Lecture Notes for Math 5630/6630 **Fall 2024**

Note 9: Spline Interpolation

Tags: math.na *Date: 10/08/2024*

Disclaimer: *This lecture note is for math 5630/6630 class only.*

Let us review polynomial interpolation's error estimate

$$f(x) - p_n(x) = \frac{f^{(n+1)}(\zeta)}{(n+1)!}\prod_{i=0}^{n}(x - x_i).$$

The error estimate may not be small if $f^{(n+1)}$ grows fast as $n \to \infty$ (Runge Phenomenon), thus high-order polynomial interpolation is often a "bad" idea for practical problems.

1 Spline Interpolation

If we cannot afford a large degree n, then polynomial interpolation can only be applied to short intervals. Thus it is possible to use polynomial interpolation only locally (on subintervals). Globally, this constructs a *piecewise polynomial interpolation*.

Let the interval be divided into smaller pieces

$$a = x_0 < x_1 < \cdots < x_n = b.$$

and use a low-degree polynomial interpolation in each subinterval $[x_i, x_{i+1}]$, $i = 0, \cdots, n-1$. The polynomial pieces, denoted by $\phi_i(x)$, are then *glued* together to construct a **continuous** function. This function, denoted by $s_k(x)$, is a spline of degree k if

$$s_k(x) = \phi_i(x) \in \Pi_k, \quad x \in [x_i, x_{i+1}], \quad i = 0, 1, \cdots, n-1.$$

and $s_k \in C^{k-1}([a, b])$. The spline function s_k is $(k-1)$ times continuously differentiable and piecewise polynomial of degree k.

Then the space of splines s_k will be $(n+k)$ dimension: each interval has $(k+1)$ dimensions, each interface imposes k constraints, therefore $n(k+1) - (n-1)k = n+k$ dimensions. This shows that to determine a spline on the nodes uniquely, we will require $n+1$ interpolation values and $k-1$ additional constraints. Usual choices are

1. periodic splines. $s_k^{(m)}(a) = s_k^{(m)}(b)$ for $m = 0, 1, \ldots, k-1$.
2. natural splines. $s_k^{(l+j)}(a) = s_k^{(l+j)}(b) = 0$, $j = 0, 1, \ldots, l-2$ and $k = 2l-1$ with $l \geq 2$.

In the following, we discuss some useful examples of spline.

1.1 Linear Spline

The simplest spline is the linear spine, that is, $\deg(\phi_i) = 1$, $i = 0, 1, \cdots, n-1$.

On each interval $[x_i, x_{i+1}]$, the function $s(x)$ is defined by

$$s_k(x) = \phi_i(x) = f(x_i) + f[x_i, x_{i+1}](x - x_i), \quad x \in [x_i, x_{i+1}]$$

Geometrically, the $s_k(x)$ graph connects the data points with short segments.

The interpolation error can be derived from the previous error formula for two interpolation nodes. Let $h = \max_{1 \le i \le n}(t_i - t_{i-1})$. On the interval $[x_{i-1}, x_i]$, the error of interpolation is

$$|f(x) - s_1(x)| \le \frac{\max_{\zeta \in [x_{i-1}, x_i]} |f''(\zeta)|}{2!} |(x - x_{i-1})(x - x_i)| \le \frac{\max_{\zeta \in [x_{i-1}, x_i]} |f''(\zeta)|}{8} h^2.$$

Remark 1.1. *Once $|f''|$ is not uniformly bounded, the interpolation error will be replaced by the modulus of continuity.*

1.2 Cubic Spline

The cubic splines are particularly important in practice. Let $a = x_0 < x_1 < \cdots < x_n = b$, and the corresponding values are y_j, $j = 0, \ldots, n$. The constraints for cubic splines are: piecewise polynomial of degree 3 and continuous second derivative.

