Chapter 4

Polynomial Interpolation

4.1 Lagrange Interpolation

Definition 4.1.1. Interpolation of a given function f defined on an interval $[a, b]$ by a polynomial p: Given a set of specified points $\{(x_i, f(x_i)\}_{i=0}^n$ with $\{x_i\} \subset [a, b]$, the polynomial p of degree n satisfying

$$p(x_i) = f(x_i), \quad i = 0, \cdots, n$$

is called an **polynomial interpolation**. The points $\{x_i\}$ are called **nodes**.

We let $P_n(x)$ be the set of all polynomial of degree less or equal to n. Nonpolynomial interpolation can be defined, but rarely used.

Here we shall use the semi-norm to measure the error:

$$|f|_1 \equiv \sum_{i=0}^{n} |f(x_i)|.$$

Theorem 4.1.2. *For given f, there exist a unique $p \in P_n(x)$ such that $|f - p|_1 = 0$.*

Proof. If $p = a_0 + a_1 x + \cdots + a_n x^n$. We want to find a polynomial of the form $p = a_0 + a_1 x + \cdots + a_n x^n$ such that $p(x_i) = f(x_i)$, for $i = 0, \ldots, n$, i.e,

$$\begin{aligned} p(x_0) &= a_0 + a_1 x_0 + \cdots + a_n x_0^n = f(x_0) \\ &= \cdots \\ p(x_n) &= a_0 + a_1 x_n + \cdots + a_n x_n^n = f(x_n). \end{aligned}$$

In matrix form,

$$\begin{bmatrix} 1 & x_0 & \cdots & x_0^n \\ 1 & x_1 & \cdots & x_1^n \\ & & \cdots & \\ 1 & x_n & \cdots & x_n^n \end{bmatrix} \begin{bmatrix} a_0 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} f(x_0) \\ \vdots \\ f(x_n) \end{bmatrix} \tag{4.1}$$

The equation (4.1) involves a Van der Monde matrix whose determinant is $\prod_{i>j}(x_i - x_j)$. Thus we see a unique solution exists as long as the interpolation points are distinct. □

A basis for $V = P_n$

A naive basis for $P_n(x)$ is $\{1, x, x^2, \ldots, x^n\}$. Is it convenient? No! There are other bases, and usually other bases are better.

(1) To compute the coefficients in (4.1), one has to solve a linear system.

(2) Moreover, the Van der Monde matrix is ill-conditioned.

Example 4.1.3. Consider

$$\begin{array}{rcrcl} x & & -y & = & 1+\epsilon \\ (1+10^{-9})x & & -y & = & 1. \end{array} \tag{4.2}$$

Without ϵ, the exact solution is $x = 0, y = -1$, while with ϵ perturbation, $x = -10^9\epsilon$, $y = -10^9\epsilon - 1 - \epsilon$!

Why Polynomial Interpolation ?

For example consider e^x or $\sin x$. How to evaluate or integrate them?

(1) On a computer we approximate a given function by only arithmetic operations which is done by polynomial interpolation. Taylor series is rarely used. So Find a polynomial $p(x)$ s.t. $p(x_i) = e^{x_i}$ for $i = 1, 2, \cdots$. This is better than Taylor series which loses accuracy when x is large. (non uniform error) Taylor series requires higher order derivatives which may be difficult to obtain or unavailable.

(2) It is easier to handle Polynomial interpolations systematically (in solving the p.d.e. or integrations)

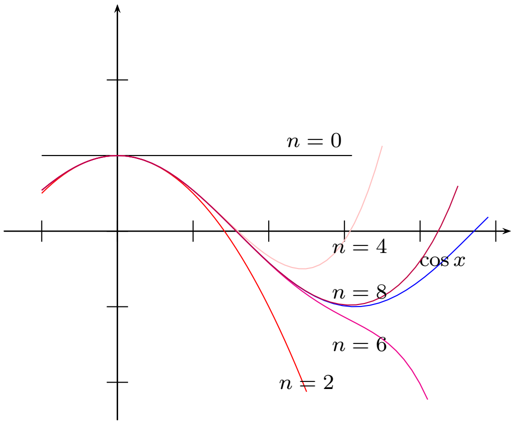

Figure 4.1: Taylor expansion of $\cos x$ up to p_8

Lagrange basis

Suppose we wish to construct a linear interpolation with two points x_0 and x_1. Define

$$L_0(x) = \frac{x - x_1}{x_0 - x_1}, \quad L_1(x) = \frac{x - x_0}{x_1 - x_0}.$$

Then

$$\begin{aligned} L_0(x_0) = 1, &\quad L_0(x_1) = 0 \\ L_1(x_0) = 0, &\quad L_1(x_1) = 1. \end{aligned}$$

