  $n$  (比如  $n=10, 20, \dots$  等), 试求出  $n$  次 Lagrange 插值多项式  $L_n(x)$ 、分段线性插值函数  $I_1^h(x)$  和三次样条插值函数  $S_1^h(x)$  (采用自然边界条件), 画出它们的图像, 并对结果做一个比较说明.

### 二、方法描述——多项式简单插值与分段低次插值

插值方法是函数逼近的重要方法, 即构造插值函数使其在节点处与被插值函数的性质相似, 如函数值、一阶导数值相同等等. 我们期望随着节点个数的增多, 插值函数在特定范数的意义下能够趋近于被插值函数.

从插值条件上看, 除节点处函数值相等这一最基本的要求外, 还有节点导数值相等的 Hermite 条件、分段插值函数在节点处的样条条件及对应的若干边界条件, 而这就衍生出了简单插值(只要求节点处函数值相等)、Hermite 插值(提高插值逼近的密切程度)、样条插值(提高插值函数光滑性)等多种思路. 无论哪种思路, 从插值函数的构造方式上看, 大致都分为 Lagrange 与 Newton 两种类型, 前者应用了基函数的思想, 这在各种类型的插值中都有广泛的应用; 后者采用了均差的工具, 使得插值函数的表达式更为优美统一, 便于迭代推广.

为了提高插值的数值稳定性与逼近精度, 尤其是避免诸如 Runge 现象的发生, 被插值函数的光滑性、各阶导数的一致有界性起到了很大作用. 但这在实际情况中很难保证, 如函数存在阶跃点、导数不具有有一致有界性等等, 这就要求我们从改进插值方法的角度考虑. 一方面, 可以扩大用于插值的函数类, 从单一多项式插值过渡到分段低次插值、有理函数差值等等, 在保证数值稳定性的前提下提高精度; 另一方面, 使用不等距节点, 即先估计被插值函数导数值较大的区间, 然后在这些区间设置更加密集的节点以获得更好的逼近效果.

这次上机实验选取了 Lagrange 简单插值以及分段低次插值中的分段线性插值、三次样条插值三种比较有代表性的方法, 对目标函数进行插值, 并进行结果的比较.

#### 1. 多项式 Lagrange 简单插值

(1)插值条件:

a.在整体区间上为  $n$  次多项式;

b.在节点处与被插值函数的函数值相等;

(2)插值函数构造:

利用 Lagrange 基函数的方法, 即写

$$p_n(x) = \sum_{k=0}^n f(x_k)l_k(x).$$

其中,  $l_k(x)$  线性无关且满足

$$l_k(x_m) = \delta_{km}, \quad k, m = 0, 1, \dots, n.$$

这样,  $\{l_k(x)\}_{k=0}^n$  就构成了  $P_n$  的一组基函数.

我们令

$$l_k(x) = \prod_{i=0, i \neq k}^n \frac{x - x_i}{x_k - x_i}, \quad k = 0, 1, \dots, n.$$

即可满足要求.

## 2.分段线性插值

由于简单插值的收敛性定理对被插值函数的要求很高, 尤其是在对一些各阶导数不一致有界的函数进行高次多项式插值时, 常常会出现所谓的 Runge 现象, 因此人们会更多地采用分段低次插值的方法提高逼近精度, 这也往往会收获比较理想的逼近效果. 分段线性插值是分段低次插值中最简单的情况。

(1)插值条件:

a.在整体区间上连续;

b.在节点处与被插值函数的函数值相等;

c.在各子区间内为线性函数.

(2)插值函数构造:

利用 Lagrange 基函数的构造思想, 写

$$I_1^n(x) = \sum_{j=0}^n f(x_j)\phi_j(x).$$

其中,  $\phi_j(x)$  满足

$$\phi_j(x_k) = \delta_{jk}, \quad j, k = 0, 1, \dots, n.$$

我们令

$$\phi_j(x) = \begin{cases} \frac{x - x_{j-1}}{x_j - x_{j-1}}, & x_{j-1} \leq x \leq x_j (j = 0 \text{ 时略去}) \\ \frac{x - x_{j+1}}{x_j - x_{j+1}}, & x_j \leq x \leq x_{j+1} (j = n \text{ 时略去}), \quad j = 0, 1, \dots, n. \\ 0, & \text{其他情形} \end{cases}$$

即可满足要求.

