Interpolation and Polynomial Approximation - Lecture 02

Dr. Serhat Hosder

Associate Professor of Aerospace Engineering

Mechanical and Aerospace Engineering

290B Toomey Hall

Missouri S&T

Rolla, MO 65409

Phone: 573-341-7239

E-mail: hosders@mst.edu

Outline

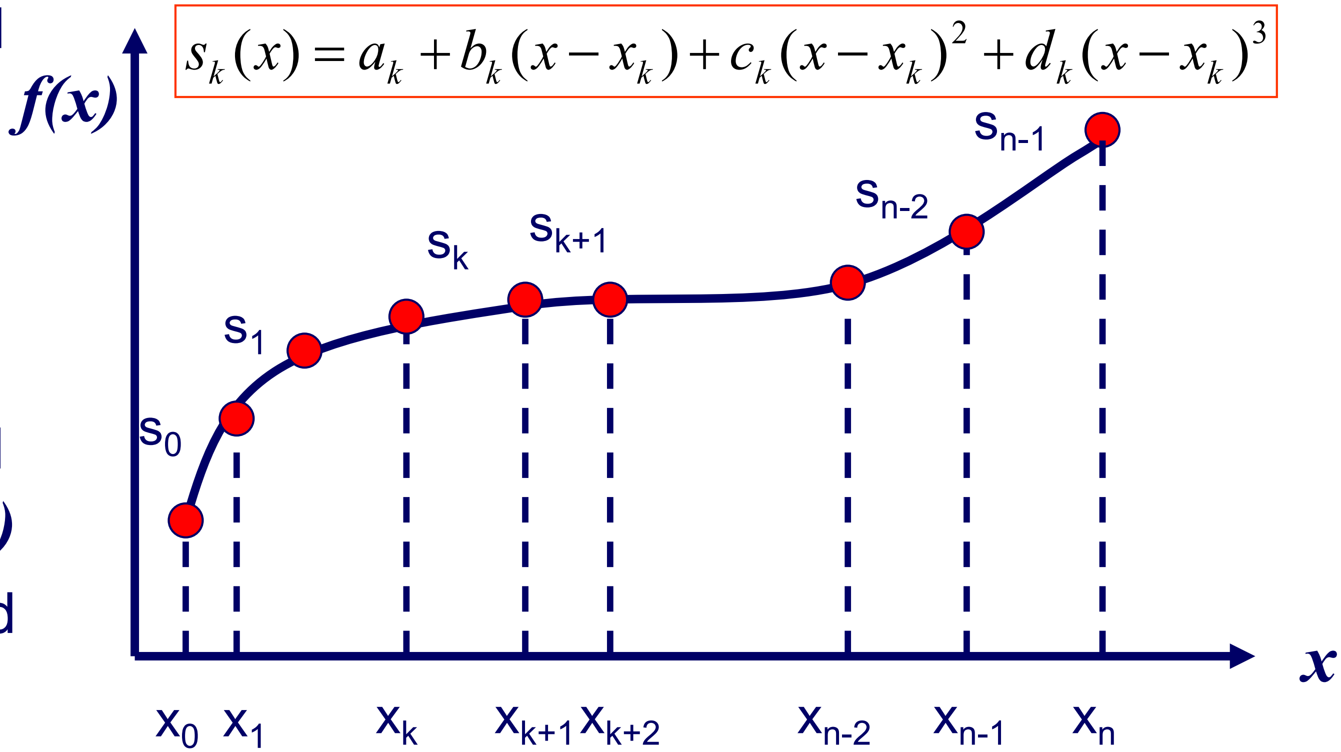
Cubic Splines (Section 3.4 in the textbook)

- Basic idea
  - Construct a set of piecewise cubic polynomials for a global approximation
  - Constrain cubic polynomials to match the function and meet continuity and smoothness requirements
- Derive the spline equations
- Cubic spline linear system – tridiagonal
- Boundary Conditions
- Implementation summary

Cubic Splines

Consider a set of $n+1$ data pairs that form the curve shown.

The conditions imposed for obtaining the cubic spline passing through the given points are:



(a) The spline approximation should match the function at all the points $s_k(x_k) = f(x_k)$

(b) The functions should be continuous

$s_{k+1}(x_{k+1}) = s_k(x_{k+1})$ *where* $k=0,1,..,(n-2)$

(c) Function should be smooth. i.e., $s'_{k+1}(x_{k+1}) = s'_k(x_{k+1})$ $\quad k=0,1,..,(n-2)$

(d) 2nd Derivative should be equal. i.e., $s''_k(x_{k+1}) = s''_{k+1}(x_{k+1})$ $\quad k=0,1,..,(n-2)$

Cubic Spline Derivation (1)

The conditions (a), (b), (c) and (d) are used for arriving at the expressions giving the value of the constants in the general cubic spline equation.

