Cubic Spline Interpolation

MATH 375, *Numerical Analysis*

J Robert Buchanan

Department of Mathematics

Spring 2022

History

Given nodes and data $\{(x_0, f(x_0)), (x_1, f(x_1)), \ldots, (x_n, f(x_n))\}$ we have interpolated using:

- Lagrange (or Hermite) interpolating polynomials of degree n (or $2n+1$), with $n+1$ (or $2n+2$) coefficients,

  unfortunately,

History

Given nodes and data $\{(x_0, f(x_0)), (x_1, f(x_1)), \ldots, (x_n, f(x_n))\}$ we have interpolated using:

- Lagrange (or Hermite) interpolating polynomials of degree n (or $2n+1$), with $n+1$ (or $2n+2$) coefficients,

  unfortunately,

- such polynomials can possess large oscillations, and
- the error term can be difficult to construct and estimate.

History

Given nodes and data $\{(x_0, f(x_0)), (x_1, f(x_1)), \ldots, (x_n, f(x_n))\}$ we have interpolated using:

- Lagrange (or Hermite) interpolating polynomials of degree n (or $2n+1$), with $n+1$ (or $2n+2$) coefficients,

  unfortunately,

- such polynomials can possess large oscillations, and
- the error term can be difficult to construct and estimate.

An alternative is **piecewise** polynomial approximation, but of what degree polynomial?

History

Given nodes and data $\{(x_0, f(x_0)), (x_1, f(x_1)), \ldots, (x_n, f(x_n))\}$ we have interpolated using:

- Lagrange (or Hermite) interpolating polynomials of degree n (or $2n+1$), with $n+1$ (or $2n+2$) coefficients,

  unfortunately,

- such polynomials can possess large oscillations, and
- the error term can be difficult to construct and estimate.

An alternative is **piecewise** polynomial approximation, but of what degree polynomial?

- Piecewise linear results are not differentiable at x_i, $i = 0, 1, \ldots, n$.
- Piecewise quadratic results are not twice differentiable at x_i, $i = 0, 1, \ldots, n$.
- Piecewise cubic!

Cubic Splines

- A cubic polynomial $p(x) = a + bx + cx^2 + dx^3$ is specified by 4 coefficients.
- The cubic spline is twice continuously differentiable.
- The cubic spline has the flexibility to satisfy general types of boundary conditions.
- While the spline may agree with $f(x)$ at the nodes, we cannot guarantee the derivatives of the spline agree with the derivatives of f.

Cubic Spline Interpolant (1 of 2)

Given a function $f(x)$ defined on $[a, b]$ and a set of nodes

$$a = x_0 < x_1 < x_2 < \cdots < x_n = b,$$

a **cubic spline interpolant**, $S(x)$, for $f(x)$ is a piecewise cubic polynomial with components $S_j(x)$ defined on $[x_j, x_{j+1}]$ for $j = 0, 1, \ldots, n-1$.

$$S(x) = \begin{cases} a_0 + b_0(x - x_0) + c_0(x - x_0)^2 + d_0(x - x_0)^3 & \text{if } x_0 \le x \le x_1 \\ a_1 + b_1(x - x_1) + c_1(x - x_1)^2 + d_1(x - x_1)^3 & \text{if } x_1 \le x \le x_2 \\ \vdots & \vdots \\ a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3 & \text{if } x_i \le x \le x_{i+1} \\ \vdots & \vdots \\ a_{n-1} + b_{n-1}(x - x_{n-1}) + c_{n-1}(x - x_{n-1})^2 + d_{n-1}(x - x_{n-1})^3 & \text{if } x_{n-1} \le x \le x_n \end{cases}$$

Cubic Spline Interpolant (2 of 2)

The cubic spline interpolant will have the following properties.

