Numerical Analysis I
Interpolation and Polynomial Approximation

Instructor: Wei-Cheng Wang [1]

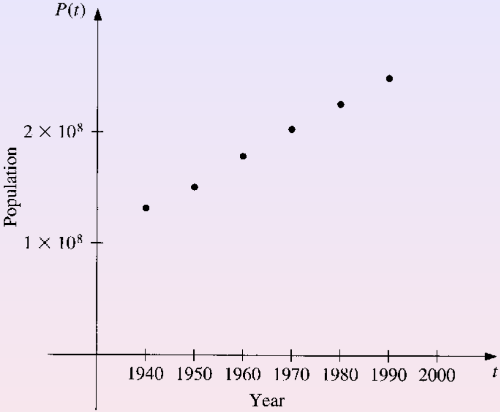
Department of Mathematics
National TsingHua University

Fall 2017

[1] These slides are based on Prof. Tsung-Ming Huang(NTNU)'s original slides

Outline

Introduction



Question

From these data, how do we get a reasonable estimate of the population, say, in 1965, or even in 2010?

Interpolation

Suppose we do not know the function f, but a few information (data) about f, now we try to compute a function g that approximates f.

Theorem (Weierstrass Approximation Theorem)

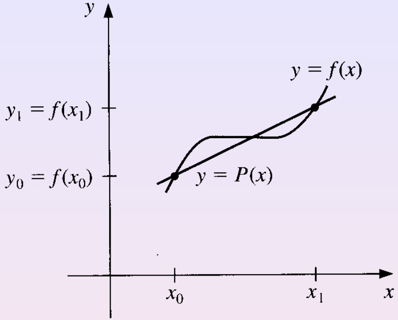
Suppose that f is defined and continuous on $[a, b]$. For any $\varepsilon > 0$, there exists a polynomial $P(x)$, such that

$$|f(x) - P(x)| < \varepsilon, \qquad \text{for all } x \text{ in } [a, b].$$

Reason for using polynomial

1. They uniformly approximate continuous functions (Weierstrass Theorem)
2. The derivatives and indefinite integral of a polynomial are easy to determine and are also polynomials.

Interpolation and the Lagrange polynomial



Property

The linear function passing through $(x_0, f(x_0))$ and $(x_1, f(x_1))$ is unique.

Let

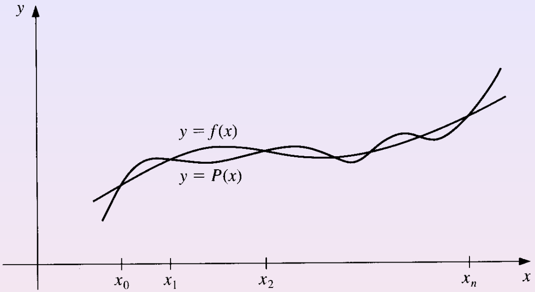
$$L_0(x) = \frac{x - x_1}{x_0 - x_1}, \quad L_1(x) = \frac{x - x_0}{x_1 - x_0}$$

and

$$P(x) = L_0(x)f(x_0) + L_1(x)f(x_1).$$

Then

$$P(x_0) = f(x_0), \quad P(x_1) = f(x_1).$$



Question

How to find the polynomial of degree n that passes through $(x_0, f(x_0)), \ldots, (x_n, f(x_n))$?

Theorem

If (x_i, y_i), $x_i, y_i \in \mathbb{R}$, $i = 0, 1, \ldots, n$, are $n+1$ distinct pairs of data point, then there is a unique polynomial P_n of degree at most n such that

$$P_n(x_i) = y_i, \quad (0 \le i \le n). \tag{1}$$

Proof of Existence (by mathematical induction)

1. The theorem clearly holds for $n = 0$ (only one data point (x_0, y_0)) since one may choose the constant polynomial $P_0(x) = y_0$ for all x.
2. Assume that the theorem holds for $n \leq k$, that is, there is a polynomial P_k, $\deg(P_k) \leq k$, such that $y_i = P_k(x_i)$, for $0 \leq i \leq k$.
3. Next we try to construct a polynomial of degree at most $k+1$ to interpolate (x_i, y_i), $0 \leq i \leq k+1$. Let

$$P_{k+1}(x) = P_k(x) + c(x - x_0)(x - x_1)\cdots(x - x_k),$$

where

$$c = \frac{y_{k+1} - P_k(x_{k+1})}{(x_{k+1} - x_0)(x_{k+1} - x_1)\cdots(x_{k+1} - x_k)}.$$

Since x_i are distinct, the polynomial $P_{k+1}(x)$ is well-defined and $\deg(P_{k+1}) \leq k+1$. It is easy to verify that

$$P_{k+1}(x_i) = y_i, \quad 0 \leq i \leq k+1.$$

Proof of Uniqueness

Suppose there are two such polynomials P_n and Q_n satisfying (1). Define

$$S_n(x) = P_n(x) - Q_n(x).$$

Since both deg$(P_n) \leq n$ and deg$(Q_n) \leq n$, deg$(S_n) \leq n$. Moreover

$$S_n(x_i) = P_n(x_i) - Q_n(x_i) = y_i - y_i = 0,$$

for $0 \leq i \leq n$. This means that S_n has at least $n+1$ zeros, it therefore must be $S_n = 0$. Hence $P_n = Q_n$. □

Idea

Construct polynomial $P(x)$ with $\deg(P) \leq n$ as

$$P(x) = f(x_0)L_{n,0}(x) + \cdots + f(x_n)L_{n,n}(x),$$

where

1. $L_{n,k}(x)$ are polynomial with degree n for $0 \leq k \leq n$.
2. $L_{n,k}(x_k) = 1$ and $L_{n,k}(x_i) = 0$ for $i \neq k$.

Then

$$P(x_k) = f(x_k) \quad \text{for } k = 0, 1, \ldots, n.$$

1. $\deg(L_{n,k}) = n$ and $L_{n,k}(x_i) = 0$ for $i \neq k$:

   $$L_{n,k}(x) = c_k(x - x_0)(x - x_1)\cdots(x - x_{k-1})(x - x_{k+1})\cdots(x - x_n)$$

2. $L_{n,k}(x_k) = 1$:

   $$L_{n,k}(x) = \frac{(x - x_0)(x - x_1)\cdots(x - x_{k-1})(x - x_{k+1})\cdots(x - x_n)}{(x_k - x_0)(x_k - x_1)\cdots(x_k - x_{k-1})(x_k - x_{k+1})\cdots(x_k - x_n)}$$

Theorem

If $x_0, x_1, \ldots, x_n$ are $n+1$ distinct numbers and f is a function whose values are given at these numbers, then a unique polynomial of degree at most n exists with

$$f(x_k) = P(x_k), \quad \text{for } k = 0, 1, \ldots, n.$$

This polynomial is given by

$$P(x) = \sum_{k=0}^{n} f(x_k) L_{n,k}(x)$$

which is called the nth Lagrange interpolating polynomial.

Note that we will write $L_{n,k}(x)$ simply as $L_k(x)$ when there is no confusion as to its degree.

Example

Given the following 4 data points,

x_i	0	1	3	5
y_i	1	2	6	7

find a polynomial in Lagrange form to interpolate these data.