### 3.三次样条插值

虽然分段线性插值、分段三次 Hermite 插值克服了高次多项式的弱点,但是光滑性较差.在实际问题中我们往往要求函数具有二阶光滑性,而对于三次样条插值,我们得到的是一个样条曲线,它是由分段三次曲线拼接而成,在节点处的二阶导数连续,从而满足了光滑性的要求.

(1)插值条件:

- a.在整体区间上二阶连续可导;
- b.在节点处与被插值函数的函数值相等;
- c.在各子区间内为三次多项式;
- d.为保证定解而补充的特定边界条件.

(2)插值函数构造:

这里采用转角方程的思路.利用化归的思想,我们先假设样条函数  $S'(x) = m_j (j = 0, \dots, n)$ ,从而构成了分段 Hermite 插值条件,便可顺利地写出插值函数的表达形式;接下来问题转变成了如何求解  $\{m_j\}_{j=0}^n$  的值,这利用二阶导数连续条件以及边界条件即可求出,这里我们采用的是自然边界条件,即第 II 类边界条件的特殊情况

$$S'(x_0) = S'(x_n) = 0. \quad (1)$$

a.写分段三次 Hermite 插值函数表达式

$$S(x) = \sum_{j=0}^n [f_j \alpha_j(x) + m_j \beta_j(x)].$$

其中,

$$\alpha_j(x) = \begin{cases} (1 + 2 \frac{x - x_j}{x_{j-1} - x_j}) (\frac{x - x_{j-1}}{x_j - x_{j-1}})^2, & x_{j-1} \leq x \leq x_j (j = 0 \text{ 时略去}) \\ (1 + 2 \frac{x - x_j}{x_{j+1} - x_j}) (\frac{x - x_{j+1}}{x_j - x_{j+1}})^2, & x_j \leq x \leq x_{j+1} (j = n \text{ 时略去}), \quad j = 0, 1, \dots, n. \\ 0, & \text{其他情形} \end{cases}$$

$$\beta_j(x) = \begin{cases} (x-x_j)\left(\frac{x-x_{j-1}}{x_j-x_{j-1}}\right)^2, & x_{j-1} \leq x \leq x_j (j=0 \text{ 时略去}) \\ (x-x_j)\left(\frac{x-x_{j+1}}{x_j-x_{j+1}}\right)^2, & x_j \leq x \leq x_{j+1} (j=n \text{ 时略去}), \quad j=0,1,\dots,n. \\ 0, & \text{其他情形} \end{cases}$$

特别地，在  $[x_j, x_{j+1}]$  上可写

$$S(x) = f_j \alpha_j(x) + m_j \beta_j(x) + f_{j+1} \alpha_{j+1}(x) + m_{j+1} \beta_{j+1}(x).$$

进而有

$$S''(x) = f_j \alpha_j''(x) + m_j \beta_j''(x) + f_{j+1} \alpha_{j+1}''(x) + m_{j+1} \beta_{j+1}''(x). \quad (2)$$

**b.** 求样条函数在节点处的一阶导数值  $\{m_j\}_{j=0}^n$

根据节点处二阶导数连续条件(由(2)可得)以及自然边界条件(1)可列出如下线性方程组

$$\begin{pmatrix} 2 & \mu_0 & & & \\ \lambda_1 & 2 & \mu_1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & 2 & \mu_{n-1} \\ & & & \lambda_n & 2 \end{pmatrix} \begin{pmatrix} m_0 \\ m_1 \\ \vdots \\ m_{n-1} \\ m_n \end{pmatrix} = \begin{pmatrix} g_0 \\ g_1 \\ \vdots \\ g_{n-1} \\ g_n \end{pmatrix}.$$