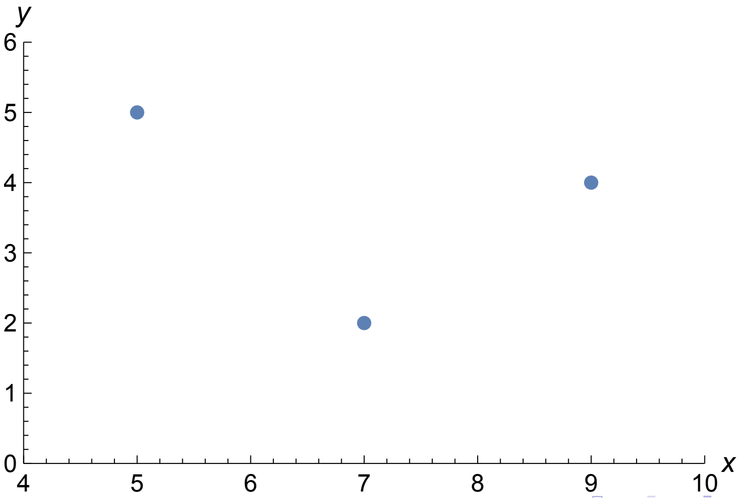
- $S(x_j) = f(x_j)$ for $j = 0, 1, \ldots, n$.
- $S_j(x_{j+1}) = S_{j+1}(x_{j+1})$ for $j = 0, 1, \ldots, n-2$.
- $S'_j(x_{j+1}) = S'_{j+1}(x_{j+1})$ for $j = 0, 1, \ldots, n-2$.
- $S''_j(x_{j+1}) = S''_{j+1}(x_{j+1})$ for $j = 0, 1, \ldots, n-2$.
- One of the following boundary conditions (BCs) is satisfied:
  - $S''(x_0) = S''(x_n) = 0$ (**free** or **natural** BCs).
  - $S'(x_0) = f'(x_0)$ and $S'(x_n) = f'(x_n)$ (**clamped** BCs).

Example (1 of 7)

Construct a piecewise cubic spline interpolant for the curve passing through

$$\{(5,5),\ (7,2),\ (9,4)\},$$

with natural boundary conditions.



Example (2 of 7)

This will require two cubics:

$$S_0(x) = a_0 + b_0(x-5) + c_0(x-5)^2 + d_0(x-5)^3$$
$$S_1(x) = a_1 + b_1(x-7) + c_1(x-7)^2 + d_1(x-7)^3$$

Since there are 8 coefficients, we must derive 8 equations to solve.

Example (2 of 7)

This will require two cubics:

$$
\begin{aligned}
S_0(x) &= a_0 + b_0(x-5) + c_0(x-5)^2 + d_0(x-5)^3 \\
S_1(x) &= a_1 + b_1(x-7) + c_1(x-7)^2 + d_1(x-7)^3
\end{aligned}
$$

Since there are 8 coefficients, we must derive 8 equations to solve.

The splines must agree with the function (the y-coordinates) at the nodes (the x-coordinates).

$$
\begin{aligned}
5 &= S_0(5) = a_0 \\
2 &= S_0(7) = a_0 + 2b_0 + 4c_0 + 8d_0 \\
2 &= S_1(7) = a_1 \\
4 &= S_1(9) = a_1 + 2b_1 + 4c_1 + 8d_1
\end{aligned}
$$

Example (3 of 7)

The first and second derivatives of the cubics must agree at their shared node $x = 7$.

$$
\begin{aligned}
S_0'(7) &= b_0 + 4c_0 + 12d_0 = b_1 = S_1'(7) \\
S_0''(7) &= 2c_0 + 12d_0 = 2c_1 = S_1''(7)
\end{aligned}
$$

Example (3 of 7)

The first and second derivatives of the cubics must agree at their shared node $x = 7$.

$$S_0'(7) = b_0 + 4c_0 + 12d_0 = b_1 = S_1'(7)$$
$$S_0''(7) = 2c_0 + 12d_0 = 2c_1 = S_1''(7)$$

The final two equations come from the natural boundary conditions.

$$S_0''(5) = 0 = 2c_0$$
$$S_1''(9) = 0 = 2c_1 + 12d_1$$

Example (4 of 7)

All eight linear equations together form the system:

$$
\begin{aligned}
5 &= a_0 \\
2 &= a_0 + 2b_0 + 4c_0 + 8d_0 \\
2 &= a_1 \\
4 &= a_1 + 2b_1 + 4c_1 + 8d_1 \\
0 &= b_0 + 4c_0 + 12d_0 - b_1 \\
0 &= 2c_0 + 12d_0 - 2c_1 \\
0 &= 2c_0 \\
0 &= 2c_1 + 12d_1
\end{aligned}
$$

Example (5 of 7)