Solution: The interpolating polynomial in the Lagrange form is

$$P_3(x) = L_0(x) + 2L_1(x) + 6L_2(x) + 7L_3(x) \quad \text{with}$$

$$\begin{aligned}
L_0(x) &= \frac{(x-1)(x-3)(x-5)}{(0-1)(0-3)(0-5)} = -\frac{1}{15}(x-1)(x-3)(x-5), \\
L_1(x) &= \frac{(x-0)(x-3)(x-5)}{(1-0)(1-3)(1-5)} = \frac{1}{8}x(x-3)(x-5), \\
L_2(x) &= \frac{(x-0)(x-1)(x-5)}{(3-0)(3-1)(3-5)} = -\frac{1}{12}x(x-1)(x-5), \\
L_3(x) &= \frac{(x-0)(x-1)(x-3)}{(5-0)(5-1)(5-3)} = \frac{1}{40}x(x-1)(x-3).
\end{aligned}$$

Question

What's the error involved in approximating $f(x)$ by the interpolating polynomial $P(x)$?

Theorem

Suppose

1. $x_0, \ldots, x_n$ *are distinct numbers in* $[a,b]$,
2. $f \in C^{n+1}[a,b]$.

Then, $\forall x \in [a,b]$, $\exists\ \xi(x) \in (a,b)$ *such that*

$$f(x) = P(x) + \frac{f^{n+1}(\xi(x))}{(n+1)!}(x-x_0)(x-x_1)\cdots(x-x_n). \tag{2}$$

Proof: If $x = x_k$, for any $k = 0, 1, \ldots, n$, then $f(x_k) = P(x_k)$ and (2) is satisfied.

If $x \neq x_k$, for all $k = 0, 1, \ldots, n$, define

$$g(t) = f(t) - P(t) - [f(x) - P(x)] \prod_{i=0}^{n} \frac{t - x_i}{x - x_i}.$$

Since $f \in C^{n+1}[a, b]$ and $P \in C^{\infty}[a, b]$, it follows that $g \in C^{n+1}[a, b]$.
Since

$$g(x_k) = [f(x_k) - P(x_k)] - [f(x) - P(x)] \prod_{i=0}^{n} \frac{x_k - x_i}{x - x_i} = 0,$$

and

$$g(x) = [f(x) - P(x)] - [f(x) - P(x)] \prod_{i=0}^{n} \frac{x - x_i}{x - x_i} = 0,$$

it implies that g is zero at $x, x_0, x_1, \ldots, x_n$. By the Generalized Rolle's Theorem, $\exists\ \xi \in (a, b)$ such that $g^{n+1}(\xi) = 0$. That is

$$
\begin{aligned}
0 &= g^{(n+1)}(\xi) \qquad (3)\\
&= f^{(n+1)}(\xi) - P^{(n+1)}(\xi) - [f(x) - P(x)] \frac{d^{n+1}}{dt^{n+1}} \left[\prod_{i=0}^{n} \frac{(t - x_i)}{(x - x_i)} \right]_{t=\xi}.
\end{aligned}
$$

Since $\deg(P) \leq n$, it implies that $P^{(n+1)}(\xi) = 0$. On the other hand, $\prod_{i=0}^{n}[(t - x_i)/(x - x_i)]$ is a polynomial of degree $(n+1)$, so

$$
\prod_{i=0}^{n} \frac{(t - x_i)}{(x - x_i)} = \left[\frac{1}{\prod_{i=0}^{n}(x - x_i)} \right] t^{n+1} + (\text{lower-degree terms in } t),
$$

and

$$
\frac{d^{n+1}}{dt^{n+1}} \left[\prod_{i=0}^{n} \frac{(t - x_i)}{(x - x_i)} \right] = \frac{(n+1)!}{\prod_{i=0}^{n}(x - x_i)}.
$$

Equation (3) becomes

$$0 = f^{(n+1)}(\xi) - [f(x) - P(x)] \frac{(n+1)!}{\prod_{i=0}^{n}(x - x_i)},$$

i.e.,

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

$\square$

Example

1. Goal: Prepare a table for the function $f(x) = e^x$ for $x \in [0, 1]$.
2. $x_{j+1} - x_j = h$ for $j = 0, 1, \ldots, n-1$.
3. What should h be for linear interpolation to give an absolute error of at most 10^{-6}?

Suppose $x \in [x_j, x_{j+1}]$. Equation (2) implies that

$$
\begin{aligned}
|f(x) - P(x)| &= \left| \frac{f^{(2)}(\xi)}{2!}(x - x_j)(x - x_{j+1}) \right| \\
&= \frac{|f^{(2)}(\xi)|}{2!}|x - x_j||x - x_{j+1}| \\
&= \frac{e^{\xi}}{2}|x - jh||x - (j+1)h| \\
&\leq \frac{1}{2}\left(\max_{\xi \in [0,1]} e^{\xi}\right)\left(\max_{x_j \leq x \leq x_{j+1}} |x - jh||x - (j+1)h|\right) \\
&= \frac{e}{2}\left(\max_{x_j \leq x \leq x_{j+1}} |x - jh||x - (j+1)h|\right).
\end{aligned}
$$

Let

$$g(x) = (x - jh)(x - (j+1)h), \quad \text{for } jh \le x \le (j+1)h.$$

Then

$$\max_{x_j \le x \le x_{j+1}} |g(x)| = \left| g\left(\left(j + \frac{1}{2}\right) h\right) \right| = \frac{h^2}{4}.$$

Consequently,

$$|f(x) - P(x)| \le \frac{eh^2}{8} \le 10^{-6},$$

which implies that

$$h < 1.72 \times 10^{-3}.$$

Since $n = (1-0)/h$ must be an integer, one logical choice for the step size is $h = 0.001$.

Difficulty for the Lagrange interpolation

1. If more data points are added to the interpolation problem, all the cardinal functions L_k have to be recalculated.
2. We shall now derive the interpolating polynomials in a manner that uses the previous calculations to greater advantage.

Definition

1. f is a function defined at $x_0, x_1, \ldots, x_n$
2. Suppose that $m_1, m_2, \ldots, m_k$ are k distinct integers with $0 \leq m_i \leq n$ for each i.

The Lagrange polynomial that interpolates f at the k points $x_{m_1}, x_{m_2}, \ldots, x_{m_k}$ is denoted $P_{m_1,m_2,\ldots,m_k}(x)$.

Theorem

Let f be defined at distinct points $x_0, x_1, \ldots, x_k$, and $0 \le i, j \le k$, $i \ne j$. Then

$$P(x) = \frac{(x - x_j)}{(x_i - x_j)} P_{0,1,\ldots,j-1,j+1,\ldots,k}(x) - \frac{(x - x_i)}{(x_i - x_j)} P_{0,1,\ldots,i-1,i+1,\ldots,k}(x)$$

describes the k-th Lagrange polynomial that interpolates f at the $k+1$ points $x_0, x_1, \ldots, x_k$.

Proof: Since

$$\deg(P_{0,1,\ldots,j-1,j+1,\ldots,k}) \le k - 1$$

and

$$\deg(P_{0,1,\ldots,i-1,i+1,\ldots,k}) \le k - 1,$$

it implies that $\deg(P) \le k$. If $0 \le r \le k$ and $r \ne i, j$, then

$$
\begin{aligned}
P(x_r) &= \frac{(x_r - x_j)}{(x_i - x_j)} P_{0,1,\ldots,j-1,j+1,\ldots,k}(x_r) - \frac{(x_r - x_i)}{(x_i - x_j)} P_{0,1,\ldots,i-1,i+1,\ldots,k}(x_r) \\
&= \frac{(x_r - x_j)}{(x_i - x_j)} f(x_r) - \frac{(x_r - x_i)}{(x_i - x_j)} f(x_r) = f(x_r).
\end{aligned}
$$

Moreover

$$
\begin{aligned}
P(x_i) &= \frac{(x_i - x_j)}{(x_i - x_j)} P_{0,1,\ldots,j-1,j+1,\ldots,k}(x_i) - \frac{(x_i - x_i)}{(x_i - x_j)} P_{0,1,\ldots,i-1,i+1,\ldots,k}(x_i) \\
&= f(x_i)
\end{aligned}
$$

and

$$
\begin{aligned}
P(x_j) &= \frac{(x_j - x_j)}{(x_i - x_j)} P_{0,1,\ldots,j-1,j+1,\ldots,k}(x_j) - \frac{(x_j - x_i)}{(x_i - x_j)} P_{0,1,\ldots,i-1,i+1,\ldots,k}(x_j) \\
&= f(x_j).
\end{aligned}
$$

Therefore $P(x)$ agrees with f at all points $x_0, x_1, \ldots, x_k$. By the uniqueness theorem, $P(x)$ is the k-th Lagrange polynomial that interpolates f at the $k+1$ points $x_0, x_1, \ldots, x_k$, i.e., $P \equiv P_{0,1,\ldots,k}$.