Denote the interpolation spline as s_3, then s_3'' is a piecewise linear function. On the sub-interval $[x_{j-1}, x_j]$, it can be represented by

$$s_3''(x) = M_{j-1} \frac{x_j - x}{h_j} + M_j \frac{x - x_{j-1}}{h_j}, \quad j = 1, \ldots, n,$$

where $h_j = x_j - x_{j-1}$, $M_j = s_3''(x_j)$. Integrating the above formula twice,

$$s_3(x) = M_{j-1} \frac{(x_j - x)^3}{6h_j} + M_j \frac{(x - x_{j-1})^3}{6h_j} + A_j(x - x_{j-1}) + B_j$$

The additional constants A_j, B_j can be determined by imposing $f(x_{j-1}) = y_{j-1}$ and $f(x_j) = y_j$. That is

$$A_j = \frac{y_j - y_{j-1}}{h_j} - \frac{h_j}{6}(M_j - M_{j-1}), \quad B_j = y_{j-1} - M_{j-1} \frac{h_j^2}{6}.$$

Now we will determine the constants M_j using the first derivative's continuity.

$$s_3'(x_j^-) = s_3'(x_j^+), \quad j = 1, \ldots, n-1.$$

That is equivalent to $j = 1, \ldots, n-1$,

$$\begin{aligned} s_3'(x_j^-) &= M_j \frac{h_j}{3} + M_{j-1} \frac{h_j}{6} + \frac{y_j - y_{j-1}}{h_j} \\ &= -M_j \frac{h_{j+1}}{3} - M_{j+1} \frac{h_{j+1}}{6} + \frac{y_{j+1} - y_j}{h_{j+1}} = s_3'(x_j^+). \end{aligned} \tag{$\dagger$}$$

We can write the corresponding equations into a tridiagonal linear system

$$\begin{pmatrix} \frac{h_1}{6} & \frac{h_1+h_2}{3} & \frac{h_2}{6} & & & \\ & \frac{h_2}{6} & \frac{h_2+h_3}{3} & \frac{h_3}{6} & & \\ & & \ddots & \ddots & \ddots & \\ & & & \frac{h_{n-1}}{6} & \frac{h_{n-1}+h_n}{3} & \frac{h_n}{6} \end{pmatrix} \begin{pmatrix} M_0 \\ M_1 \\ \vdots \\ M_n \end{pmatrix} = \begin{pmatrix} \frac{y_2-y_1}{h_2} - \frac{y_1-y_0}{h_1} \\ \frac{y_3-y_2}{h_3} - \frac{y_2-y_1}{h_2} \\ \vdots \\ \frac{y_n-y_{n-1}}{h_n} - \frac{y_{n-1}-y_{n-2}}{h_{n-1}} \end{pmatrix}$$

In practice, the system will be rescaled for numerical stability.

$$\begin{pmatrix} \frac{h_1}{2(h_1+h_2)} & 1 & \frac{h_2}{2(h_1+h_2)} & & & \\ & \frac{h_2}{2(h_2+h_3)} & 1 & \frac{h_3}{2(h_2+h_3)} & & \\ & & \ddots & \ddots & \ddots & \\ & & & \frac{h_{n-1}}{2(h_{n-1}+h_n)} & 1 & \frac{h_n}{2(h_{n-1}+h_n)} \end{pmatrix} \begin{pmatrix} M_0 \\ M_1 \\ \vdots \\ M_n \end{pmatrix} = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_{n-1} \end{pmatrix}$$

where $d_j = \frac{3}{h_{j-1}+h_j}\left[\frac{y_j-y_{j-1}}{h_j} - \frac{y_{j-1}-y_{j-2}}{h_{j-1}}\right]$. The above system still lacks 2 more constraints, since the matrix is of size $(n-1)\times(n+1)$. Then we can apply the periodic spline or natural spline conditions. For example, if the natural constraint is applied: ${s_3}''(a) = {s_3}''(b) = 0$. We should have two more equations:

$$M_0 = M_n = 0.$$

Then we can simply ignore the first and last columns of the matrix (also M_0 and M_n). If the periodic constraint is imposed, then we can add two more constraints: $M_0 = M_n$ and

$$-M_0\frac{h_1}{3} - M_1\frac{h_1}{6} + \frac{y_1-y_0}{h_1} = M_n\frac{h_n}{3} + M_{n-1}\frac{h_n}{6} + \frac{y_n-y_{n-1}}{h_n}.$$

In both cases, the resulting linear system is still tridiagonal and the solution takes $\mathcal{O}(n)$ time complexity with the Thomas algorithm.