其中，对于  $j=1, \dots, n-1$ ，有

$$\lambda_j = \frac{h_{j+1}}{h_j + h_{j+1}}, \mu_j = 1 - \lambda_j (h_j = x_j - x_{j-1})$$

$$g_j = 3\{\lambda_j f[x_{j-1}, x_j] + \mu_j f[x_j, x_{j+1}]\}.$$

对于  $j=0, n$ ，有

$$\lambda_0 = \mu_n = 0, \lambda_n = \mu_0 = 0.$$

$$g_0 = 3f[x_0, x_1], g_n = 3f[x_{n-1}, x_n].$$

考虑到系数矩阵为严格三对角阵，因此用追赶法可以很容易地求解该线性方程组。

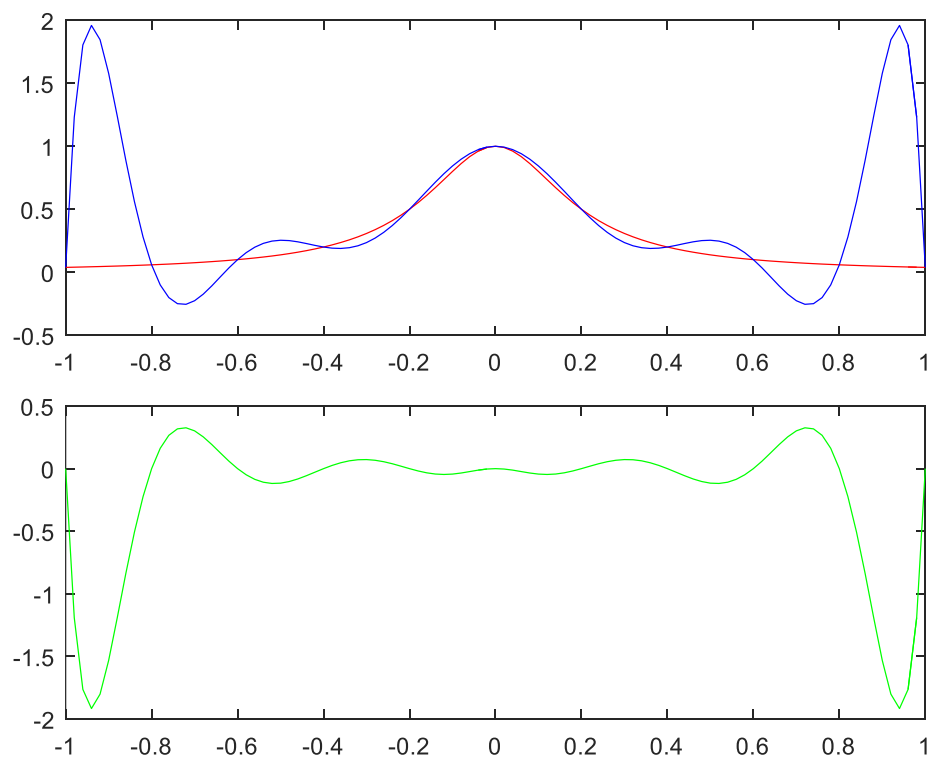
### 三、方案设计

我们通过编写 Matlab 程序来实现对于任意给定函数、给定区间、给定节点个数要求下的上述三种等距插值，并在同一张图内作出插值函数与被插值函数的图像以便比较。三个主程序文件分别为 `interp_Lagrange.m`，`interp_linear.m`，`interp_spline.m`；每个文件的主程序输入变量均为区间端点值与节点个数，被插值函数的选取通过设置子函数  $[y]=f(x)$  实现，程序最终将返回给定区间的函数图像。具体程序的实现可参见所给程序的相关注释。