Neville's method

The theorem implies that the Lagrange interpolating polynomial can be generated recursively. The procedure is called the Neville's method.

1. Denote
$$Q_{i,j} = P_{i-j,i-j+1,\ldots,i-1,i}.$$
2. Hence $Q_{i,j}$, $0 \leq j \leq i$, denotes the interpolating polynomial of degree j on the $j+1$ points $x_{i-j}, x_{i-j+1}, \ldots, x_{i-1}, x_i$.
3. The polynomials can be computed in a manner as shown in the following table.

x_0	$P_0 = Q_{0,0}$				
x_1	$P_1 = Q_{1,0}$	$P_{0,1} = Q_{1,1}$			
x_2	$P_2 = Q_{2,0}$	$P_{1,2} = Q_{2,1}$	$P_{0,1,2} = Q_{2,2}$		
x_3	$P_3 = Q_{3,0}$	$P_{2,3} = Q_{3,1}$	$P_{1,2,3} = Q_{3,2}$	$P_{0,1,2,3} = Q_{3,3}$	
x_4	$P_4 = Q_{4,0}$	$P_{3,4} = Q_{4,1}$	$P_{2,3,4} = Q_{4,2}$	$P_{1,2,3,4} = Q_{4,3}$	$P_{0,1,2,3,4} = Q_{4,4}$

x_0	$P_0 = Q_{0,0}$				
x_1	$P_1 = Q_{1,0}$	$P_{0,1} = Q_{1,1}$			
x_2	$P_2 = Q_{2,0}$	$P_{1,2} = Q_{2,1}$	$P_{0,1,2} = Q_{2,2}$		
x_3	$P_3 = Q_{3,0}$	$P_{2,3} = Q_{3,1}$	$P_{1,2,3} = Q_{3,2}$	$P_{0,1,2,3} = Q_{3,3}$	
x_4	$P_4 = Q_{4,0}$	$P_{3,4} = Q_{4,1}$	$P_{2,3,4} = Q_{4,2}$	$P_{1,2,3,4} = Q_{4,3}$	$P_{0,1,2,3,4} = Q_{4,4}$

$$
\begin{aligned}
P_{0,1}(x) &= \frac{(x-x_1)P_0(x) - (x-x_0)P_1(x)}{x_0 - x_1}, \\
P_{1,2}(x) &= \frac{(x-x_2)P_1(x) - (x-x_1)P_2(x)}{x_1 - x_2}, \\
&\vdots
\end{aligned}
$$

$$
\begin{aligned}
P_{0,1,2}(x) &= \frac{(x-x_2)P_{0,1}(x) - (x-x_0)P_{1,2}(x)}{x_0 - x_2} \\
&\vdots
\end{aligned}
$$

Example

Compute approximate value of $f(1.5)$ by using the following data:

x	$f(x)$
1.0	0.7651977
1.3	0.6200860
1.6	0.4554022
1.9	0.2818186
2.2	0.1103623

1. The first-degree approximation:

$$
\begin{aligned}
Q_{1,1}(1.5) &= \frac{(x-x_0)Q_{1,0}-(x-x_1)Q_{0,0}}{x_1-x_0} \\
&= \frac{(1.5-1.0)Q_{1,0}-(1.5-1.3)Q_{0,0}}{1.3-1.0} = 0.5233449, \\
Q_{2,1}(1.5) &= \frac{(x-x_1)Q_{2,0}-(x-x_2)Q_{1,0}}{x_2-x_1} \\
&= \frac{(1.5-1.3)Q_{2,0}-(1.5-1.6)Q_{1,0}}{1.6-1.3} = 0.5102968, \\
Q_{3,1}(1.5) &= \frac{(x-x_2)Q_{3,0}-(x-x_3)Q_{2,0}}{x_3-x_2} \\
&= \frac{(1.5-1.6)Q_{3,0}-(1.5-1.9)Q_{2,0}}{1.9-1.6} = 0.5132634, \\
Q_{4,1}(1.5) &= \frac{(x-x_3)Q_{4,0}-(x-x_4)Q_{3,0}}{x_4-x_3} \\
&= \frac{(1.5-1.9)Q_{4,0}-(1.5-2.2)Q_{3,0}}{2.2-1.9} = 0.5104270.
\end{aligned}
$$

1. The second-degree approximation:

$$
\begin{aligned}
Q_{2,2}(1.5) &= \frac{(x-x_1)Q_{2,1}-(x-x_2)Q_{1,1}}{x_2-x_1} \\
&= \frac{(1.5-1.3)Q_{2,1}-(1.5-1.6)Q_{1,1}}{1.6-1.3} = 0.5124715, \\
Q_{3,2}(1.5) &= \frac{(x-x_2)Q_{3,1}-(x-x_3)Q_{2,1}}{x_3-x_2} \\
&= \frac{(1.5-1.6)Q_{3,1}-(1.5-1.9)Q_{2,1}}{1.9-1.6} = 0.5112857, \\
Q_{4,2}(1.5) &= \frac{(x-x_3)Q_{4,1}-(x-x_4)Q_{3,1}}{x_4-x_3} \\
&= \frac{(1.5-1.9)Q_{4,1}-(1.5-2.2)Q_{3,1}}{2.2-1.9} = 0.5137361.
\end{aligned}
$$

x	$f(x)$	1st-deg	2nd-deg	3rd-deg	4th-deg
1.0	0.7651977				
1.3	0.6200860	0.5233449			
1.6	0.4554022	0.5102968	0.5124715		
1.9	0.2818186	0.5132634	0.5112857	0.5118127	
2.2	0.1103623	0.5104270	0.5137361	0.5118302	0.5118200

Table: Results of the higher-degree approximations

x	$f(x)$	1st-deg	2nd-deg	3rd-deg	4th-deg	5th-deg
1.0	0.7651977					
1.3	0.6200860	0.5233449				
1.6	0.4554022	0.5102968	0.5124715			
1.9	0.2818186	0.5132634	0.5112857	0.5118127		
2.2	0.1103623	0.5104270	0.5137361	0.5118302	0.5118200	
2.5	−0.0483838	0.4807699	0.5301984	0.5119070	0.5118430	0.5118277

Table: Results of adding $(x_5, f(x_5))$

Divided Differences

- Neville's method: generate successively higher-degree polynomial approximations at a specific point.
- Divided-difference methods: successively generate the polynomials themselves.