Now given any data $f(x_0) = y_0$, $f(x_1) = y_1$, we can construct a linear interpolant by

$$\ell(x) = y_0 L_0(x) + y_1 L_1(x). \tag{4.3}$$

We now generalize this. Given distinct nodes $x_0, x_1, \cdots, x_n$, we construct polynomials $L_{n,i}(x)(i = 0, \cdots, n)$ such that $L_{n,i}(x_j) = \delta_{ij}, \quad j = 0, \cdots, n$. We can easily see that $L_{n,i}$ has the following form:

$$L_{n,i}(x) = C \prod_{j \neq i}^{n} (x - x_j), \text{ for some } C.$$

We set

$$L_{n,i}(x_i) \quad = \quad C \prod_{j \neq i} (x_i - x_j) = 1.$$

Then we obtain $C = 1/\prod_{j\neq i}(x_i - x_j)$ and hence

$$L_{n,i}(x) = \prod_{j\neq i} \frac{(x - x_j)}{(x_i - x_j)}.$$

These are the **Lagrange basis polynomials**. Now using these, one can readily construction a polynomial interpolation.

Proposition 4.1.4. *Let $f \in C[a,b]$ and let p_n be the unique element of $P_n(x)$ such that $f(x_i) = p_n(x_i)$, $i = 0, 1, \ldots, n$. Then the* **Lagrange interpolating polynomial** *is given by*

$$p_n(x) = \sum_{i=0}^{n} f(x_i) \prod_{j\neq i} \frac{(x - x_j)}{(x_i - x_j)}.$$

(1) $\|f(x) - p_n(x)\|_\infty$?

(2) What happens if nodes are close?

(3) $\lim_{n\to\infty} p_n(x) =$?

Polynomial of degree n has the tendency of $n - 1$ oscillation(where the derivative vanishes). Thus, if the interval is fixed and n becomes larger, it may oscillate. Indeed the following example by Runge shows it.

Example 4.1.5 (Runge)**.**

$$f(x) = \frac{1}{1 + x^2} \qquad \text{on} \quad [-5, 5]$$

Given $\{(x_i, f(x_i)) \mid i = 0, 1, \ldots, n\}$

$$h = \frac{b - a}{2n}, \qquad x_k = kh, \quad k = -n, \ldots, n$$

Choose $n = 5, 10, \cdots$ for example, and interpolate $f(x)$ by polynomial of degree $2n$.

$$p_{2n}(\frac{5}{n}k) = f(\frac{5k}{n}), \qquad k = 0, \pm1, \ldots, \pm n.$$

Runge showed $\lim_{n\to\infty} \|f(x) - p_{2n}(x)\| = \infty$.

Thus Runge's example shows higher degree polynomial is not always good for interpolation. This suggests us to use lower degree polynomial on each subinterval.

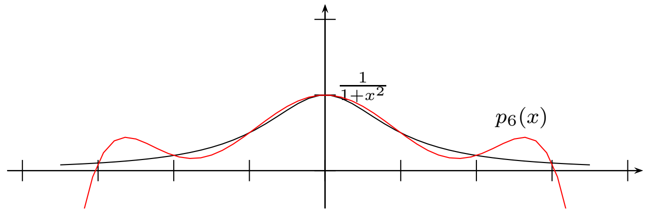

Figure 4.2: Runge function and a polynomial interpolation

Let $f(x)$ on $[-3,3]$ with $n=3$. Then

$$
\begin{aligned}
p_6(x) \quad = \quad & f(x_0)\frac{(x+2)(x+1)x(x-1)(x-2)(x-3)}{(-3+2)(-3+1)(-3)(-4)(-5)(-6)} \\
& +f(x_1)\frac{(x+3)(x+1)x(x-1)(x-2)(x-3)}{(-2+3)(-2+1)(-2)(-3)(-4)(-5)} \\
& +f(x_2)\frac{(x+3)(x+2)x(x-1)(x-2)(x-3)}{(-1+3)(-1+2)(-1)(-2)(-3)(-4)} \\
& +f(x_3)\frac{(x+3)(x+2)(x+1)(x-1)(x-2)(x-3)}{(0+3)(0+2)1(-1)(-2)(-3)} \\
& +f(x_4)\frac{(x+3)(x+2)(x+1)x(x-2)(x-3)}{(1+3)(1+2)(1+1)(1)(-1)(-2)} \\
& +f(x_5)\frac{(x+3)(x+2)(x+1)x(x-1)(x-3)}{(2+3)(2+2)(2+1)2(1)(-1)} \\
& +f(x_6)\frac{(x+3)(x+2)(x+1)x(x-1)(x-2)}{(3+3)(3+2)(3+1)3(2)(1)} \\
= \quad & 1-\frac{16x^2}{25}+\frac{3x^4}{20}-\frac{x^6}{100}
\end{aligned}
$$

Error Estimate

Theorem 4.1.6. *Let $f(x)\in C^{n+1}[a,b]$. If $a=x_0,x_1,\ldots,x_n=b$ are $n+1$ distinct points and $p_n(x)$ is the Lagrange interpolating polynomial, then there exists a function $\xi(x)\in(a,b)$ such that*

