Interpolation and Polynomial Approximation II

APM1137 - Numerical Analysis

Department of Mathematics, Institute of Arts and Sciences
Far Eastern University, Sampaloc, Manila

Hermite Interpolation

For a given function, f, a polynomial that agree with f and f' at $x_0, x_1, \ldots, x_n$ is a **Hermite polynomial.**

Theorem

If $f \in C^1[a, b]$ and $x_0, \ldots, x_n \in [a, b]$ are distinct, the unique polynomial of least degree agreeing with f and f' at $x_0, \ldots, x_n$ is the Hermite polynomial of degree at most $2n + 1$ given by

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

where, for $L_{n,j}(x)$ denoting the jth Lagrange coefficient polynomial of degree n, we have

$$H_{n,j}(x) = [1 - 2(x - x_j)L'_{n,j}(x_j)]L^2_{n,j}(x) \text{ and } \hat{H}_{n,j}(x) = (x - x_j)L^2_{n,j}(x).$$

Theorem continued...

Moreover, if $f \in C^{2n+2}[a, b]$, then

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

for some (generally unknown) $\xi(x)$ in the interval $[a, b]$.

Example

Use the polynomial that agrees with the data listed below to find an approximation of $f(1.5)$

k	x_k	$f(x_k)$	$f'(x_k)$
0	1.3	0.6200860	−0.5220232
1	1.6	0.4554022	−0.5698959
2	1.9	0.2818186	−0.5811571

Solution

$$L_{2,0}(x) = \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} = \frac{50}{9}x^2 - \frac{175}{9}x + \frac{152}{9} \text{ and}$$

$$L'_{2,0} = \frac{100}{9}x - \frac{175}{9}$$

$$L_{2,1}(x) = \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} = -\frac{100}{9}x^2 + \frac{320}{9}x - \frac{247}{9} \text{ and}$$

$$L'_{2,1} = -\frac{200}{9}x + \frac{320}{9}$$

$$
\begin{aligned}
H_{2,0}(x) &= (10x-12)\left(\frac{50}{9}x^2 - \frac{175}{9}x + \frac{152}{9}\right)^2 \\
H_{2,1}(x) &= 1\cdot\left(-\frac{100}{9}x^2 + \frac{320}{9}x - \frac{247}{9}\right)^2 \\
H_{2,2}(x) &= 10(2-x)\left(\frac{50}{9}x^2 - \frac{145}{9}x + \frac{104}{9}\right)^2 \\
\hat{H}_{2,0}(x) &= (x-1.3)\left(\frac{50}{9}x^2 - \frac{175}{9}x + \frac{152}{9}\right)^2 \\
\hat{H}_{2,1}(x) &= (x-1.6)\left(-\frac{100}{9}x^2 + \frac{320}{9}x - \frac{247}{9}\right)^2 \\
\hat{H}_{2,2}(x) &= (x-1.9)\left(\frac{50}{9}x^2 - \frac{145}{9}x + \frac{104}{9}\right)^2
\end{aligned}
$$

Thus

$$
\begin{gathered}
H_5 = 0.6200860H_{2,0}(x) + 0.4554022H_{2,1}(x) + 0.2818186H_{2,2}(x) - \\
0.522023\hat{H}_{2,0}(x) - 0.5698959\hat{H}_{2,1}(x) - 0.5811571\hat{H}_{2,2}(x)
\end{gathered}
$$

and $H_5(1.5) = 0.5118277$

Hermite Interpolation

To obtain the coefficients of the Hermite interpolating polynomial $H(x)$ on the $(n+1)$ distinct numbers $x_0, \ldots, x_n$ for the function f:

INPUT numbers $x_0, \ldots, x_n$; values $f(x_0), \ldots, f(x_n)$ and $f'(x_0), \ldots, f'(x_n)$.

OUTPUT numbers $Q_{0,0}, Q_{1,1}, \ldots, Q_{2n+1,2n+1}$ where

$$
\begin{aligned}
H(x) \quad = \quad & Q_{0,0} + Q_{1,1}(x-x_0) + Q_{2,2}(x-x_0)^2 + Q_{3,3}(x-x_0)^2(x-x_1) \\
& + Q_{4,4}(x-x_0)^2(x-x_1)^2 + \cdots \\
& + Q_{2n+1,2n+1}(x-x_0)^2(x-x_1)^2 \cdots (x-x_{n-1})^2(x-x_n).
\end{aligned}
$$

Step 1 For $i = 0, 1, \ldots, n$ do Steps 2 and 3.