Suppose that $P_n(x)$ is the nth Lagrange polynomial that agrees with f at distinct $x_0, x_1, \ldots, x_n$. Express $P_n(x)$ in the form

$$\begin{aligned} P_n(x) &= a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \cdots \\ &\quad + a_n(x - x_0)(x - x_1)\cdots(x - x_{n-1}). \end{aligned}$$

- Determine constant a_0:

$$a_0 = P_n(x_0) = f(x_0)$$

- Determine a_1:

$$f(x_0) + a_1(x_1 - x_0) = P_n(x_1) = f(x_1) \quad \Rightarrow \quad a_1 = \frac{f(x_1) - f(x_0)}{x_1 - x_0}.$$

- The zero divided difference of the function f with respect to x_i:

$$f[x_i] = f(x_i)$$

- The first divided difference of f with respect to x_i and x_{i+1} is denoted and defined as

$$f[x_i, x_{i+1}] = \frac{f[x_{i+1}] - f[x_i]}{x_{i+1} - x_i}.$$

- The second divided difference of f is defined as

$$f[x_i, x_{i+1}, x_{i+2}] = \frac{f[x_{i+1}, x_{i+2}] - f[x_i, x_{i+1}]}{x_{i+2} - x_i}.$$

- The k-th divided difference relative to $x_i, x_{i+1}, \ldots, x_{i+k}$ is given by

$$f[x_i, x_{i+1}, \ldots, x_{i+k}] = \frac{f[x_{i+1}, x_{i+2}, \ldots, x_{i+k}] - f[x_i, x_{i+1}, \ldots, x_{i+k-1}]}{x_{i+k} - x_i}$$

As might be expected from the evaluation of $a_0, a_1, \ldots, a_n$ in $P_n(x)$, the required coefficients are given by

$$a_k = f[x_0, x_1, \ldots, x_k], \qquad k = 0, 1, \ldots, n.$$

Therefore $P_n(x)$ can be expressed as

$$P_n(x) = f[x_0] + \sum_{k=1}^{n} f[x_0, x_1, \ldots, x_k](x - x_0)(x - x_1)\cdots(x - x_{k-1}).$$

This formula is known as the Newton's divided-difference formula.

$$\begin{array}{c|lclclcl}
x_i & k=0 & & k=1 & & k=2 & & k=3 \\
\hline
x_0 & f[x_0] & & & & & & \\
 & & > & f[x_0,x_1] & & & & \\
x_1 & f[x_1] & & & > & f[x_0,x_1,x_2] & & \\
 & & > & f[x_1,x_2] & & & > & f[x_0,x_1,x_2,x_3] \\
x_2 & f[x_2] & & & > & f[x_1,x_2,x_3] & & \\
 & & > & f[x_2,x_3] & & & & \\
x_3 & f[x_3] & & & & & &
\end{array}$$

Table: Dependency diagram of Newton's divided differences.

Step 0 INPUT $(x_0, y_0), \cdots, (x_n, y_n)$. Set $F_{0,0} = y_0, \cdots, F_{n,0} = y_n$.

Step 1. For $i = 1, 2, \cdots, n$

For $k = 1, 2, \cdots, i$

$$F_{i,k} = \frac{F_{i,k-1} - F_{i-1,k-1}}{x_i - x_{i-k}} \quad (F_{i,k} = f[x_{i-k}, \cdots, x_k])$$

End k

End i

Step 2. OUTPUT $F_{0,0}, F_{1,1}, \cdots, F_{n,n}$ $(F_{i,i} = f[x_0, \cdots, x_i])$, STOP.

x_i	$k=0$	$k=1$	$k=2$	$k=3$
x_0	$f[x_0]$			
		> $f[x_0, x_1]$		
x_1	$f[x_1]$		> $f[x_0, x_1, x_2]$	
		> $f[x_1, x_2]$		> $f[x_0, x_1, x_2, x_3]$
x_2	$f[x_2]$		> $f[x_1, x_2, x_3]$	
		> $f[x_2, x_3]$		
x_3	$f[x_3]$			

Table: Divided differences generated from left to right, then from top to bottom. Red: $i = 0$; green: $i = 1$; blue: $i = 2$; black: $i = 3$.

Newton's Divided Difference (Storage Saving Version)

Step 0 INPUT $(x_0, y_0), \cdots, (x_n, y_n)$. Set $F_0 = y_0, \cdots, F_n = y_n$.

Step 1. For $k = 1, 2, \cdots, n$

For $i = n, n-1, \cdots, k$

$$F_i \leftarrow \frac{F_i - F_{i-1}}{x_i - x_{i-k}} \quad \left(F_{i,k} = \frac{F_{i,k-1} - F_{i-1,k-1}}{x_i - x_{i-k}} \right)$$

End i

End k

Step 2. OUTPUT $F_0, F_1, \cdots, F_n$ $(F_i = f[x_0, \cdots, x_i])$, STOP.

x_i	$k=0$		$k=1$		$k=2$		$k=3$
x_0	$f[x_0]$						
		$>$	$f[x_0, x_1]$				
x_1	$f[x_1]$			$>$	$f[x_0, x_1, x_2]$		
		$>$	$f[x_1, x_2]$			$>$	$f[x_0, x_1, x_2, x_3]$
x_2	$f[x_2]$			$>$	$f[x_1, x_2, x_3]$		
		$>$	$f[x_2, x_3]$				
x_3	$f[x_3]$						

Table: Divided differences generated from bottom to top, then from left to right.

Example

Given the following 4 points ($n = 3$)

x_i	0	1	3	5
y_i	1	2	6	7

find a polynomial of degree 3 in Newton's form to interpolate these data.

Solution:

$$
\begin{array}{c|cccc}
x_i & k=0 & k=1 & k=2 & k=3 \\
\hline
0 & 1 & & & \\
 & & > 1 & & \\
1 & 2 & & > \frac{1}{3} & \\
 & & > 2 & & > -\frac{17}{120} \\
3 & 6 & & > -\frac{3}{8} & \\
 & & > \frac{1}{2} & & \\
5 & 7 & & &
\end{array}
$$

Therefore, $P(x) = 1 + x + \dfrac{1}{3}x(x-1) - \dfrac{17}{120}x(x-1)(x-3)$.

Theorem (11)

Suppose $f \in C^n[a,b]$ *and* $x_0, x_1, \ldots, x_n$ *are distinct numbers in* $[a,b]$. *Then there exists* $\xi \in (a,b)$ *such that*

$$f[x_0, x_1, \ldots, x_n] = \frac{f^{(n)}(\xi)}{n!}.$$

Proof: Define

$$g(x) = f(x) - P_n(x).$$

Since $P_n(x_i) = f(x_i)$ for $i = 0, 1, \ldots, n$, g has $n+1$ distinct zeros in $[a,b]$. By the generalized Rolle's Theorem, $\exists\ \xi \in (a,b)$ such that

$$0 = g^{(n)}(\xi) = f^{(n)}(\xi) - P_n^{(n)}(\xi).$$

Note that

$$P_n^{(n)}(x) = n! f[x_0, x_1, \ldots, x_n].$$

As a consequence

$$f[x_0, x_1, \ldots, x_n] = \frac{f^{(n)}(\xi)}{n!}.$$

Let

$$h = \frac{x_n - x_0}{n} = x_{i+1} - x_i, \qquad i = 0, 1, \ldots, n-1.$$

Then each $x_i = x_0 + ih$, $i = 0, 1, \ldots, n$. For any $x \in [a, b]$, write

$$x = x_0 + sh, \qquad s \in \mathbb{R}.$$

Hence $x - x_i = (s - i)h$ and

$$\begin{aligned} P_n(x) &= f[x_0] + \sum_{k=1}^{n} f[x_0, x_1, \ldots, x_k](x - x_0)(x - x_1) \cdots (x - x_{k-1}) \\ &= f(x_0) + \sum_{k=1}^{n} f[x_0, x_1, \ldots, x_k](s-0)h(s-1)h \cdots (s-k+1)h \\ &= f(x_0) + \sum_{k=1}^{n} f[x_0, x_1, \ldots, x_k]s(s-1) \cdots (s-k+1)h^k \\ &= f(x_0) + \sum_{k=1}^{n} f[x_0, x_1, \ldots, x_k]k!\binom{s}{k}h^k, \end{aligned} \tag{4}$$