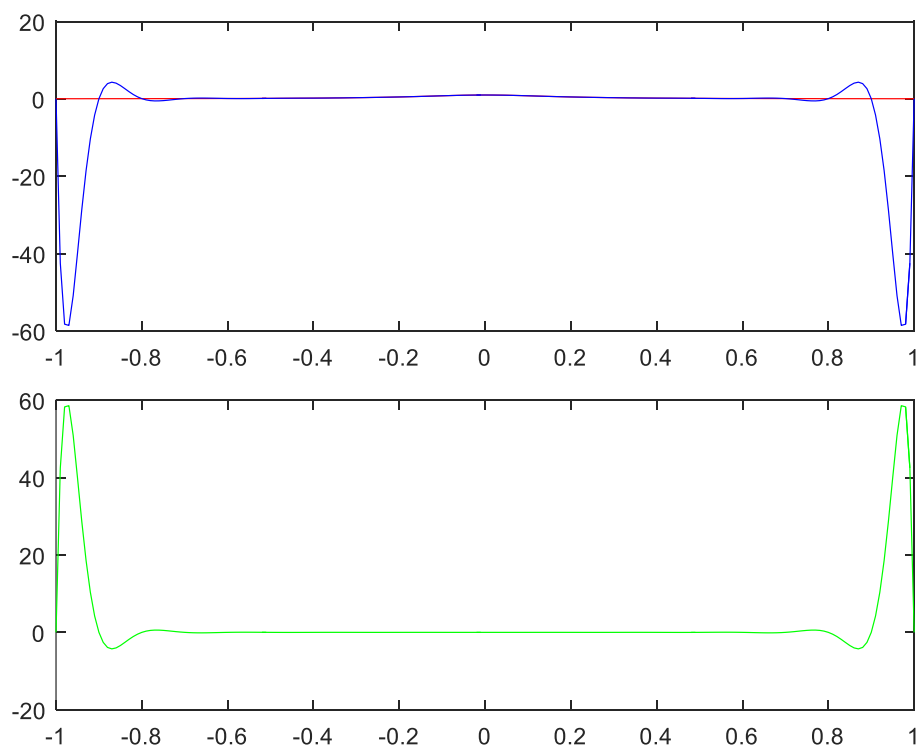
由于都采用了 Lagrange 基函数方法，因此各程序解决问题的思路类似，即先根据给定函数的解析表达式求出各节点处函数值，然后设置子程序确定各自的基函数，最后叠加上插值函数并作出函数图像。稍有不同的是，在三次样条插值程序中，还包括用于计算节点一阶导数值的子程序。

#### 四、计算结果及其分析

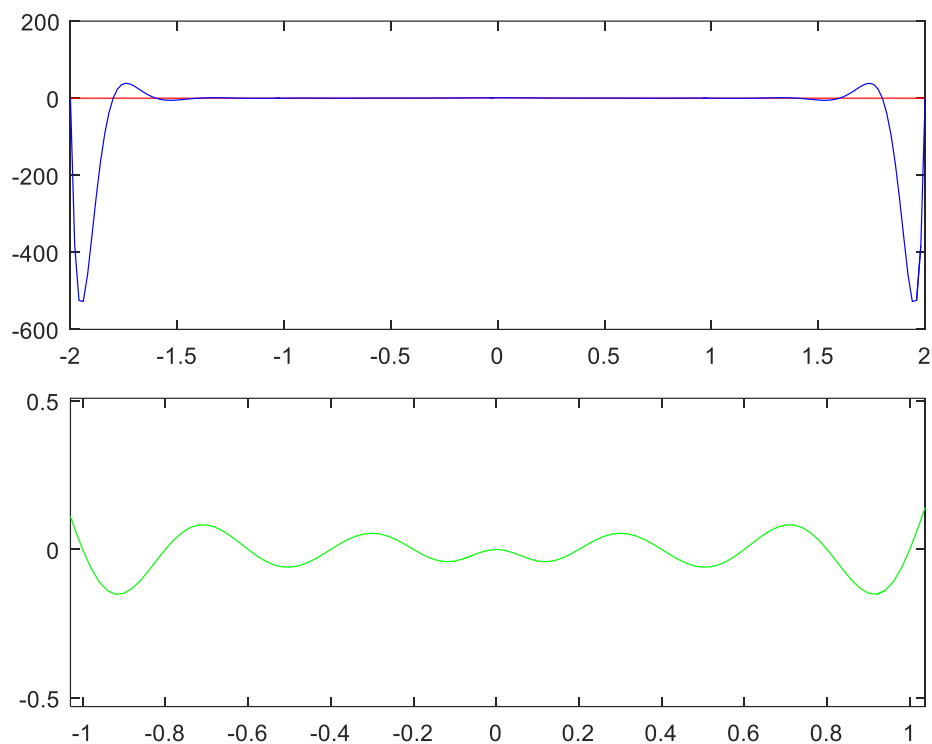
通过运行 Matlab 程序，我们得到了对应于  $f(x) = \frac{1}{1+25x^2}$ ,  $x \in [-1,1]$  的各类插值函数。为了更加直观地展现出函数逼近的效果，我们采取绘制图像的结果呈现方式(如下):