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

Proof. Let $d(x) = \prod_{i=0}^{n}(x - x_i)$ and define

$$\Phi(x) \equiv \frac{f(x) - p_n(x)}{d(x)}. \tag{4.4}$$

Let x be **fixed** number different from all x_i's. Then the function defined by

$$\Omega(z) = f(z) - p_n(z) - d(z)\Phi(x) \in C^{(n+1)}[a, b]$$

vanishes(as a function of z) at $x_i, i = 0, \cdots, n$ ($n+1$ nodal points). But it also vanishes at x by (4.4). Thus $\Omega(z)$ vanishes at $n + 2$ distinct points in (a, b). Thus by Rolle's Theorem,

$$\begin{array}{ll} \Omega'(z) & \text{has } n+1 \text{ distinct zeros in } (a,b) \\ \Omega''(z) & \text{has } n \text{ distinct zeros in } (a,b) \\ & \cdots \\ \Omega^{(n+1)}(z) & \text{has at least one zero in } (a,b). \end{array}$$

Thus there exists a point $\xi(x)$ such that

$$\Omega^{(n+1)}(\xi(x)) = 0 = f^{(n+1)}(\xi) - (n+1)!\,\Phi(x).$$

Hence

$$\begin{aligned} \Phi(x) &\equiv \frac{f(x) - p_n(x)}{d(x)} = \frac{f^{(n+1)}(\xi)}{(n+1)!} \\ \therefore \qquad f(x) - p_n(x) &= \frac{f^{(n+1)}(\xi(x))}{(n+1)!} \prod_{i=0}^{n}(x - x_i). \end{aligned}$$

□

Newton's form of interpolating polynomial

We now describe an efficient way of calculating the coefficients of an interpolating polynomial.

There are three typical basis for polynomial space. First,

$$\{1, x, x^2, \cdots, x^n\}$$

is the natural basis. Next, with $L_i(x) = \prod_{j \neq i}^{j=n} \frac{x - x_j}{x_i - x_j}$ the Lagrangian basis is

$$\{L_0, L_1, \cdots, L_n\}$$

The Lagrangian form is useful for analysis, but not efficient for computation. In the Newton form of interpolating polynomial the following basis is taken:

$$\{1, (x - x_0), (x - x_0)(x - x_1), \cdots, (x - x_0)(x - x_1) \cdots (x - x_{n-1})\}$$

We compute $p_n(x)$ by induction. Since $p_n(x)$ is one degree higher than $p_{n-1}(x)$, one can set

$$p_n(x) = p_{n-1}(x) + q_n(x)$$

with $q_n(x_j) = 0$, $j = 0, 1, \ldots, n-1$. Thus $q_n(x) = a_n \prod_{i=0}^{n-1}(x - x_i)$ and

$$\begin{aligned}
p_n(x) &= p_{n-1}(x) + a_n \prod_{i=0}^{n-1}(x - x_i) \\
&= p_{n-2}(x) + a_{n-1} \prod_{i=0}^{n-2}(x - x_i) + a_n \prod_{i=0}^{n-1}(x - x_i) \\
&\cdots \\
&= a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \cdots + a_n \prod_{i=0}^{n-1}(x - x_i).
\end{aligned}$$

If we denote $f[x_0, \ldots, x_k]$ for a_k, then

$$p_n(x) = \sum_{k=0}^{n} a_k \prod_{i=0}^{k-1}(x - x_i) \equiv \sum_{k=0}^{n} f[x_0, \ldots, x_k] \prod_{i=0}^{k-1}(x - x_i). \tag{4.5}$$

This is called the **Newton's form of interpolation** and $f[x_0, \ldots, x_k]$ are called the **divided difference**. Now we study how to compute a_k.

Computing the divided difference

Now we show how to compute $f[x_0, \ldots, x_k]$ efficiently. One can write $p_n(x)$ in two ways(*by reordering the points starting from* x_n),

$$\begin{aligned}
p_n(x) &= a_0 + a_1(x - x_0) + \cdots + a_n(x - x_0)(x - x_1) \cdots (x - x_{n-1}) \\
p_n(x) &= b_0 + b_1(x - x_n) + \cdots + b_n(x - x_n)(x - x_{n-1}) \cdots (x - x_1),
\end{aligned}$$

where $b_n = f[x_n, \ldots, x_0]$ by definition. Comparing the highest order term, we get

$$a_n = f[x_0, \ldots, x_n] = b_n = f[x_n, \ldots, x_0].$$

Reordering the points, we also see that

$$b_n := f[x_n, \ldots, x_0] = f[x_{i(0)}, \ldots, x_{i(n)}]$$

for any permutation $i(k)$ of numbers $\{0, 1, \cdots, n\}$. Hence the **divided difference** is independent of the order of its arguments x_i.