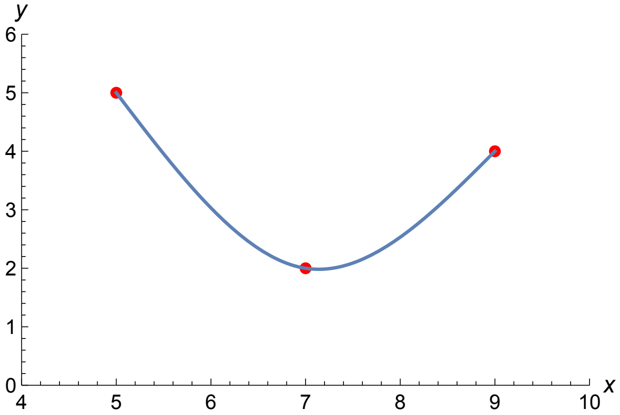
The solution is:

i	a_i	b_i	c_i	d_i
0	5	$-\frac{17}{8}$	0	$\frac{5}{32}$
1	2	$-\frac{1}{4}$	$\frac{15}{16}$	$-\frac{5}{32}$

Example (6 of 7)

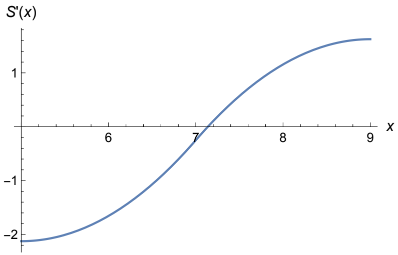
The natural cubic spline can be expressed as:

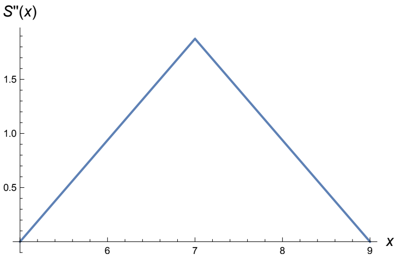
$$S(x) = \begin{cases} 5 - \frac{17}{8}(x-5) + \frac{5}{32}(x-5)^3 & \text{if } 5 \leq x \leq 7 \\ 2 - \frac{1}{4}(x-7) + \frac{15}{16}(x-7)^2 - \frac{5}{32}(x-7)^3 & \text{if } 7 \leq x \leq 9 \end{cases}$$



Example (7 of 7)

We can verify the continuity of the first and second derivatives from the following graphs.





General Construction Process

Given $n + 1$ nodal/data values: $\{(x_0, f(x_0)), (x_1, f(x_1)), \ldots, (x_n, f(x_n))\}$ we will create n cubic polynomials.

General Construction Process

Given $n+1$ nodal/data values: $\{(x_0, f(x_0)), (x_1, f(x_1)), \ldots, (x_n, f(x_n))\}$ we will create n cubic polynomials.

For $j = 0, 1, \ldots, n-1$ assume

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

We must find a_j, b_j, c_j and d_j (a total of $4n$ unknowns) subject to the conditions specified in the definition.

First Set of Equations

Let $h_j = x_{j+1} - x_j$ then

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

So far we know a_j for $j = 0, 1, \ldots, n-1$ and have n equations and $3n$ unknowns.

$$\begin{aligned} a_1 &= a_0 + b_0 h_0 + c_0 h_0^2 + d_0 h_0^3 \\ &\vdots \\ a_{j+1} &= a_j + b_j h_j + c_j h_j^2 + d_j h_j^3 \\ &\vdots \\ a_n &= a_{n-1} + b_{n-1} h_{n-1} + c_{n-1} h_{n-1}^2 + d_{n-1} h_{n-1}^3 \end{aligned}$$

First Derivative

The continuity of the first derivative at the nodal points produces n more equations.

For $j = 0, 1, \ldots, n-1$ we have

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

Thus

$$\begin{aligned} S_j'(x_j) &= b_j \\ S_{j+1}'(x_{j+1}) &= b_{j+1} = S_j'(x_{j+1}) = b_j + 2c_jh_j + 3d_jh_j^2 \end{aligned}$$

Now we have $2n$ equations and $3n$ unknowns.

Equations Derived So Far

$$
\begin{aligned}
a_1 &= a_0 + b_0 h_0 + c_0 h_0^2 + d_0 h_0^3 \\
&\vdots \\
a_{j+1} &= a_j + b_j h_j + c_j h_j^2 + d_j h_j^3 \\
&\vdots \\
a_n &= a_{n-1} + b_{n-1} h_{n-1} + c_{n-1} h_{n-1}^2 + d_{n-1} h_{n-1}^3 \\
b_1 &= b_0 + 2c_0 h_0 + 3d_0 h_0^2 \\
&\vdots \\
b_{j+1} &= b_j + 2c_j h_j + 3d_j h_j^2 \\
&\vdots \\
b_n &= b_{n-1} + 2c_{n-1} h_{n-1} + 3d_{n-1} h_{n-1}^2
\end{aligned}
$$

The unknowns are b_j, c_j, and d_j for $j = 0, 1, \ldots, n-1$.