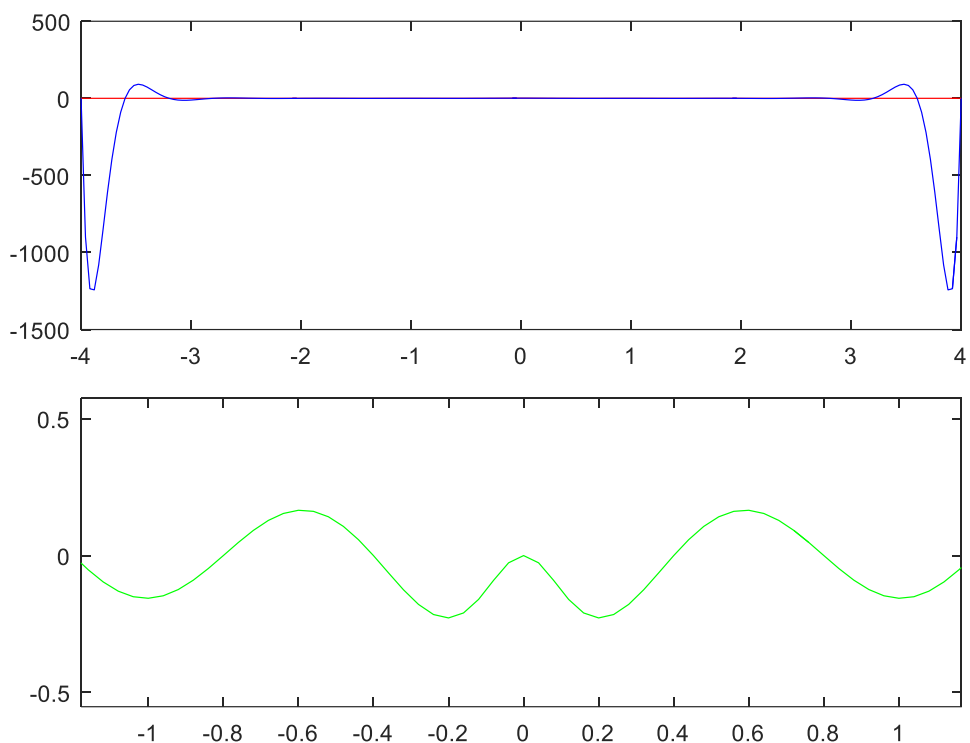
Picture1-1.  $Lagrange_{10}(x)$  on  $[-1,1]$



Picture1-2.  $Lagrange_{20}(x)$  on  $[-1,1]$



Picture1-3.  $Lagrange_{20}(x)$  on  $[-2, 2]$



Picture1-4.  $Lagrange_{40}(x)$  on  $[-4, 4]$

- 注：1. 上述图像中，红线代表被插值函数，蓝线代表插值函数，绿线代表逐点误差(下同)；  
 2. 为了行文的连贯性，已将图 2、3 系列作为附图置于文末。

从上述一系列图像中，我们可以作出如下分析：

(1)Lagrange 插值(图 1 系列)：可以看到，区间端点附近的 Runge 现象是非常明显的，而且随着节点个数的增加，误差急剧增大。注意到图 1-2 的区间中段的误差有可能比较小，这样我们期望通过增加区间长度的方式提高区间中点附近的逼近精度，然而经过试验(图 1-3,图 1-4)发现即使做了这样的迂回，精度也未有提升，同时还牺牲了计算量。综上，Lagrange 插值的逼近精度不甚理想。