- The binomial formula

$$\binom{s}{k} = \frac{s(s-1)\cdots(s-k+1)}{k!}$$

- The forward difference notation $\triangle$

$$\triangle f(x_i) = f(x_{i+1}) - f(x_i)$$

and for $i = 0, 1, \ldots, n-1$,

$$\triangle^k f(x_i) = \triangle^{k-1} f(x_{i+1}) - \triangle^{k-1} f(x_i) = \triangle\left(\triangle^{k-1} f(x_i)\right).$$

- With this notation,

$$\begin{aligned} f[x_0, x_1] &= \frac{f(x_1) - f(x_0)}{x_1 - x_0} = \frac{1}{h}\triangle f(x_0) \\ f[x_0, x_1, x_2] &= \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} \\ &= \frac{\frac{1}{h}\triangle f(x_1) - \frac{1}{h}\triangle f(x_0)}{2h} = \frac{1}{2h^2}\triangle^2 f(x_0), \end{aligned}$$

- In general

$$f[x_0, x_1, \ldots, x_k] = \frac{1}{k!\, h^k} \triangle^k f(x_0).$$

- The Newton forward-difference formula:

$$P_n(x) = f(x_0) + \sum_{k=1}^{n} \binom{s}{k} \triangle^k f(x_0).$$

- If x is close to x_0, the first few terms in the above summation also gives good lower degree interpolating polynomials.

- In case x is close to x_n, we could rearrange interpolation nodes as $x_n, x_{n-1}, \ldots, x_0$:

$$\begin{aligned} P_n(x) &= f[x_n] + f[x_n, x_{n-1}](x - x_n) \\ &\quad + f[x_n, x_{n-1}, x_{n-2}](x - x_n)(x - x_{n-1}) + \cdots \\ &\quad + f[x_n, \ldots, x_0](x - x_n)(x - x_{n-1}) \cdots (x - x_1). \end{aligned}$$

- If the nodes are equally spaced with

$$h = \frac{x_n - x_0}{n}, \qquad x_i = x_n - (n-i)h, \qquad x = x_n + sh,$$

then

$$\begin{aligned} P_n(x) &= f[x_n] + \sum_{k=1}^{n} f[x_n, x_{n-1}, \ldots, x_{n-k}] sh(s+1)h \cdots (s+k-1)h \\ &= f(x_n) + \sum_{k=1}^{n} f[x_n, x_{n-1}, \ldots, x_{n-k}](-1)^k k! \binom{-s}{k} h^k \end{aligned}$$

- The first few terms in the above summation gives good lower degree approximation for x near x_n.

- Binomial formula:

$$\binom{-s}{k} = \frac{-s(-s-1)\cdots(-s-k+1)}{k!} = (-1)^k \frac{s(s+1)\cdots(s+k-1)}{k!}.$$

- Backward-difference:

$$\nabla^k f(x_i) = \nabla^{k-1} f(x_i) - \nabla^{k-1} f(x_{i-1}) = \nabla\left(\nabla^{k-1} f(x_i)\right),$$

then

$$f[x_n, x_{n-1}] = \frac{1}{h}\nabla f(x_n), \qquad f[x_n, x_{n-1}, x_{n-2}] = \frac{1}{2h^2}\nabla^2 f(x_n),$$

and, in general,

$$f[x_n, x_{n-1}, \ldots, x_{n-k}] = \frac{1}{k!\, h^k}\nabla^k f(x_n).$$

- Newton backward-difference formula:

$$P_n(x) = f(x_0) + \sum_{k=1}^{n} (-1)^k \binom{-s}{k} \nabla^k f(x_k).$$

Hermite Interpolation

Given $n+1$ data points $x_0 < x_1 < \cdots < x_n$, and

$$
\begin{array}{cccc}
y_0^{(0)} = f(x_0) & y_1^{(0)} = f(x_1) & \cdots & y_n^{(0)} = f(x_n) \\
y_0^{(1)} = f'(x_0) & y_1^{(1)} = f'(x_1) & \cdots & y_n^{(1)} = f'(x_n) \\
\vdots & \vdots & & \vdots \\
y_0^{(m_0)} = f^{(m_0)}(x_0) & y_1^{(m_1)} = f^{(m_1)}(x_1) & \cdots & y_n^{(m_n)} = f^{(m_n)}(x_n) \\
\downarrow & \downarrow & & \downarrow \\
m_0 + 1 \text{ conditions} & m_1 + 1 \text{ conditions} & \cdots & m_n + 1 \text{ conditions}
\end{array}
$$

for some function $f \in C^m[a,b]$, where $m = \max\{m_0, m_1, \ldots, m_n\}$.

- Determine a polynomial P of degree at most N, where

$$N = \left(\sum_{i=0}^{n} m_i\right) + n, \tag{5}$$

to satisfy the following interpolation conditions:

$$P^{(k)}(x_i) = y_i^{(k)}, \qquad k = 0, 1, \ldots m_i, \quad i = 0, 1, \ldots, n. \tag{6}$$

- If $n = 0$, then P is the m_0th Taylor polynomial for f at x_0.
- If $m_i = 0$ for each i, then P is the nth Lagrange polynomial interpolating f on $x_0, \ldots, x_n$.
- If $m_i = 1$ for each i, then P is called the Hermite polynomial.

Theorem

If $f \in C^1[a,b]$ *and* $x_0, \ldots, x_n \in [a,b]$ *are distinct, then the polynomial of least degree agreeing with* f *and* f' *at* $x_0, \ldots, x_n$ *is unique and is given by*

$$H_{2n+1}(x) = \sum_{j=0}^{n} f(x_j) H_{n,j}(x) + \sum_{j=0}^{n} f'(x_j) \widehat{H}_{n,j}(x),$$

where

$$H_{n,j}(x) = \left[1 - 2(x - x_j) L'_{n,j}(x_j)\right] L^2_{n,j}(x), \quad \widehat{H}_{n,j}(x) = (x - x_j) L^2_{n,j}(x),$$

and

$$L_{n,j}(x) = \frac{(x - x_0) \cdots (x - x_{j-1})(x - x_{j+1}) \cdots (x - x_n)}{(x_j - x_0) \cdots (x_j - x_{j-1})(x_j - x_{j+1}) \cdots (x_j - x_n)}.$$

Moreover, if $f \in C^{2n+2}[a,b]$*, then* $\exists\ \xi(x) \in [a,b]$ *s.t.*

$$f(x) = H_{2n+1}(x) + \frac{(x - x_0)^2 \cdots (x - x_n)^2}{(2n+2)!} f^{(2n+2)}(\xi(x)).$$

Proof: The representation

$$H_{2n+1}(x) = \sum_{j=0}^{n} f(x_j) H_{n,j}(x) + \sum_{j=0}^{n} f'(x_j) \widehat{H}_{n,j}(x),$$

suggests that it suffices to construct $H_{n,j}(x)$ and $\hat{H}_{n,j}(x)$ with

$$\begin{cases} H_{n,j}(x_j) = 1 \\ H'_{n,j}(x_j) = 0 \end{cases}, \quad H_{n,j}(x_i) = H'_{n,j}(x_i) = 0 \quad \text{if } i \neq j,$$

and

$$\begin{cases} \hat{H}_{n,j}(x_j) = 0 \\ \hat{H}'_{n,j}(x_j) = 1 \end{cases}, \quad \hat{H}_{n,j}(x_i) = \hat{H}'_{n,j}(x_i) = 0 \quad \text{if } i \neq j,$$