Second Derivative

The continuity of the second derivative at the nodal points produces n more equations.

For $j = 0, 1, \ldots, n-1$ we have

$$S_j''(x) = 2c_j + 6d_j(x - x_j).$$

Thus

$$\begin{aligned} S_j''(x_j) &= 2c_j \\ S_{j+1}''(x_{j+1}) &= 2c_{j+1} = S_j''(x_{j+1}) = 2c_j + 6d_jh_j \end{aligned}$$

Now we have $3n$ equations and $3n$ unknowns.

Equations Derived So Far

$$
\begin{aligned}
a_{j+1} &= a_j + b_j h_j + c_j h_j^2 + d_j h_j^3 \text{ (for } j = 0, 1, \ldots, n-1) \\
b_{j+1} &= b_j + 2c_j h_j + 3d_j h_j^2 \text{ (for } j = 0, 1, \ldots, n-1) \\
2c_1 &= 2c_0 + 6d_0 h_0 \\
&\vdots \\
2c_{j+1} &= 2c_j + 6d_j h_j \\
&\vdots \\
2c_n &= 2c_{n-1} + 6d_{n-1} h_{n-1}
\end{aligned}
$$

The unknowns are b_j, c_j, and d_j for $j = 0, 1, \ldots, n-1$.

Summary of Equations

For $j = 0, 1, \ldots, n-1$ we have

$$
\begin{aligned}
a_{j+1} &= a_j + b_j h_j + c_j h_j^2 + d_j h_j^3 \\
b_{j+1} &= b_j + 2c_j h_j + 3d_j h_j^2 \\
c_{j+1} &= c_j + 3d_j h_j.
\end{aligned}
$$

Note: The quantities a_j and h_j are known.

Summary of Equations

For $j = 0, 1, \ldots, n-1$ we have

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

Note: The quantities a_j and h_j are known.
Solve the third equation for d_j and substitute into the other two equations.

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

This eliminates n equations of the third type.

Solving the Equations (1 of 3)

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

Solving the Equations (1 of 3)

$$\begin{aligned} a_{j+1} &= a_j + b_j h_j + c_j h_j^2 + d_j h_j^3 \\ &= a_j + b_j h_j + c_j h_j^2 + \left(\frac{c_{j+1} - c_j}{3 h_j} \right) h_j^3 \\ &= a_j + b_j h_j + \frac{h_j^2}{3} (2 c_j + c_{j+1}) \end{aligned}$$

Solving the Equations (1 of 3)

$$\begin{aligned} a_{j+1} &= a_j + b_j h_j + c_j h_j^2 + d_j h_j^3 \\ &= a_j + b_j h_j + c_j h_j^2 + \left(\frac{c_{j+1} - c_j}{3 h_j} \right) h_j^3 \\ &= a_j + b_j h_j + \frac{h_j^2}{3} (2 c_j + c_{j+1}) \\ b_{j+1} &= b_j + 2 c_j h_j + 3 d_j h_j^2 \end{aligned}$$

Solving the Equations (1 of 3)

$$
\begin{aligned}
a_{j+1} &= a_j + b_j h_j + c_j h_j^2 + d_j h_j^3 \\
&= a_j + b_j h_j + c_j h_j^2 + \left(\frac{c_{j+1} - c_j}{3h_j} \right) h_j^3 \\
&= a_j + b_j h_j + \frac{h_j^2}{3}(2c_j + c_{j+1}) \\
b_{j+1} &= b_j + 2c_j h_j + 3d_j h_j^2 \\
&= b_j + 2c_j h_j + 3 \left(\frac{c_{j+1} - c_j}{3h_j} \right) h_j^2 \\
&= b_j + h_j(c_j + c_{j+1})
\end{aligned}
$$

Solving the Equations (1 of 3)

$$
\begin{aligned}
a_{j+1} &= a_j + b_j h_j + c_j h_j^2 + d_j h_j^3 \\
&= a_j + b_j h_j + c_j h_j^2 + \left(\frac{c_{j+1} - c_j}{3h_j} \right) h_j^3 \\
&= a_j + b_j h_j + \frac{h_j^2}{3}(2c_j + c_{j+1}) \\
b_{j+1} &= b_j + 2c_j h_j + 3d_j h_j^2 \\
&= b_j + 2c_j h_j + 3 \left(\frac{c_{j+1} - c_j}{3h_j} \right) h_j^2 \\
&= b_j + h_j(c_j + c_{j+1})
\end{aligned}
$$

Solve the first equation for b_j.