(2)分段线性插值(图 2 系列)：可以看到，在同样条件下(同样的被插值函数、同样的插值区间、同样的节点个数)，分段线性插值的精度相比 Lagrange 插值而言有很大提升。尽管如此，在驻点( $x = 0$ )附近的误差仍相对比较明显，这是由被插值函数在该点附近的性态以及线性函数的特点共同决定的；同时，光滑程度不高也是分段线性插值的软肋。

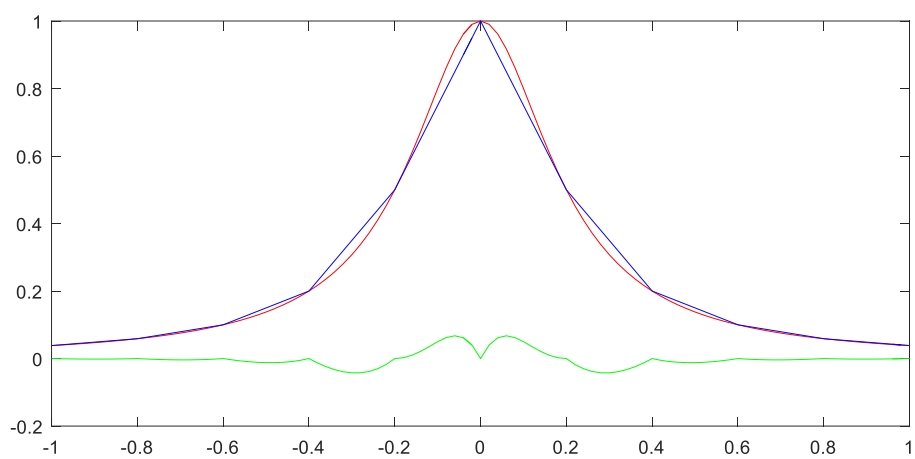
(3)三次样条插值(图 3 系列)：可以看到，在同样条件下，样条插值无论在逼近精度上还是在光滑程度上都比较理想。但也要注意，如图 3-3 所见，当区间长度对称地增大时，驻点( $x = 0$ )附近的逼近精度相对而言还是比较明显的，初步分析这是由于节点选择不当造成的。因此，即使是相对理想的样条插值，节点的选择也是非常关键的因素。

## 五、结论

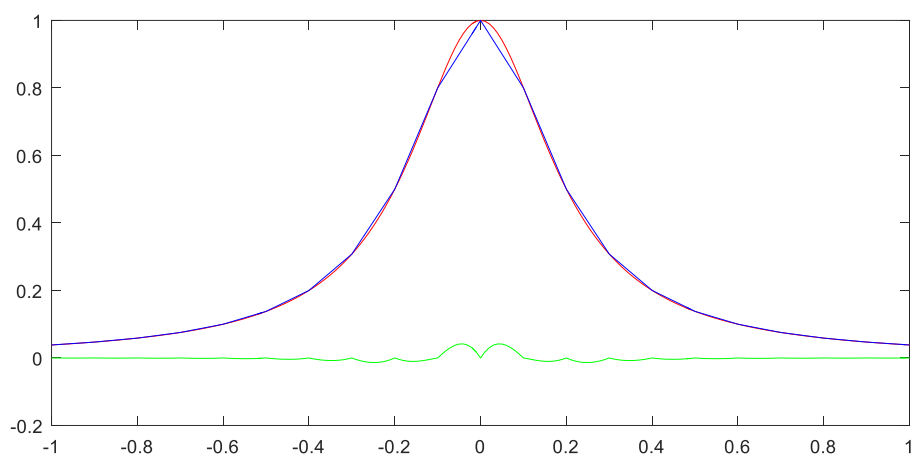
在本次上机实验中，通过比较三种典型插值法的结果，我们比较直观地认识了三者的优势与局限性，即 Lagrange 插值初步达到了函数逼近的要求，但由于 Runge 现象而造成逼近性质较坏；分段线性插值克服了逼近精度的问题，但光滑性不高；三次样条插值无论在逼近精度上，还是插值函数的光滑程度上都有较大提升。