Subtracting the two expressions for the same polynomial p_n,

$$0 = a_n[(x - x_0) - (x - x_n)](x - x_1) \cdots (x - x_{n-1}) + (a_{n-1} - b_{n-1})x^{n-1} + \cdots$$

Comparing the coefficients of x^{n-1}, we see

$$a_n(x_n - x_0) + (a_{n-1} - b_{n-1}) = 0.$$

Since $b_{n-1} = f[x_n, \ldots, x_1] = f[x_1, \ldots, x_n]$ and $a_{n-1} = f[x_0, \ldots, x_{n-1}]$($\because$ b_{n-1} is defined using n points $x_n, \cdots x_1$ and a_{n-1} is defined using n points $x_0, \cdots x_{n-1}$), we see

$$a_n = f[x_0, \ldots, x_n] = \frac{b_{n-1} - a_{n-1}}{x_n - x_0} = \frac{f[x_1, \ldots, x_n] - f[x_0, \ldots, x_{n-1}]}{x_n - x_0}.$$

Thus Newton's formula is useful when a new point of interpolation is added to an existing interpolation.

Example 4.1.7. Suppose we are given $x_0, \ldots, x_n$ and p_n. If we have one more point x_{n+1}. Then p_{n+1} is constructed by adding one more term to p_n:

$$p_n(x) = p_{n-1}(x) + f[x_0, \ldots, x_{n-1}, x_n] \prod_{i=0}^{n-1} (x - x_i). \tag{4.6}$$

Example 4.1.8. (1) $f[x_i] = f(x_i)$.

(2) $f[x_0, x_1] = \frac{f[x_1] - f[x_0]}{x_1 - x_0} = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$.

(3) $f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0}$.

(4) $p_2(x) = f(x_0) + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1)$.

$$\begin{array}{ll|lll}
x_0 & f[x_0] & f[x_0,x_1] & f[x_0,x_1,x_2] & f[x_0,x_1,x_2,x_3] \\
x_1 & f[x_1] & f[x_1,x_2] & f[x_1,x_2,x_3] & \\
x_2 & f[x_2] & f[x_2,x_3] & & \\
x_3 & f[x_3] & & &
\end{array}$$

4.2 Piecewise linear approximation

We have seen from the Runge example, that higher degree polynomial can be a bad choice. We thus consider a different method. One is to use piecewise linear functions, another is spline functions.

Let $f \in C^0[a,b]$ and let $K = \{x_0, x_1, \ldots, x_n\} \subset [a,b]$. Define $V \equiv PL(a,b\,;\,K)$: the set of continuous functions which is piecewise linear on each subintervals (x_i, x_{i+1}). Then $\dim V = n+1$ and basis functions are

$$T_i(x) = \begin{cases} \dfrac{x - x_{i-1}}{x_i - x_{i-1}} & \text{on } [x_{i-1}, x_i] \\ \dfrac{x - x_{i+1}}{x_i - x_{i+1}} & \text{on } [x_i, x_{i+1}] \\ 0 & \text{elsewhere.} \end{cases}$$

In this case $\exists$ unique $p \in V$ such that $\|f - p\|_t = 0$.

Given $\{(x_i, f(x_i)\}_{i=0}^m$ we want to approximate $f(x)$ for $x \notin \{x_0, \ldots, x_m\}$ by some piecewise linear approximation. It will be good if

(1) f is continuous.

(2) $x_{k+1} - x_k$ are small for $k = 0, \ldots, m-1$.

(3) $f''(x)$ (curvature) is small.

For $x \in (x_k, x_{k+1})$ define

$$\begin{aligned} F(x) &\equiv f(x_k) + \frac{f(x_{k+1}) - f(x_k)}{x_{k+1} - x_k}(x - x_k) \\ &= \frac{x_{k+1} - x}{x_{k+1} - x_k} f(x_k) + \frac{x - x_k}{x_{k+1} - x_k} f(x_{k+1}) \\ &= w_0(x) f(x_k) + w_1(x) f(x_{k+1}), \quad w_0 + w_1 = 1,\ 0 \le w_0, w_1 \le 1 \end{aligned}$$

Then $\lim_{m\to\infty} F(x) = f(x)$ if mesh $\to 0$ uniformly and $f \in C^0[a,b]$.

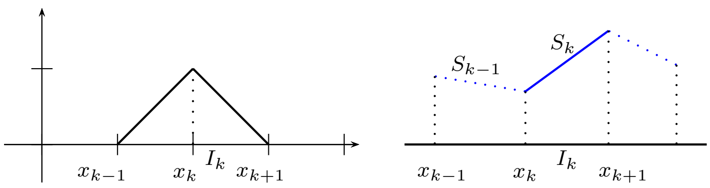

Figure 4.3: Piecewise Linear basis $\Lambda_k(x)$ and $S''_k(x)$

Proof.

$$
\begin{aligned}
f(x) - F(x) &= f(x) - w_0(x)f(x_k) - w_1(x)f(x_{k+1}) \\
&= w_0[f(x) - f(x_k)] + w_1[f(x) - f(x_{k+1})] \\
|f(x) - F(x)| &\leq |w_0|\,|f(x) - f(x_k)| + |w_1|\,|f(x) - f(x_{k+1})] \\
&\leq \max\{|f(x) - f(x_k)|, |f(x) - f(x_{k+1})|\}
\end{aligned}
$$

□

Moreover, we have the following result.