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

Solving the Equations (2 of 3)

We have for $j = 0, 1, \ldots, n-1$,

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

Replace index j by $j - 1$ to obtain

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

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

Solving the Equations (2 of 3)

We have for $j = 0, 1, \ldots, n-1$,

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

Replace index j by $j-1$ to obtain

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

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

We can also re-index the earlier equation

$$b_{j+1} = b_j + h_j(c_j + c_{j+1})$$

to obtain

$$b_j = b_{j-1} + h_{j-1}(c_{j-1} + c_j).$$

Solving the Equations (3 of 3)

$$b_j = b_{j-1} + h_{j-1}(c_{j-1} + c_j).$$

Substitute the equations for b_{j-1} and b_j into the equation above. This step eliminates n equations of the first type.

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

Collect all terms involving c_k to one side.

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

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

Solving the Equations (3 of 3)

$$b_j = b_{j-1} + h_{j-1}(c_{j-1} + c_j).$$

Substitute the equations for b_{j-1} and b_j into the equation above. This step eliminates n equations of the first type.

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

Collect all terms involving c_k to one side.

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

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

Remark: we have $n-1$ equations and $n+1$ unknowns.

Natural Boundary Conditions

If $S''(x_0) = S_0''(x_0) = 2c_0 = 0$ then $c_0 = 0$ and if $S''(x_n) = S_{n-1}''(x_n) = 2c_n = 0$ then $c_n = 0$.

Natural Boundary Conditions

If $S''(x_0) = S_0''(x_0) = 2c_0 = 0$ then $c_0 = 0$ and if $S''(x_n) = S_{n-1}''(x_n) = 2c_n = 0$ then $c_n = 0$.

Theorem

If f is defined at $a = x_0 < x_1 < \cdots < x_n = b$ then f has a unique natural cubic spline interpolant.

Natural BC Linear System (1 of 3)

In matrix form the system of $n+1$ equations has the form $A\mathbf{c} = \mathbf{y}$ where

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

Note: A is a tridiagonal matrix.

Natural BC Linear System (2 of 3)

The vector **y** on the right-hand side is formed as

$$\mathbf{y} = \begin{bmatrix} 0 \\ \frac{3}{h_1}(a_2 - a_1) - \frac{3}{h_0}(a_1 - a_0) \\ \frac{3}{h_2}(a_3 - a_2) - \frac{3}{h_1}(a_2 - a_1) \\ \vdots \\ \frac{3}{h_{n-2}}(a_{n-1} - a_{n-2}) - \frac{3}{h_{n-3}}(a_{n-2} - a_{n-3}) \\ \frac{3}{h_{n-1}}(a_n - a_{n-1}) - \frac{3}{h_{n-2}}(a_{n-1} - a_{n-2}) \\ 0 \end{bmatrix}$$

Note: A is a tridiagonal matrix.

Natural BC Linear System (3 of 3)

$$A \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_{n-1} \\ c_n \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{3}{h_1}(a_2 - a_1) - \frac{3}{h_0}(a_1 - a_0) \\ \frac{3}{h_2}(a_3 - a_2) - \frac{3}{h_1}(a_2 - a_1) \\ \vdots \\ \frac{3}{h_{n-1}}(a_n - a_{n-1}) - \frac{3}{h_{n-2}}(a_{n-1} - a_{n-2}) \\ 0 \end{bmatrix}$$

We solve this linear system of equations using a common algorithm for handling tridiagonal systems.

Natural Cubic Spline Algorithm

INPUT $\{(x_0, f(x_0)), (x_1, f(x_1)), \ldots, (x_n, f(x_n))\}$

STEP 1 For $i = 0, 1, \ldots, n-1$ set $a_i = f(x_i)$; set $h_i = x_{i+1} - x_i$.

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 $l_0 = 1$; set $\mu_0 = 0$; set $z_0 = 0$.

STEP 4 For $i = 1, 2, \ldots, n-1$ set $l_i = 2(x_{i+1} - x_{i-1}) - h_{i-1}\mu_{i-1}$;

set $\mu_i = \dfrac{h_i}{l_i}$; set $z_i = \dfrac{\alpha_i - h_{i-1}z_{i-1}}{l_i}$.