最重要的收获是，我们发现无论在何种插值法中，插值节点的合理选取都是至关重要的。虽然等距节点最易操作，但在很多情况下，由于舍入误差影响或者被插值函数特定点的固有性质局限，一味地采用等距节点得到的逼近效果往往不尽理想。在实际情况中，应充分考虑被插值函数的性态：在驻值点附近的节点应适当密集一些；在导数值偏大的点(含第一类间断点)应借鉴 ENO 插值的思想，选取合适的模版进行插值。

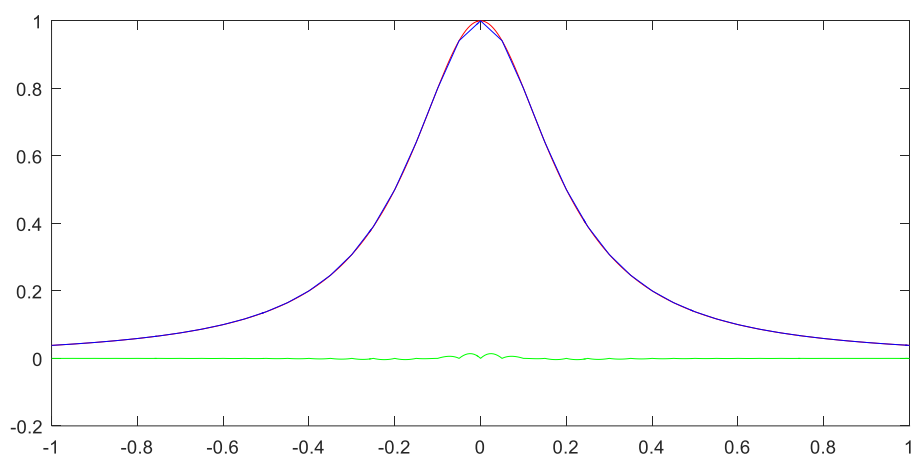
(附图)