Theorem 4.2.1. *Let $f(x) \in C^2[a,b]$, $x \in (a,b)$. If $F(x) = f(a) + \frac{f(b)-f(a)}{b-a}(x-a)$ then $\exists\ c(x) \in (a,b)$ such that*

$$f(x) - F(x) = \frac{(x-a)(x-b)}{2} f''(c(x)).$$

Moreover, if $|f''(x)| \leq M_2$, then $|f(x) - F(x)| \leq \frac{(b-a)^2}{8} M_2$.

4.3 Piecewise Cubic Approximation

The drawback of piecewise linear approximation is that it is not differentiable. Thus we may try piecewise quadratic approximation. But we will soon there are some problems.

Example 4.3.1. Suppose on each subinterval $[x_i, x_{i+1}]$ we have data

$$\{f(x_0), f'(x_0), f(x_1), f'(x_1)\}.$$

Then we can find a cubic polynomial which fits the given data.(called **piecewise Hermite interpolations**) Even though there are some ad hoc choices of degrees of freedom for piecewise quadratics, they are seldom used. In general, we can consider odd degree piecewise polynomials.

In applications, there are instances to require twice differentiable functions, because it has the physical meaning of acceleration. However, derivative information usually is not provided in advance. Thus we consider

4.3.1 Piecewise polynomials without derivative information

Consider the problem of interpolating the data $\{(x_0, f(x_0), (x_1, f(x_1), (x_2, f(x_2))\}$ by a C^2 piecewise polynomial. We must have a polynomial p_1 on $[x_0, x_1]$ and another polynomial p_2 on $[x_1, x_2]$ such that

$$
\begin{aligned}
p_1(x_0) &= f(x_0),\ p_1(x_1) = f(x_1) \\
p_1'(x_1) &= p_2'(x_1),\ p_1''(x_1) = p_2''(x_1) \\
p_2(x_1) &= f(x_1),\ p_2(x_2) = f(x_2).
\end{aligned}
$$

We can use quadratic polynomials. But if we have three intervals, the extra conditions are

$$
\begin{aligned}
p_3(x_2) &= f(x_2),\ p_3(x_3) = f(x_3) \\
p_2'(x_2) &= p_3'(x_2),\ p_2''(x_2) = p_2''(x_2).
\end{aligned}
$$

Hence the quadratic would not work.

In general, we have data $\{(x_0, y_0), (x_1, y_1), \cdots, (x_n, y_n)\}$ and we use cubic in each subinterval. Count the unknowns: We have $4n$ parameters. How many conditions?

- Each of n cubic function interpolates at two end points gives $2n$
- Two derivative conditions at each $n - 1$ internal points, which gives $2(n - 1)$ conditions.

We need two more conditions to determine a unique interpolant.

Boundary conditions

Free or **natural boundary condition**:

$$p_0''(x_0) = 0,\ p_{n-1}''(x_n) = 0.$$

The **clamped boundary condition**:

$$p_0'(x_0) = s_0,\ p_{n-1}'(x_n) = s_n.$$

The **not-a knot boundary condition**:

$$p_0(x) = p_1(x),\ p_{n-2}(x) = p_{n-1}(x).$$

The last conditions are equivalent to requiring S is three times continuously differentiable.

4.4 Computation of Cubic splines

Let $y = f \in C^2[a,b]$, $x_0 = a < x_1 < \ldots < x_{n-1} < x_n = b$. We wish to construct a $C^2[a,b]$, piecewise cubic polynomial $S(x)$ on $I_k = [x_k, x_{k+1}]$ interpolating f, i.e, construct $S(x)$ satisfying

(1) $S(x_k) = y_k$ for $k = 0, \cdots, n$,

(2) $S'(x_k^-) = S'(x_k^+)$ for $k = 1, \cdots, n-1$,

(3) $S''(x_k^-) = S''(x_k^+)$ for $k = 1, \cdots, n-1$.

Thus we have $2n + 2(n-1) = 4n - 2$ conditions in $4n$ unknowns.