It is easy to see that $\deg H_{n,j} \leq 2n+1$ and $\deg H_{n,j} \leq 2n+1$.
Since $\deg L^2_{n,j} = 2n$ and

$$L^2_{n,j}(x_i) = (L^2_{n,j})'(x_i) = 0, \text{for } i \neq j$$

We can simply seek for $H_{n,j}(x)$ and $\hat{H}_{n,j}(x)$ of the form

$$H_{n,j}(x) = (a(x - x_j) + b) L^2_{n,j}(x), \quad \hat{H}_{n,j}(x) = (\hat{a}(x - x_j) + \hat{b}) L^2_{n,j}(x)$$

The coefficients a, b and $\hat{a}$, $\hat{b}$ can be easily solved from the conditions

$$H_{n,j}(x_i) = 1, \quad H'_{n,j}(x_i) = 0,$$

and

$$\hat{H}_{n,j}(x_i) = 0, \quad \hat{H}'_{n,j}(x_i) = 1,$$

respectively. □

Proof of uniqueness:

- Since $\deg(P) \le 2n+1$, write

$$P(x) = a_0 + a_1 x + \cdots + a_{2n+1} x^{2n+1}.$$

- $2n+2$ coefficients, $a_0, a_1, \ldots, a_{2n+1}$, to be determined and $2n+2$ conditions given

$$P(x_i) = f(x_i), \quad P'(x_i) = f'(x_i), \quad \text{for } i = 0, \ldots, n.$$

$\Rightarrow$ $2n+2$ linear equations in $2n+2$ unknowns to solve
$\Rightarrow$ show that the coefficient matrix A of this system is nonsingular.

- To prove A is nonsingular, it suffices to prove that $Au = 0$ has only the trivial solution $u = 0$.
- $Au = 0$ iff

$$P(x_i) = 0, \quad P'(x_i) = 0, \quad \text{for } i = 0, \ldots, n.$$

$\Rightarrow$ P is a multiple of the polynomial given by

$$q(x) = \prod_{i=0}^{n} (x - x_i)^2 .$$

- However, $\deg(q) = 2n + 2$ whereas P has degree at most N.
- Therefore, $P(x) = 0$, i.e. $u = 0$.
- That is, A is nonsingular, and the Hermite interpolation problem has a unique solution.

Divided Difference Method for Hermite Interpolation

Given the $2n+2$ condition pairs

$$(x_0, f(x_0)), (x_0, f'(x_0)), (x_1, f(x_1)), (x_1, f'(x_1)), \cdots, (x_n, f(x_n)), (x_n, f'(x_n$$

Rename the x-coordinates as $z_0, z_1, \cdots, z_{2n+1}$, where

$$z_0 = z_1 = x_0, z_2 = z_3 = x_1, \cdots, z_{2n+1} = z_{2n+2} = x_n.$$

Note that $z_0 \leq z_1 \leq \cdots \leq z_N$. If z_j were distinct, then the unique Hermite interpolating polynomial in Newton's form is given by

$$H_{2n+1}(x) = f[z_0] + \sum_{k=1}^{2n+2} f[z_0, z_1, \ldots, z_k](x - z_0)(x - z_1)\cdots(x - z_{k-1}).$$

When $k \geq 2$, $z_i \neq z_{i+k}$, the k-th divided difference is well defined:

$$f[z_i, z_{i+1}, \ldots, z_{i+k}] = \frac{f[z_{i+1}, z_{i+2}, \ldots, z_{i+k}] - f[z_i, z_{i+1}, \ldots, z_{i+k-1}]}{z_{i+k} - z_i}.$$

However the first divided-difference formula has to be modified since $z_{2i} = z_{2i+1} = x_i$ for each i. Let

$$z_{2i} = x_i, \qquad z^{\epsilon}_{2i+1} = x_i + \epsilon.$$

and let $\epsilon \to 0$. Formally, it suffices to replace the first divided differences by

$$f[z_{2i}, z_{2i+1}] := \lim_{\epsilon \to 0} f[z_{2i}, z^{\epsilon}_{2i+1}] = f'(z_{2i}) = f'(x_i)$$

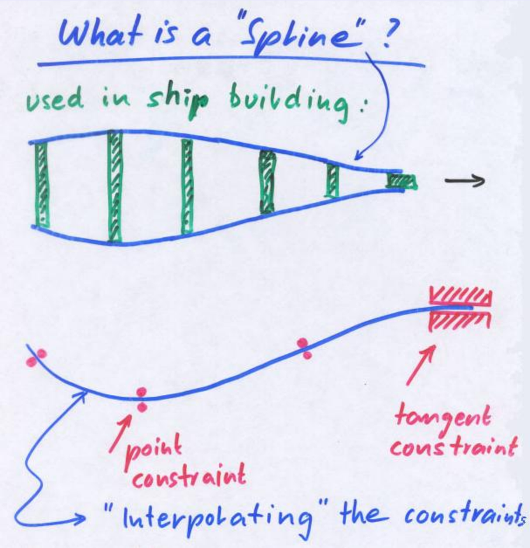
z	$f(z)$		
$z_0 = x_0$	$f[z_0] = f(x_0)$		
		$f[z_0, z_1^\epsilon] = \frac{f[z_1^\epsilon]-f[z_0]}{z_1^\epsilon - z_0}$	
$z_1^\epsilon = x_0 + \epsilon$	$f[z_1^\epsilon] = f(z_1^\epsilon)$		$f[z_0, z_1^\epsilon, z_2] = \frac{f[z_1^\epsilon, z_2]-f[z_0, z_1^\epsilon]}{z_2 - z_0}$
		$f[z_1^\epsilon, z_2] = \frac{f[z_2]-f[z_1^\epsilon]}{z_2 - z_1^\epsilon}$	
$z_2 = x_1$	$f[z_2] = f(x_1)$		$f[z_1^\epsilon, z_2, z_3^\epsilon] = \frac{f[z_2, z_3^\epsilon]-f[z_1^\epsilon, z_2]}{z_3^\epsilon - z_1^\epsilon}$
		$f[z_2, z_3^\epsilon] = \frac{f[z_3^\epsilon]-f[z_2]}{z_3^\epsilon - z_2}$	
$z_3^\epsilon = x_1 + \epsilon$	$f[z_3^\epsilon] = f(z_3^\epsilon)$		$f[z_2, z_3^\epsilon, z_4] = \frac{f[z_3^\epsilon, z_4]-f[z_2, z_3^\epsilon]}{z_4 - z_2}$
		$f[z_3^\epsilon, z_4] = \frac{f[z_4]-f[z_3^\epsilon]}{z_4 - z_3^\epsilon}$	
$z_4 = x_2$	$f[z_4] = f(x_2)$		$f[z_3^\epsilon, z_4, z_5^\epsilon] = \frac{f[z_4, z_5^\epsilon]-f[z_3^\epsilon, z_4]}{z_5^\epsilon - z_3^\epsilon}$
		$f[z_4, z_5^\epsilon] = \frac{f[z_5^\epsilon]-f[z_4]}{z_5^\epsilon - z_4}$	
$z_5^\epsilon = x_2 + \epsilon$	$f[z_5^\epsilon] = f(z_5^\epsilon)$		

As $\epsilon \to 0$, $z_1^\epsilon \to z_1 := x_0$, $f[z_1^\epsilon] \to f(x_0)$, $f[z_0, z_1^\epsilon] \to f[z_0, z_1] := f'(x_0)$, etc.