On the k[th] segment, the equation for the cubic is:

$$s_k(x) = a_k + b_k(x - x_k) + c_k(x - x_k)^2 + d_k(x - x_k)^3 \rightarrow (0)$$

From a), we have

$$\boxed{s_k(x_k) = a_k = f(x_k) \rightarrow (1)}$$

Note that the a_k are thus known since $f(x_k)$ is given.

Let's define $h_k = x_{k+1} - x_k$. Then, **from b),** i.e., $\boldsymbol{s_{k+1}(x_{k+1}) = s_k(x_{k+1})}$

we can obtain

$$s_{k+1}(x_{k+1}) = a_{k+1}$$

$$s_k(x_{k+1}) = a_k + b_k h_k + c_k h_k^2 + d_k h_k^3$$

$$\boxed{a_{k+1} = a_k + b_k h_k + c_k h_k^2 + d_k h_k^3 \rightarrow (2)}$$

Cubic Spline Derivation (2)

Given the spline equation,

$$s_k(x) = a_k + b_k(x - x_k) + c_k(x - x_k)^2 + d_k(x - x_k)^3$$

The first derivative is,

$$s'_k(x) = b_k + 2c_k(x\text{-}x_k) + 3d_k(x\text{-}x_k)^2$$

Consequently, $\Rightarrow s'_k(x_k) = b_k$

Then from **condition c),** $\boldsymbol{s'_{k+1}(x_{k+1}) = s'_k(x_{k+1})}$

$$\boxed{b_{k+1} = b_k + 2c_k h_k + 3d_k h_k^2 \rightarrow (3)}$$

Cubic Spline Derivation (3)

Second Derivative

$$\Rightarrow \mathrm{s}''_{\mathrm{k}}(\mathrm{x}) = 2c_k + 6d_k(x - x_k)$$

$$\Rightarrow \mathrm{s}''_{\mathrm{k}}(\mathrm{x}_k) = 2c_k$$

$$\Rightarrow c_k = \frac{\mathrm{s}''_{\mathrm{k}}(\mathrm{x}_k)}{2}$$

$$\text{From (d)} \Rightarrow 2c_k + 6d_k h_k = 2c_{k+1}$$

$$\Rightarrow \boxed{d_k = \frac{c_{k+1} - c_k}{3h_k} \rightarrow (4)}$$

Substituting (4) into (2) yields,

$$a_{k+1} = a_k + b_k h_k + c_k h_k^{\,2} + \frac{c_{k+1} - c_k}{3h_k} h_k^{\,3}$$

Rearranging this equation and solving for b_k,

$$\boxed{b_k = \frac{a_{k+1} - a_k}{h_k} - \frac{h_k}{3}(2c_k + c_{k+1})}$$

Cubic Spline Derivation (4)

Finally substituting for b_k and d_k in (3) and rearranging gives

$$\boxed{\begin{aligned} & h_{k\text{-}1}c_{k\text{-}1} + c_k(2h_{k-1} + h_k) + h_k c_{k+1} = \\ & \frac{3}{h_k}(a_{k+1} - a_k) - \frac{3}{h_{k-1}}(a_k - a_{k-1}) \ \forall \, \mathrm{k} = 1,2,3,\ldots \mathrm{n} - 1 \end{aligned}}$$

Note that the above expression is a tri-diagonal system for the variable c_k with $h_{k\text{-}1}$ on the lower diagonal, $2(h_{k-1} + h_k)$ on the main diagonal and h_k on the upper diagonal with a known right hand side given by

$$\frac{3}{h_k}(a_{k+1} - a_k) - \frac{3}{h_{k-1}}(a_k - a_{k-1})$$

We have **n-1** equations (valid at the interior nodes) in **n+1** unknowns. Thus, we need to supply boundary conditions on each end.

Boundary Conditions

1. Natural Splines:

Natural Splines are spline equations formed with $s'' = 0$ at end points.

$$s''(x_0) = s''(x_n) = 0$$

This implies that

$$c_0 = c_n = 0$$

Equivalent to assuming that the end cubics approach linearity.

2. Clamped Boundary Conditions:

$$s'(x_0) = f'(x_0) \qquad\qquad s'(x_n) = f'(x_n)$$

Note that this requires knowledge of the derivative of the function at the endpoints. In difference form, at the left endpoint this reduces to:

$$2h_0c_0 + h_0c_1 = \frac{3}{h_0}(a_1 - a_0) - 3f'(x_0)$$

For the right endpoint.