Step 2 Set $z_{2i} = x_i$;

$z_{2i+1} = x_i$;

$Q_{2i,0} = f(x_i)$;

$Q_{2i+1,0} = f(x_i)$;

$Q_{2i+1,1} = f'(x_i)$.

Step 3 If $i \neq 0$ then set

$$Q_{2i,1} = \frac{Q_{2i,0} - Q_{2i-1,0}}{z_{2i} - z_{2i-1}}.$$

Step 4 For $i = 2, 3, \ldots, 2n + 1$

for $j = 2, 3, \ldots, i$ set $Q_{i,j} = \dfrac{Q_{i,j-1} - Q_{i-1,j-1}}{z_i - z_{i-j}}$.

Step 5 OUTPUT $(Q_{0,0}, Q_{1,1}, \ldots, Q_{2n+1,2n+1})$;
STOP

Cubic Spline Interpolation

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 interpolant** S for f is a function that satisfies the following conditions:

1. $S(X)$ is a cubic polynomial, denoted $S_j(x)$, on the subinterval $[x_j, x_{j+1}]$ for each $j = 0, 1, \ldots, n-1$
2. $S_j(x_j) = f(x_j)$ and $S_j(x_{j+1}) = f(x_{j+1})$ for each $j = 0, 1, \ldots, n-1$
3. $S_{j+1}(x_{j+1}) = S_j(x_{j+1})$ for each $j = 0, 1, \ldots, n-2$
4. $S'_{j+1}(x_{j+1}) = S'_j(x_{j+1})$ for each $j = 0, 1, \ldots, n-2$
5. $S''_{j+1}(x_{j+1}) = S''_j(x_{j+1})$ for each $j = 0, 1, \ldots, n-2$
6. One of the following sets of boundary conditions is satisfied:
    1. $S''(x_0) = S''(x_n) = 0$ (**natural** *(or free)* **boundary**)
    2. $S'(x_0) = f'(x_0)$ and $S'(x_n) = f'(x_n)$ (**clamped boundary**)

Construction of a Cubic Spline

A spline defined on an interval that is divided into n subintervals will require determining $4n$ constants. To construct the cubic spline interpolant for a given function f , the conditions in the definition are applied to the cubic polynomials

$$S_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d(x - x_j)^3$$

for each $j = 0, 1, \ldots, n - 1$.

Since $S_j(x_j) = a_j = f(x_j)$, then

$$a_{j+1} = S_{j+1}(x_{j+1}) = S_j(x_{j+1}) =$$
$$a_j + b_j(x_{j+1} - x_j) + c_j(x_{j+1} - x_j)^2 + d(x_{j+1} - x_j)^3$$

for each $j = 0, 1, \ldots, n - 2$.

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

If we also define $a_n = f(x_n)$, then

$$a_{j+1} = a_j + b_j h_j + c_j h_j^2 + d_j h_j^3$$

holds for each $j = 0, 1, \ldots, n-1$.

Define $b_n = S'(x_n)$ and observe that $S'(x_j) = b_j$ for each $j = 0, 1, \ldots, n-1$. Thus

$$b_{j+1} = b_j + 2c_j h_j + 3d_j h_j^2,$$

for each $j = 0, 1, \ldots, n-1$.
Other relationships obtained from the conditions are

$$c_{j+1} = c_j + 3d_j h_j$$

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

for each $j = 0, 1, \ldots, n-1$.

Natural Cubic Spline

To construct the cubic spline interpolant S for the function f, defined at the numbers $x_0 < x_1 < \cdots < x_n$, satisfying $S''(x_0) = S''(x_n) = 0$

INPUT $n; x_0, x_1, \ldots, x_n; a_0 = f(x_0), a_1 = f(x_1), \ldots, a_n = f(x_n)$

OUTPUT a_j, b_j, c_j, d_j for $j = 0, 1, \ldots, n-1$.