Now we show how to construct $S(x)$. Let $\Lambda_k(x)$ be the continuous, piecewise linear hat function on $[x_{k-1}, x_{k+1}]$ for $k = 0, \cdots, n$.(When the interval falls out of $[x_0, x_n]$, we just cut out.) We write $p_k(x) \equiv S(x)|_{I_k}$. Since $p_k''(x)$ is linear, we may write

$$S''(x) = \sum_{k=0}^{n} \sigma_k \Lambda_k(x_k). \tag{4.7}$$

On the interval $I_k = [x_k, x_{k+1}]$,$(k = 0, \cdots, n-1)$ we have

$$p_k''(x) = \sigma_k(\frac{x_{k+1} - x}{h_k}) + \sigma_{k+1}(\frac{x - x_k}{h_k}), \quad h_k = x_{k+1} - x_k. \tag{4.8}$$

Hence

$$p_k'(x) = -\sigma_k \frac{(x_{k+1} - x)^2}{2h_k} + \sigma_{k+1} \frac{(x - x_k)^2}{2h_k} + \tau_k \tag{4.9}$$

$$p_k(x) = \sigma_k \frac{(x_{k+1} - x)^3}{6h_k} + \sigma_{k+1} \frac{(x - x_k)^3}{6h_k} + \tau_k(x - x_k) + \kappa_k. \tag{4.10}$$

Since $S(x)$ interpolates $f(x)$ at x_k and x_{k+1}, we see

$$p_k(x_k) = \sigma_k \frac{h_k^2}{6} + \kappa_k = y_k, \quad p_k(x_{k+1}) = \sigma_{k+1} \frac{h_k^2}{6} + \tau_k h_k + \kappa_k = y_{k+1}.$$

Thus $\kappa_k = y_k - \frac{\sigma_k h_k^2}{6}$, $\tau_k = \frac{y_{k+1}-y_k}{h_k} - \frac{h_k}{6}(\sigma_{k+1} - \sigma_k)$. Now compute the derivative at x_k: We see from (4.9)

$$\begin{aligned} p_k'(x_k) &= -\frac{\sigma_k h_k}{2} + \tau_k & (4.11)\\ &= -\frac{\sigma_k h_k}{3} - \frac{\sigma_{k+1} h_k}{6} + (\frac{y_{k+1} - y_k}{h_k}). & (4.12) \end{aligned}$$

On the other hand, we consider $p_{k-1}'(x_k)$. Replacing k by $k-1$ in (4.9) and evaluating at x_k, we get

$$\begin{aligned} p_{k-1}'(x_k) &= \frac{\sigma_k h_{k-1}}{2} + \tau_{k-1} \\ &= \frac{\sigma_k h_{k-1}}{3} + \frac{\sigma_{k-1} h_{k-1}}{6} + (\frac{y_{k-1} - y_k}{h_{k-1}}). & (4.13) \end{aligned}$$

The continuity of derivative: Equating (4.12) with (4.13) and arranging in terms of unknowns σ_k, $k = 0, 1 \cdots, n$

$$\frac{h_{k-1}}{6}\sigma_{k-1} + \frac{h_k + h_{k-1}}{3}\sigma_k + \frac{h_k}{6}\sigma_{k+1} = \frac{y_{k+1} - y_k}{h_k} - \frac{y_k - y_{k-1}}{h_{k-1}}, \quad 1 \le k \le n-1.$$

We see two more conditions are needed to guarantee the existence of the solution. A common choice is to let $\sigma_0 = \sigma_n = 0$(The **natural spline**). In this case, we obtain a $(n-1) \times (n-1)$ tridiagonal system Do not divide the entry as in Leader book. The original system is symmetric.

$$A = \begin{bmatrix} h_0 + h_1 & h_1 & & \cdots \\ h_1 & h_1 + h_2 & h_2 & \vdots \\ \vdots & \ddots & \ddots & h_{n-2} \\ 0 & \cdots & h_{n-2} & h_{n-2} + h_{n-1} \end{bmatrix}$$

Definition 4.4.1. We say a matrix is **irreducibly diagonally dominant** if

$$|a_{ii}| \ge \sum_{j \ne i}^{n} |a_{ij}|, \text{ for all } i = 1, 2, \cdots, n$$

and a strict inequality holds at least for one i.

Hence A is irreducibly diagonally dominant and thus nonsingular. (It is known that a irreducibly diagonally dominant and thus nonsingular). For such matrix, Gauss elimination can be performed without pivoting.

This is a system of $n-1$ unknowns.(Recall we have started with $4n$ unknowns.) To find it, one has to solve a system of linear equations. Fortunately, the system is tridiagonal, which is easy to solve.