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

Exercise: Derive these equations

The Linear System for Natural Cubic Splines

This tri-diagonal system of equation can be written in matrix form as:

$$
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & \dots & 0 \\
h_0 & 2(h_0+h_1) & h_1 & 0 & 0 & 0 & 0 & \dots & \dots & 0 \\
0 & h_1 & 2(h_1+h_2) & h_2 & 0 & 0 & 0 & \dots & \dots & 0 \\
0 & 0 & \dots & \dots & \dots & 0 & 0 & \dots & \dots & 0 \\
0 & 0 & 0 & \dots & \dots & \dots & 0 & \dots & \dots & 0 \\
0 & 0 & 0 & 0 & \dots & \dots & \dots & \dots & \dots & 0 \\
\dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & 0 \\
\dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & h_{n-2} & 2(h_{n-2}+h_{n-1}) & h_{n-1} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
\text{x}
\begin{bmatrix}
c_0 \\ c_1 \\ c_2 \\ . \\ . \\ . \\ . \\ . \\ . \\ c_n
\end{bmatrix}
=
\begin{bmatrix}
0 \\
\frac{3}{h_1}(a_2-a_1)-\frac{3}{h_0}(a_1-a_0) \\
\dots. \\
\dots. \\
\dots. \\
\dots. \\
\dots. \\
\dots. \\
\frac{3}{h_{n-1}}(a_n-a_{n-1})-\frac{3}{h_{n-2}}(a_{n-1}-a_{n-2}) \\
0
\end{bmatrix}
$$

The boundary values used here are for a *natural* cubic spline.

Note, once this system is solved for the c_k, then one can generate b_k and d_k directly and since the a_k are known, this completes the definition of the cubics for each interval.

The Linear System for Cubic Splines with Clamped Ends

This tri-diagonal system of equation can be written in matrix form as:

$$
\begin{bmatrix}
2h_0 & h_0 & 0 & 0 & 0 & 0 & 0 & \dots & \dots & 0 \\
h_0 & 2(h_0+h_1) & h_1 & 0 & 0 & 0 & 0 & \dots & \dots & 0 \\
0 & h_1 & 2(h_1+h_2) & h_2 & 0 & 0 & 0 & \dots & \dots & 0 \\
0 & 0 & \dots & \dots & \dots & 0 & 0 & \dots & \dots & 0 \\
0 & 0 & 0 & \dots & \dots & \dots & 0 & \dots & \dots & 0 \\
0 & 0 & 0 & 0 & \dots & \dots & \dots & \dots & \dots & 0 \\
\dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & 0 \\
\dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & h_{n-2} & 2(h_{n-2}+h_{n-1}) & h_{n-1} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & h_{n-1} & 2h_{n-1}
\end{bmatrix}
\text{x}
\begin{bmatrix}
c_0 \\ c_1 \\ c_2 \\ . \\ . \\ . \\ . \\ . \\ . \\ c_n
\end{bmatrix}
=
\begin{bmatrix}
\frac{3}{h_0}(a_1-a_0)-3f'(x_0) \\
\frac{3}{h_1}(a_2-a_1)-\frac{3}{h_0}(a_1-a_0) \\
\dots. \\
\dots. \\
\dots. \\
\dots. \\
\dots. \\
\dots. \\
\frac{3}{h_{n-1}}(a_n-a_{n-1})-\frac{3}{h_{n-2}}(a_{n-1}-a_{n-2}) \\
3f'(x_n)-\frac{3}{h_{n-1}}(a_n-a_{n-1})
\end{bmatrix}
$$

The boundary values used here are for a cubic spline with *clamped* ends

Implementation Summary

1. Solve the tri-diagonal equation for the coefficients c_k $(k=0,1,2,\ldots,n)$

2. Obtain b_k and d_k $(k=0,1,2,\ldots,n-1)$

$$b_k = \frac{a_{k+1}-a_k}{h_k} - \frac{h_k}{3}\left(2c_k + c_{k+1}\right) \qquad d_k = \frac{c_{k+1}-c_k}{3h_k}$$

3. Note that $a_k=f(x_k)$ $(k=0,1,2,\ldots,n)$

4. Approximate the function with the set of cubic splines

$$f(x) \approx S(x) = \left\{ s_k(x) \mid k = 0,1,\cdots,n-1 \right\}$$

5. For any x, select the correct spline fit (choose k) and its coefficients and interpolate.

Summary

We have

- Derived the equations for cubic splines based on matching the function, and forcing the splines and their first two derivatives to match at interfaces.
- Wrote the tri-diagonal linear system for the spline coefficients, c_k. This effectively yields all coefficients.
- Discussed boundary conditions and their impact on the linear system and the function approximation.