Picture2-1.  $Linear_{10}(x)$

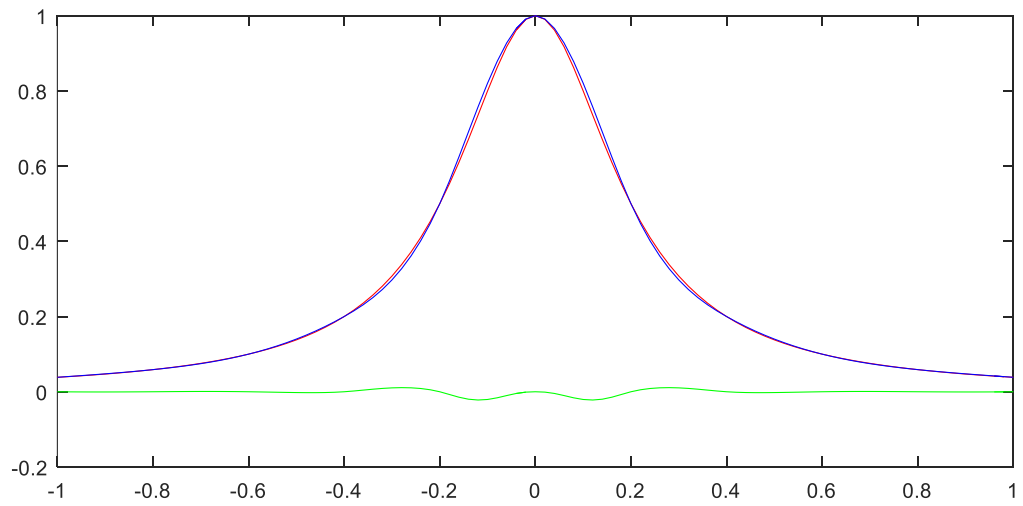


Picture2-2.  $Linear_{20}(x)$

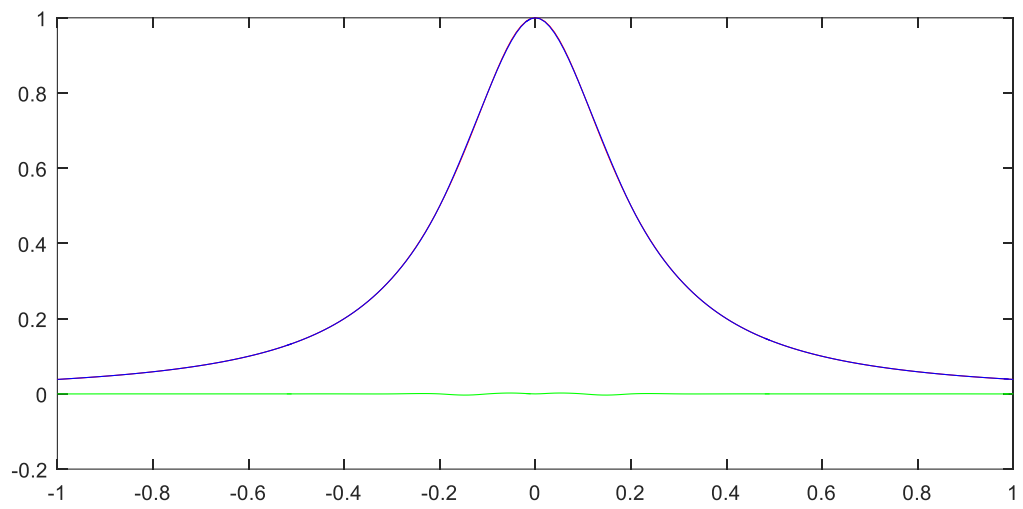


Picture2-3.  $Linear_{30}(x)$

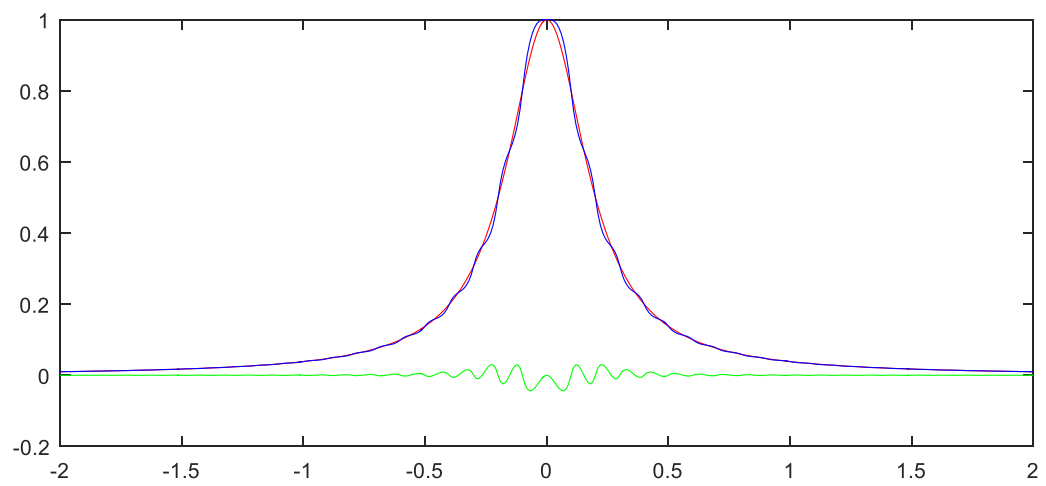




Picture3-1.  $Spline_{10}(x)$  on  $[-1, 1]$



Picture3-2.  $Spline_{20}(x)$  on  $[-1, 1]$



Picture3-3.  $Spline_{40}(x)$  on  $[-2, 2]$