$$
\begin{array}{c|llll}
z & f(z) & & \\
\hline
z_0 = x_0 & f[z_0] = f(x_0) & & \\
 & & f[z_0, z_1] = f'(x_0) & \\
z_1 = x_0 & f[z_1] = f(x_0) & & f[z_0, z_1, z_2] = \frac{f[z_1,z_2]-f[z_0,z_1]}{z_2-z_0} \\
 & & f[z_1, z_2] = \frac{f[z_2]-f[z_1]}{z_2-z_1} & \\
z_2 = x_1 & f[z_2] = f(x_1) & & f[z_1, z_2, z_3] = \frac{f[z_2,z_3]-f[z_1,z_2]}{z_3-z_1} \\
 & & f[z_2, z_3] = f'(x_1) & \\
z_3 = x_1 & f[z_3] = f(x_1) & & f[z_2, z_3, z_4] = \frac{f[z_3,z_4]-f[z_2,z_3]}{z_4-z_2} \\
 & & f[z_3, z_4] = \frac{f[z_4]-f[z_3]}{z_4-z_3} & \\
z_4 = x_2 & f[z_4] = f(x_2) & & f[z_3, z_4, z_5] = \frac{f[z_4,z_5]-f[z_3,z_4]}{z_5-z_3} \\
 & & f[z_4, z_5] = f'(x_2) & \\
z_5 = x_2 & f[z_5] = f(x_2) & &
\end{array}
$$

$$
H_{2n+1}(x) = f[z_0] + \sum_{k=1}^{2n+2} f[z_0, z_1, \ldots, z_k](x - z_0)(x - z_1)\cdots(x - z_{k-1}).
$$

What is a spline?

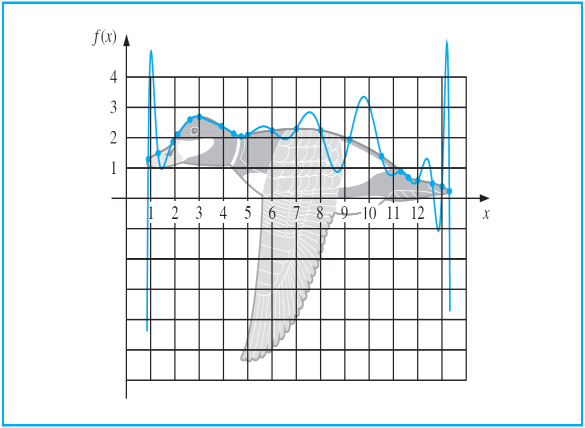


Spline (in mathematics): 'Smooth' Piecewise Polynomial Interpolation.

Why use a Spline?

Interpolation with a single Lagrange polynomial: Runge Phenomena.

Figure 3.13



Polynomial Vs. Piecewise Polynomial Interpolation

- The previous sections concern the approximation of an arbitrary function on a closed interval by a polynomial. However, the oscillatory nature of high-degree polynomials restricts their use.
- Piecewise polynomial interpolation: divide the interval into a collection of sub-intervals and construct different approximation on each sub-interval.
- The simplest piecewise polynomial approximation is piecewise linear interpolation.
- A disadvantage of linear function approximation is that the interpolating function is not smooth at each of the endpoints of the subintervals. It is often required that the approximating function is continuously differentiable.
- An alternative procedure is to use a piecewise polynomial of Hermite type. However, to use Hermite piecewise polynomials for general interpolation, we need to know the derivatives of the function being approximated, which is frequently not available.

Cubic spline interpolation

- The most common piecewise-polynomial use cubic polynomials between successive pairs of nodes, called cubic spline interpolation.

Definition

Given a function f defined on $[a, b]$, and a set of nodes $a = x_0 < x_1 < \cdots < x_n = b$, a cubic spline interpolation S for f is a function that satisfies the following conditions:

(a) S is a cubic polynomial, denoted $S_j(x)$, on $[x_j, x_{j+1}]$ for each $j = 0, 1, \ldots, n-1$;

(b) $S_j(x_j) = f(x_j)$ and $S_j(x_{j+1}) = f(x_{j+1}) \; \forall \; j = 0, 1, \ldots, n-1$;

(c) $S_{j+1}(x_{j+1}) = S_j(x_{j+1}) \forall \; j = 0, 1, \ldots, n-2$;

(d) $S'_{j+1}(x_{j+1}) = S'_j(x_{j+1}) \forall \; j = 0, 1, \ldots, n-2$;

(e) $S''_{j+1}(x_{j+1}) = S''_j(x_{j+1}) \forall \; j = 0, 1, \ldots, n-2$;

(f) One of the following sets of boundary conditions is satisfied:

  (i) $S''(x_0) = S''(x_n) = 0$ (free or natural boundary);

  (ii) $S'(x_0) = f'(x_0)$ and $S'(x_n) = f'(x_n)$ (clamped boundary).

Construction of cubic spline interpolation

Remark

In general, clamped boundary conditions lead to more accurate approximations since they include more information about the function.

Write

$$S_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3, \forall\ j = 0, 1, \ldots, n-1.$$

That is

$$a_j = S_j(x_j) = f(x_j).$$

Since $S_{j+1}(x_{j+1}) = S_j(x_{j+1})$, it implies that

$$a_{j+1} = a_j + b_j(x_{j+1} - x_j) + c_j(x_{j+1} - x_j)^2 + d_j(x_{j+1} - x_j)^3.$$

Let

$$h_j = x_{j+1} - x_j, \forall\ j = 0, 1, \ldots, n-1.$$

Then

$$a_{j+1} = a_j + b_j h_j + c_j h_j^2 + d_j h_j^3. \quad (7)$$

Define $b_n = S'(x_n)$ and observe that

$$S_j'(x) = b_j + 2c_j(x - x_j) + 3d_j(x - x_j)^2$$

implies that

$$b_j = S_j'(x_j), \ \forall \ j = 0, 1, \ldots, n-1.$$

Applying $S_{j+1}'(x_{j+1}) = S_j'(x_{j+1})$ gives

$$b_{j+1} = b_j + 2c_j h_j + 3d_j h_j^2, \ \forall \ j = 0, 1, \ldots, n-1. \tag{8}$$

Define $c_n = S''(x_n)/2$ and apply $S_{j+1}''(x_{j+1}) = S_j''(x_{j+1})$, we get

$$c_{j+1} = c_j + 3d_j h_j, \ \forall \ j = 0, 1, \ldots, n-1.$$

Hence

$$d_j = \frac{c_{j+1} - c_j}{3h_j}. \tag{9}$$

Substituting d_j in (9) into (7) and (8), it obtains

$$a_{j+1} = a_j + b_j h_j + \frac{h_j^2}{3}(2c_j + c_{j+1}) \tag{10}$$

and

$$b_{j+1} = b_j + h_j(c_j + c_{j+1}). \tag{11}$$

Eq. (10) implies that

$$b_j = \frac{1}{h_j}(a_{j+1} - a_j) - \frac{h_j}{3}(2c_j + c_{j+1}), \tag{12}$$

and then

$$b_{j-1} = \frac{1}{h_{j-1}}(a_j - a_{j-1}) - \frac{h_{j-1}}{3}(2c_{j-1} + c_j). \tag{13}$$

Substituting (12) and (13) into (11), we get

$$h_{j-1}c_{j-1} + 2(h_{j-1} + h_j)c_j + h_j c_{j+1} = \frac{3(a_{j+1} - a_j)}{h_j} - \frac{3(a_j - a_{j-1})}{h_{j-1}}. \quad (14)$$

for each $j = 1, 2, \ldots, n-1$.

Theorem (Natural boundary condition)

If f is defined at $a = x_0 < x_1 < \cdots < x_n = b$, then f has a unique natural spline interpolant S on the nodes $x_0, x_1, \ldots, x_n$ that satisfies $S''(a) = S''(b) = 0$.