Another popular choice to complete the matrix is to impose the constraints in the similar form on x_0 and x_n:

$$2M_0 + \frac{h_1}{h_0+h_1}M_1 = d_0, \qquad \frac{h_n}{h_n+h_{n+1}}M_{n-1} + 2M_n = d_n,$$

where $h_0 = h_{n+1} = 0$ and $d_0 = d_1$, $d_n = d_{n-1}$ are assumed.

The error estimate for the cubic spline can be derived in a way similar to the Lagrange polynomial interpolation. The following result is attributed to Charles Hall (1968).

Theorem 1.2. *Let $f \in C^4([a,b])$ and $a = x_0 < \cdots < x_n = b$ be a set of nodes. Then the natural cubic spline s_3 interpolating f satisfies*

$$\|f - s_3\|_\infty \leq \frac{5}{384}\|f^{(4)}\|_\infty h^4,$$

where $h = \max_j |x_j - x_{j-1}|$.

Proof. Here we only state the rough idea to prove the error bound. Let $u(x)$ be the piecewise Hermite interpolation polynomial that

$$u(x_j) = f(x_j), \quad u'(x_j) = f'(x_j),$$

then one can estimate

$$\max_{x\in[x_j,x_{j+1}]} |u - f| \leq \frac{1}{24}\|f^{(4)}\|_\infty (x-x_j)^2(x-x_{j+1})^2 \leq \frac{1}{384}\|f^{(4)}\|_\infty h^4.$$

On the subinterval $[x_i, x_{i+1}]$, s_3 and u are both cubic polynomial interpolations, thus

$$u(x) - s_3(x) = \frac{(x-x_i)(x_{i+1}-x)}{(x_{i+1}-x_i)}\left(e'(x_i)\frac{x_{i+1}-x}{x_{i+1}-x_i} - e'(x_{i+1})\frac{x-x_i}{x_{i+1}-x_i}\right),$$

where $e(x) = f(x) - s_3(x)$. Therefore,

$$\begin{aligned}\|f - s_3\|_\infty &\leq \|u - s_3\|_\infty + \|f-u\|_\infty \\ &\leq \frac{h}{4}\max_{0\leq i\leq n}|e'(x_i)| + \frac{1}{384}\|f^{(4)}\|_\infty h^4.\end{aligned} \tag{$*$}$$

Using (†), we find that

$$\frac{1}{h_j}\left[2s_3'(x_j) + s_3'(x_{j-1})\right] + \frac{1}{h_{j+1}}\left[2s_3'(x_j) + s_3'(x_{j+1})\right] = \frac{3(y_j - y_{j-1})}{h_j^2} + \frac{3(y_{j+1}-y_j)}{h_{j+1}^2},$$

Using Taylor expansion locally at x_j, there exist $\zeta \in (x_{j-1}, x_j)$ and $\xi \in (x_j, x_{j+1})$ that

$$\frac{2e'(x_j) + e'(x_{j-1})}{h_j} + \frac{2e'(x_j) + e'(x_{j+1})}{h_{j+1}} = \frac{1}{24}\left[-h_j^2 f^{(4)}(\zeta) + h_{j+1}^2 f^{(4)}(\xi)\right].$$

Suppose $\max_{0\leq i\leq n}|e'(x_i)|$ attains its maximum at node x_k, then

$$\left|\frac{2e'(x_k) + e'(x_{k-1})}{h_j} + \frac{2e'(x_k) + e'(x_{k+1})}{h_{k+1}}\right| \geq \frac{h_j + h_{j+1}}{h_j h_{j+1}}|e'(x_k)|.$$

Therefore, by AM-GM inequality,

$$\max_{0\leq i\leq n}|e'(x_i)| \leq \frac{h_j h_{j+1}}{24(h_j + h_{j+1})}(h_j^2 + h_{j+1}^2)\|f^{(4)}\|_\infty \leq \frac{1}{24}h^3\|f^{(4)}\|_\infty.$$

Finally, combined with the estimate $(*)$ will arrive at the desired bound. □