Note:

$$S_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d(x - x_j)^3$$

for $x_j \leq x \leq x_{j+1}$.

STEP 1 For $i = 0, 1, \ldots, n-1$ set $h_i = x_{j+1} - x_j$

Step 2 For $i = 1, 2, \ldots, n-1$ set

$$\alpha_i = \frac{3}{h_i}(a_{i+1} - a_i) - \frac{3}{h_{i-1}}(a_i - a_{i-1}).$$

Step 3 Set $l_0 = 1$; *(Steps 3, 4, 5, and part of Step 6 solve a tridiagonal linear system using a method described in Algorithm 6.7.)*

$\mu_0 = 0$;
$z_0 = 0$.

Step 4 For $i = 1, 2, \ldots, n-1$
set $l_i = 2(x_{i+1} - x_{i-1}) - h_{i-1}\mu_{i-1}$;
$\mu_i = h_i / l_i$;
$z_i = (\alpha_i - h_{i-1} z_{i-1}) / l_i$.

Step 5 Set $l_n = 1$;
$z_n = 0$;
$c_n = 0$.

Step 6 For $j = n-1, n-2, \ldots, 0$
set $c_j = z_j - \mu_j c_{j+1}$;
$b_j = (a_{j+1} - a_j)/h_j - h_j(c_{j+1} + 2c_j)/3$;
$d_j = (c_{j+1} - c_j)/(3h_j)$.

Step 7 OUTPUT (a_j, b_j, c_j, d_j for $j = 0, 1, \ldots, n-1$);
STOP.

Clamped Cubic Spline

To construct the cubic spline interpolant S for the function f, defined at the numbers $x_0 < x_1 < \cdots < x_n$, satisfying $S'(x_0) = f'(x_0)$ and $S'(x_n) = f'(x_n)$

INPUT $n; x_0, x_1, \ldots, x_n; a_0 = f(x_0), a_1 = f(x_1), \ldots, a_n = f(x_n)$; $FP0 = f'(x_0)$; $FPN = f'(x_n)$

OUTPUT a_j, b_j, c_j, d_j for $j = 0, 1, \ldots, n-1$.

Note:

$$S_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d(x - x_j)^3$$

for $x_j \leq x \leq x_{j+1}$.

STEP 1 For $i = 0, 1, \ldots, n-1$ set $h_i = x_{j+1} - x_j$

Clamped Cubic Spline

Step 1 For $i = 0, 1, \ldots, n-1$ set $h_i = x_{i+1} - x_i$.

Step 2 Set $\alpha_0 = 3(a_1 - a_0)/h_0 - 3FPO$;
$\alpha_n = 3FPN - 3(a_n - a_{n-1})/h_{n-1}$.

Step 3 For $i = 1, 2, \ldots, n-1$

$$\text{set } \alpha_i = \frac{3}{h_i}(a_{i+1} - a_i) - \frac{3}{h_{i-1}}(a_i - a_{i-1}).$$

Step 4 Set $l_0 = 2h_0$; *(Steps 4,5,6, and part of Step 7 solve a tridiagonal linear system using a method described in Algorithm 6.7.)*
$\mu_0 = 0.5$;
$z_0 = \alpha_0 / l_0$.

Step 5 For $i = 1, 2, \ldots, n-1$
set $l_i = 2(x_{i+1} - x_{i-1}) - h_{i-1}\mu_{i-1}$;
$\mu_i = h_i / l_i$;
$z_i = (\alpha_i - h_{i-1} z_{i-1}) / l_i$.

Clamped Cubic Spline

Step 6 Set $l_n = h_{n-1}(2 - \mu_{n-1})$;
$z_n = (\alpha_n - h_{n-1}z_{n-1})/l_n$;
$c_n = z_n$.

Step 7 For $j = n-1, n-2, \ldots, 0$
set $c_j = z_j - \mu_j c_{j+1}$;
$b_j = (a_{j+1} - a_j)/h_j - h_j(c_{j+1} + 2c_j)/3$;
$d_j = (c_{j+1} - c_j)/(3h_j)$.

Step 8 OUTPUT (a_j, b_j, c_j, d_j for $j = 0, 1, \ldots, n-1$);
STOP.