Proof: The boundary conditions imply that

$$\begin{aligned} c_n &:= S''(x_n)/2 = 0, \\ 0 &= S''(x_0) = 2c_0 + 6d_0(x_0 - x_0) \;\Rightarrow\; c_0 = 0. \end{aligned}$$

Substituting $c_0 = c_n = 0$ into (14), it produces a linear system $Ax = b$, where A is an $(n+1) \times (n+1)$ matrix

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

$$b = \begin{bmatrix} 0 \\ \frac{3}{h_1}(a_2 - a_1) - \frac{3}{h_0}(a_1 - a_0) \\ \vdots \\ \frac{3}{h_{n-1}}(a_n - a_{n-1}) - \frac{3}{h_{n-2}}(a_{n-1} - a_{n-2}) \\ 0 \end{bmatrix}, \quad x = \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{bmatrix}.$$

The matrix A is strictly diagonally dominant. It is not difficult to show that $x^T A x = \langle x, Ax \rangle > 0$ as long as $x \neq 0$ (see also the clamped case below). Thus A nonsingular and the linear system has a unique solution for $c_0, c_1, \ldots, c_n$.

Theorem (Clamped boundary condition)

If f is defined at $a = x_0 < x_1 < \cdots < x_n = b$ and differentiable at a and b, then f has a unique clamped spline interpolant S on the nodes $x_0, x_1, \ldots, x_n$ that satisfies $S'(a) = f'(a)$ and $S'(b) = f'(b)$.

Proof: Since $f'(a) = S'(a) = S'(x_0) = b_0$, Eq. (12) with $j = 0$ implies

$$f'(a) = \frac{1}{h_0}(a_1 - a_0) - \frac{h_0}{3}(2c_0 + c_1).$$

Consequently,

$$2h_0c_0 + h_0c_1 = \frac{3}{h_0}(a_1 - a_0) - 3f'(a).$$

Similarly,

$$f'(b) = b_n = b_{n-1} + h_{n-1}(c_{n-1} + c_n),$$

so Eq. (12) with $j = n - 1$ implies that

$$\begin{aligned} f'(b) &= \frac{a_n - a_{n-1}}{h_{n-1}} - \frac{h_{n-1}}{3}(2c_{n-1} + c_n) + h_{n-1}(c_{n-1} + c_n) \\ &= \frac{a_n - a_{n-1}}{h_{n-1}} + \frac{h_{n-1}}{3}(c_{n-1} + 2c_n), \end{aligned}$$

and

$$h_{n-1}c_{n-1} + 2h_{n-1}c_n = 3f'(b) - \frac{3}{h_{n-1}}(a_n - a_{n-1}).$$

Eq. (14) together with the equations

$$2h_0c_0 + h_0c_1 = \frac{3}{h_0}(a_1 - a_0) - 3f'(a)$$

and

$$h_{n-1}c_{n-1} + 2h_{n-1}c_n = 3f'(b) - \frac{3}{h_{n-1}}(a_n - a_{n-1})$$

determine the linear system $Ax = b$, where

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

$$b = \begin{bmatrix} \frac{3}{h_0}(a_1 - a_0) - 3f'(a) \\ \frac{3}{h_1}(a_2 - a_1) - \frac{3}{h_0}(a_1 - a_0) \\ \vdots \\ \frac{3}{h_{n-1}}(a_n - a_{n-1}) - \frac{3}{h_{n-2}}(a_{n-1} - a_{n-2}) \\ 3f'(b) - \frac{3}{h_{n-1}}(a_n - a_{n-1}) \end{bmatrix}, \quad x = \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{bmatrix}.$$

The linear system $Ax = b$ has a unique solution $x = (c_0, c_1, \dots, c_n)^T$ since A is strictly diagonally dominant and positively definite:

$$x^T A x = \sum_{j=1}^{n-1} h_j (c_j^2 + c_{j+1}^2) + \sum_{j=0}^{n-1} h_j (c_j + c_{j+1})^2. \quad \square$$

Appendix: Solving tridiagonal linear system

A tridiagonal linear system is solved most efficiently by Gauss elimination, or equivalently the following LU decomposition. We seek to decompose A into the form

$$A = LU \equiv \begin{bmatrix} \ell_{11} & & & \\ \ell_{21} & \ell_{22} & & \\ & \ddots & \ddots & \\ & & \ell_{n,n-1} & \ell_{nn} \end{bmatrix} \begin{bmatrix} 1 & u_{12} & & & \\ & 1 & u_{23} & & \\ & & \ddots & \ddots & \\ & & & \ddots & u_{n-1,n} \\ & & & & 1 \end{bmatrix}.$$

The existence of such decomposition follows from comparing the matrices on both sides. That is, we try to find ℓ's and u's from

$$\begin{aligned} a_{11} &= \ell_{11}, \\ a_{i,i-1} &= \ell_{i,i-1}, \quad \text{for } i = 2,3,\ldots,n, \\ a_{ii} &= \ell_{i,i-1}u_{i-1,i} + \ell_{ii}, \quad \text{for } i = 2,3,\ldots,n, \\ a_{i,i+1} &= \ell_{ii}u_{i,i+1}, \quad \text{for } i = 1,2,\ldots,n-1. \end{aligned}$$

Thus the entries in L and U can be solved systematically as follows. First, we have

$$\ell_{11} = a_{11}, \quad u_{12} = a_{12}/\ell_{11}.$$

Then, for $i = 2, \ldots, n-1$, we can solve the ℓ's and u's by

$$\begin{aligned} \ell_{i,i-1} &= a_{i,i-1}, \\ \ell_{ii} &= a_{ii} - \ell_{i,i-1}u_{i-1,i}, \\ u_{i,i+1} &= a_{i,i+1}/\ell_{ii}. \end{aligned}$$

Finally,

$$\begin{aligned} \ell_{n,n-1} &= a_{n,n-1}, \\ \ell_{nn} &= a_{nn} - \ell_{n,n-1}u_{n-1,n}. \end{aligned}$$

Forward substitution

Given the decomposition $A = LU$, the linear system $Ax = LUx = b$ can be solved in two steps.
Step 1: Solve y from $Ly = b$:

$$\Rightarrow \left\{ \begin{array}{rcl} \ell_{11}y_1 & = & b_1, \\ \ell_{21}y_1 + \ell_{22}y_2 & = & b_2, \\ & \vdots & \\ \ell_{n,n-1}y_{n-1} + \ell_{nn}y_n & = & b_n. \end{array} \right.$$

$$\Rightarrow \left\{ \begin{array}{rcl} y_1 & = & b_1/\ell_{11}, \\ y_2 & = & (b_2 - \ell_{21}y_1)/\ell_{22}, \\ & \vdots & \\ y_n & = & (b_n - \ell_{n,n-1}y_{n-1})/\ell_{nn}. \end{array} \right.$$

$$\Rightarrow \left\{ \begin{array}{rcl} y_1 & = & b_1/\ell_{11}, \\ y_i & = & (b_i - \ell_{i,i-1}y_{i-1})/\ell_{ii}, \quad \text{for } \; i = 2, \ldots, n. \end{array} \right.$$

Back substitution

Step 2: Solve x from $Ux = y$:

$$\Rightarrow \left\{ \begin{array}{rcl} x_1 + u_{12}x_2 & = & y_1, \\ & \vdots & \\ x_{n-1} + u_{n-1,n}x_n & = & y_{n-1}, \\ x_n & = & y_n, \end{array} \right.$$

$$\Rightarrow \left\{ \begin{array}{rcl} x_n & = & y_n, \\ x_{n-1} & = & y_{n-1} - u_{n-1,n}x_n, \\ & \vdots & \\ x_1 & = & y_1 - u_{12}x_2, \end{array} \right.$$

$$\Rightarrow \left\{ \begin{array}{rcl} x_n & = & y_n, \\ x_i & = & y_i - u_{i,i+1}x_{i+1}, \quad \text{for } i = n-1, \ldots, 1. \end{array} \right.$$