数值分析上机习题报告 (4)

张宏毅 1500017736

March 15, 2017

1 Problem

对 Runge 函数 $R(x) = \frac{1}{1+x^2}$ ($x \in [-5, 5]$)，利用下列条件作插值逼近，并与 $R(x)$ 的图像作比较：（5）用等距节点 $x_i = -5 + i$ ($i = 0, 1, 2, \cdots, 10$)，绘出它的三次自然样条插值函数的图像。

2 Principles

条件五为三次自然样条插值，即求一个满足插值条件 $S(x_i) = y_i (0 \le i \le n)$ 的二阶连续可微的分段三次多项式 $S(x)$。记 $S_i(x)$ 为 $S(x)$ 在区间 $[x_i, x_{i+1}]$ 上的限制，则利用 Hermite 插值的思想，设

$$S_i(x) = y_i \alpha_i(x) + y_{i+1} \alpha_{i+1}(x) + m_i \beta_i(x) + m_{i+1} \beta_{i+1}(x),$$

其中 m_i 为 $S(x)$ 在 x_i 处的一阶导数（未知），且

$$\begin{cases} \alpha_j(x) = [1 - 2(x - x_j) l_j'(x_j)] \cdot l_j^2(x) \\ \beta_j(x) = (x - x_j) l_j^2(x) \end{cases}, \quad (j = i, i+1)$$

是三次多项式空间 $\mathbb{P}_3[x_i, x_{i+1}]$ 的一组基（$l_j(x)$ 为 x_j 处的 Lagrange 插值基函数）。这样的函数形式已经保证了 $S(x)$ 一阶连续可微，则为使 $S(x)$ 成为自然样条，只需保证

$$\begin{cases} S_0''(x_0) = S_{n-1}''(x_n) = 0, \\ S_{i-1}''(x_i) = S_i''(x_i), \quad (1 \le i \le n-1). \end{cases}$$

引入记号 $\delta_i = 1/(x_{i+1} - x_i)$，则

$$\alpha_i''(x) = 2\delta_i^2 [1 + 4\delta_i (x - x_{i+1}) + 2\delta_i (x - x_i)],$$
$$\beta_i''(x) = 4\delta_i^2 (x - x_{i+1}) + 2\delta_i^2 (x - x_i).$$

交换下标 i 和 $i+1$ 即求得 $\alpha_{i+1}''(x)$ 和 $\beta_{i+1}''(x)$。代入方程 $S_{i-1}''(x_i) = S_i''(x_i)$ 化简可得

$$\delta_{i-1} m_{i-1} + 2(\delta_{i-1} + \delta_i) m_i + \delta_i m_{i+1} = 3\delta_i^2 y_{i+1} + 3(\delta_{i-1}^2 - \delta_i^2) y_i - 3\delta_{i-1}^2 y_{i-1}.$$

于是有关于 $m_i(0 \le i \le n)$ 的线性方程组

$$\begin{bmatrix} 2 & \lambda_0 & & & \\ 1-\lambda_1 & 2 & \lambda_1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1-\lambda_{n-1} & 2 & \lambda_{n-1} \\ & & & 1-\lambda_n & 2 \end{bmatrix} \begin{bmatrix} m_0 \\ m_1 \\ \vdots \\ m_{n-1} \\ m_n \end{bmatrix} = \begin{bmatrix} \mu_0 \\ \mu_1 \\ \vdots \\ \mu_{n-1} \\ \mu_n \end{bmatrix},$$

其中对 $1 \le i \le n-1$ 有

$$\lambda_i = \delta_i/(\delta_{i-1}+\delta_i),$$
$$\mu_i = 3[(1-\lambda_i)\delta_{i-1}(y_i - y_{i-1}) + \lambda_i\delta_i(y_{i+1}-y_i)].$$

以及有自然边界条件 $\lambda_0 = 1$，$\lambda_n = 0$，$\mu_0 = 3\delta_0(y_1 - y_0)$，$\mu_n = 3\delta_{n-1}(y_n - y_{n-1})$。通过解该三对角矩阵方程组即可求得 $S(x)$ 的各个一阶导数，进而写出整个自然样条函数。具体代码如下：

```
def NaturalSpline(x, y, n):
    n -= 1
    delta = [1/(x[1]-x[0])]
    lam = [1]
    mu = [3*delta[0]*(y[1]-y[0])]

    # Compute the equation matrix
    for i in range(1, n):
        delta.append(1/(x[i+1]-x[i]))
        lam.append(delta[i]/(delta[i-1]+delta[i]))
        mu.append(3*(1-lam[i])*delta[i-1]*(y[i]-y[i-1])+3*lam[i]*delta[i]*(y[i+1]-y[i]))
    mu.append(3*delta[n-1]*(y[n]-y[n-1]))
    _lam = [1-t for t in lam[1:n]] + [1]

    # Solve derivatives at each point
    m = SolveTridiag(_lam, [2] * (n+1), lam, mu, n+1)

    # Compute the natural spline function
    def InterpPoly(t):
        for i in range(0, n):
            if x[i] <= t <= x[i+1]:
                d = x[i+1]-x[i]
                ans = y[i] * (1+2*(t-x[i])/d) * ((t-x[i+1])/d)**2
                ans += y[i+1] * (1-2*(t-x[i+1])/d) * ((t-x[i])/d)**2
                ans += m[i] * (t-x[i]) * ((t-x[i+1])/d)**2
                ans += m[i+1] * (t-x[i+1]) * ((t-x[i])/d)**2
                return ans
    return InterpPoly
```

3 Results

画出求得的三次自然样条插值函数图像如图 1 所示。由于信息的缺少，插值函数的计算量有了明显的增加，且如果将其与三次 Hermite 插值函数的图像做对比，可以发现自然样条的插值效果略逊于

Hermite 插值，这也是一阶导数信息的缺失所引起的。但同时，自然样条的优点也在于。只需要较少的信息，就能拟合出一个光滑性相当不错的函数作为逼近。

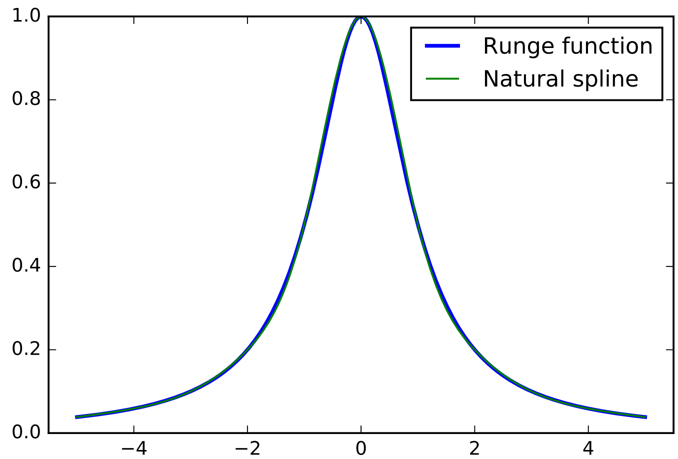

图 1: Runge 函数及其自然三次样条插值的图像