A tridiagonal system has a LU-decomposition of the following form:

$$\begin{bmatrix} b_1 & c_1 & \dots & 0 \\ a_1 & b_2 & c_2 & \vdots \\ \vdots & \ddots & \ddots & c_{n-1} \\ 0 & \dots & a_{n-1} & b_n \end{bmatrix} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ \ell_1 & 1 & 0 & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & \ell_{n-1} & 1 \end{bmatrix} \begin{bmatrix} d_1 & c_1 & \dots & 0 \\ 0 & d_2 & c_2 & \vdots \\ \vdots & \ddots & \ddots & c_{n-1} \\ 0 & \dots & 0 & d_n \end{bmatrix}$$

LU decomposition. cost: $2(n-1)$ multiplication $n-1$ addition.

$$\circ \begin{cases} d_1 = b_1 \\ \text{For } i = 1, \dots, n-1 \\ \ell_i = a_i/d_i \\ d_{i+1} = b_{i+1} - \ell_i c_i \end{cases}$$

Forward substitution $L\mathbf{y} = \mathbf{k}$. cost: $n-1$ mult. $n-1$ add.

$$\circ \begin{cases} y_1 = k_1 \\ \text{For } i = 2, \dots, n \\ y_i = k_i - \ell_{i-1} y_{i-1} \end{cases}$$

Back substitution $U\mathbf{x} = \mathbf{y}$. cost: $2(n-1)+1$ mult. $n-1$ add.

$$\circ \begin{cases} x_n = y_n/d_n \\ \text{For } i = n-1, \dots, 1 \\ x_i = (y_i - c_i y_{i+1})/d_i \end{cases}$$

It requires $(5n-4)$ multiplication and $3n-3$ addition.

Theorem 4.4.2. (Holliday) *Among all C^2-function which interpolate f at $\{x_i\}_{i=0}^n$, the natural cubic spline has smallest curvature (minimal energy-smooth),* *i.e., if $g(x)$ is any C^2-interpolant, then*

$$\int_a^b [s''(t)]^2\,dt \le \int_a^b [g''(t)]^2.$$

Proof. We see, for any C^2-interpolant $g(x)$ that

$$0 \le \int_a^b [g''(t)-s''(t)]^2\,dt = \int_a^b [g''(t)]^2 - 2\int_a^b [g''(t)-s''(t)]s''(t) - \int_a^b [s''(t)]^2\,dt.$$

We will show the second term is zero, which completes the proof,

$$\begin{aligned}\int_a^b [g''-s'']s'' &= \sum_{i=0}^{n-1}\int_{x_i}^{x_{i+1}} [g''-s'']s'' = \sum_{i=0}^{n-1}[(g'-s')s''\Big|_{x_i}^{x_{i+1}} - \int_{x_i}^{x_{i+1}}(g'-s')s'''] \\ &= \sum_{i=0}^{n-1}[(g'(x_{i+1})-s'(x_{i+1}))s''(x_{i+1}) - (g(x)-s(x))s'''(x)]\Big|_{x_i}^{x_{i+1}}.\end{aligned}$$

By telescoping series, the first term equals $(g'(x_n)-s'(x_n))s''(x_n) - (g'(x_0) - s'(x_0))s''(x_0) = 0$ by the assumption $s''(x_0) = s''(x_n) = 0$. Since s''' is constant on each interval, the second one is also zero by the assumption $g(x_i) = s(x_i)$.

□

Remark 4.4.3. There are other choices of σ_0 and σ_n. Suppose the values $f'(x_0)$, $f'(x_n)$ are known. Then we obtain so called a **clamped spline.**

Exercise 4.4.4. Put $s'(x_0) = f'(x_0)$, $s'(x_n) = f'(x_n)$ (clamped)

Theorem 4.4.5. *Let $f \in C^2[a,b]$ and s be the natural cubic spline with node $\{x_i\}_{i=0}^n$, then with $h = \max_k h_k$*

$$\begin{aligned}\|f-s\|_\infty &\le h^{3/2}\left(\int_a^b |f''|^2\right)^{1/2} \\ &\textit{and} \\ \|f'-s'\|_\infty &\le h^{1/2}\left(\int_a^b |f''|^2\right)^{1/2}.\end{aligned}$$

Thus cubic spline is also good to approximate f'.

Proof. We prove the second estimate first. Fix $x \in [x_{k-1}, x_k] = I_k$. By Rolle's theorem, there exists $\tau \in I_k$ such that $f'(\tau) - s'(\tau) = 0$. Then we have

$$\int_\tau^x [f''(t) - s''(t)]\,dt = f'(t) - s'(t)|_\tau^x = f'(x) - s'(x)$$

and by the proof of Holliday's Theorem,

$$\begin{aligned}|f'(x)-s'(x)| &= |\int_\tau^x (f''-s'')\,dt| \le \left(\int_\tau^x |f''-s''|^2 dt\right)^{1/2} (\int_\tau^x dt)^{1/2}\\ &\le h^{1/2}\left(\int_a^b |f''-s''|^2 dt\right)^{1/2} = h^{1/2}\left(\int_a^b (|f''|^2-|s''|^2) dt\right)^{1/2}\\ &\le h^{1/2}\left(\int_a^b |f''|^2 dt\right)^{1/2}.\end{aligned}$$

Now for the first estimate we see, for $x \in [x_i, x_{i+1}]$

$$\begin{aligned}|f(x)-s(x)| &= \left|\int_{x_i}^x (f'(t)-s'(t))\,dt\right| \le \int_{x_i}^x |f'(t)-s'(t)|\,dt\\ &\le h\|f'-s'\|_\infty = h^{3/2}\left(\int_a^b |f''|^2\right)^{1/2}.\end{aligned}$$

□

Example 4.4.6. Consider the data $\{(-1,y_0),(0,y_1),(1,y_2)\}$. Find the natural spline to fit these data. Since $n=2$ and $\sigma_0=\sigma_2=0$, we have from (4.14),(4.15)

$$\omega_1\sigma_0 + 2\sigma_1 + (1-\omega_1)\sigma_2 = r_0 = \frac{3}{1}\left(\frac{y_2-y_1}{1} - \frac{y_1-y_0}{1}\right) \tag{4.14}$$

$$\sigma_1 = \frac{3}{2}(y_2 - 2y_1 + y_0).$$

From (4.9), (4.10)

$$p_k(x) = \sigma_k \frac{(x_{k+1}-x)^3}{6h_k} + \sigma_{k+1}\frac{(x-x_k)^3}{6h_k} + \tau_k(x-x_k) + \kappa_k.$$

$\kappa_k = y_k - \frac{\sigma_k h_k^2}{6}$, $\tau_k = \frac{y_{k+1}-y_k}{h_k} - \frac{h_k}{6}(\sigma_{k+1}-\sigma_k)$. Hence

$$p_0(x) = \sigma_1\frac{(x+1)^3}{6} + (y_1 - y_0 - \frac{\sigma_1}{6})(x+1) + y_0.$$

$$p_1(x) = \sigma_1\frac{(1-x)^3}{6} + (y_2 - y_1 + \frac{\sigma_1}{6})x + y_1 - \frac{\sigma_1}{6}.$$

Thus if $y_0=-1, y_1=1, y_2=1$, $\sigma_1=-3$, we obtain the same solution as the book.

Not-a-knot Boundary Condition

We take $p_0(x) \equiv p_1(x)$ and $p_{n-2}(x) \equiv p_{n-1}(x)$. Then the nodes x_1 and x_{n-2} are **not** considered as a knot. Since $p_0^{(j)}(x_1) = p_1^{(j)}(x_1)$, $j = 0, 1, 2$, it suffices to impose the same condition for $j = 3$. We impose the same condition at x_{n-1}. Hence the two extra equations are

$$
\begin{aligned}
\frac{-\sigma_1 + \sigma_2}{6h_1} &= \frac{-\sigma_0 + \sigma_1}{6h_0}, \\
\frac{-\sigma_{n-2} + \sigma_{n-1}}{6h_{n-2}} &= \frac{-\sigma_{n-1} + \sigma_n}{6h_{n-1}}.
\end{aligned}
$$

Clamped Spline

If s_0, s_n(derivatives at end points) are available, one can also construct a spline satisfying the derivative condition: Applying (4.12), (4.13) for $k = 0$ and $k = n$ resp., we obtain

$$
s_0 = \frac{y_1 - y_0}{h_0} - \frac{\sigma_0 h_0}{3} - \frac{\sigma_1 h_0}{6}
$$

and

$$
s_n = \frac{y_{n-1} - y_n}{h_{n-1}} + \frac{\sigma_n h_{n-1}}{3} + \frac{\sigma_{n-1} h_{n-1}}{6}.
$$

Appending these equations to the previous equations, we get

$$
\begin{bmatrix}
2 & 1 & 0 & 0 & & 0 \\
\omega_1 & 2 & 1-\omega_1 & 0 & \dots & 0 \\
0 & \omega_2 & 2 & 1-\omega_2 & \dots & 0 \\
\vdots & \vdots & \ddots & \ddots & \ddots & 0 \\
0 & \dots & \dots & \omega_{n-1} & 2 & 1-\omega_{n-1} \\
0 & 0 & \dots & 0 & 1 & 2
\end{bmatrix}
\begin{bmatrix}
\sigma_0 \\ \sigma_1 \\ \vdots \\ \sigma_{n-1} \\ \sigma_n
\end{bmatrix}
=
\begin{bmatrix}
r_0 \\ r_1 \\ \vdots \\ r_{n-1} \\ r_n
\end{bmatrix}
$$

where r_0 and r_n are appropriate functions of y_0, y_1, h_0, s_0 and $y_{n-1}, y_n, h_{n-1}, s_n$, etc.

Homework

Construct a spline approximation to $f(x) = \frac{1}{1+x^2}$ on $[-5, 5]$ with $n = 10, 20, 30, \cdots$ with equally spaced subintervals. Estimate $\|s(x) - f(x)\|_\infty$ and $\|s'(x) - f'(x)\|_\infty$ (compute these norms by choosing sufficiently many points in each sub-interval). Compare with theoretical bound. Also compare with earlier results with p_{2n} using Chebysheff points. Draw graphs.

Do the same for $f(x) = e^{0.8x}$ on $[-3, 3]$.