STEP 5 Set $l_n = 1$; set $c_n = 0$; set $z_n = 0$.

STEP 6 For $j = n-1, n-2, \ldots, 0$ set $c_j = z_j - \mu_j c_{j+1}$; set

$b_j = \dfrac{a_{j+1} - a_j}{h_j} - \dfrac{h_j(c_{j+1} + 2c_j)}{3}$; set $d_j = \dfrac{c_{j+1} - c_j}{3h_j}$.

STEP 7 For $j = 0, 1, \ldots, n-1$ OUTPUT a_j, b_j, c_j, d_j.

Example (1 of 4)

Construct the natural cubic spline interpolant for $f(x) = \ln(e^x + 2)$ with nodal values:

x	$f(x)$
-1.0	0.86199480
-0.5	0.95802009
0.0	1.0986123
0.5	1.2943767

Calculate the absolute error in using the interpolant to approximate $f(0.25)$ and $f'(0.25)$.

Example (2 of 4)

In this case $n = 3$ and

$$h_0 = h_1 = h_2 = 0.5$$

with

$$a_0 = 0.86199480, \; a_1 = 0.95802009,$$
$$a_2 = 1.0986123, \; a_3 = 1.2943767.$$

The linear system resembles,

$$A\mathbf{c} = \begin{bmatrix} 1.0 & 0.0 & 0.0 & 0.0 \\ 0.5 & 2.0 & 0.5 & 0.0 \\ 0.0 & 0.5 & 2.0 & 0.5 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 0.0 \\ 0.267402 \\ 0.331034 \\ 0.0 \end{bmatrix} = \mathbf{y}$$

Example (3 of 4)

The coefficients of the piecewise cubics:

i	a_i	b_i	c_i	d_i
0	0.861995	0.175638	0.0	0.0656509
1	0.95802	0.224876	0.0984763	0.028281
2	1.09861	0.344563	0.140898	−0.093918

Example (3 of 4)

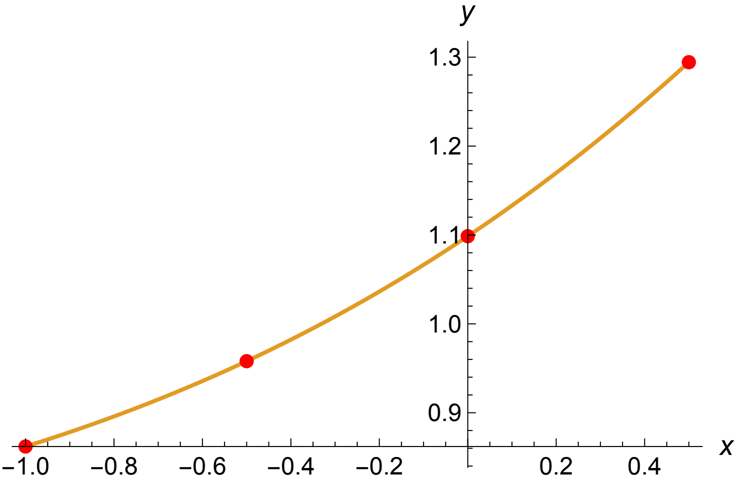
The coefficients of the piecewise cubics:

i	a_i	b_i	c_i	d_i
0	0.861995	0.175638	0.0	0.0656509
1	0.95802	0.224876	0.0984763	0.028281
2	1.09861	0.344563	0.140898	−0.093918

The cubic spline:

$$S(x) = \begin{cases} 0.861995 + 0.175638(x+1) & \text{if } -1 \leq x \leq -0.5 \\ \quad + 0.0656509(x+1)^3 & \\ 0.95802 + 0.224876(x+0.5) & \\ \quad + 0.0984763(x+0.5)^2 & \text{if } -0.5 \leq x \leq 0 \\ \quad + 0.028281(x+0.5)^3 & \\ 1.09861 + 0.344563x & \text{if } 0 \leq x \leq 0.5 \\ \quad + 0.140898x^2 - 0.093918x^3 & \end{cases}$$

Example (4 of 4)



$f(0.25)$	$S(0.25)$	Abs. Err.	$f'(0.25)$	$S'(0.25)$	Abs. Err.
1.18907	1.19209	3.02154×10^{-3}	0.390991	0.3974	6.40839×10^{-3}

Clamped Boundary Conditions (1 of 2)

If $S'(a) = S_0'(a) = f'(a) = b_0$ then

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

which is equivalent to

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

This replaces the first equation in our system of n equations.

Clamped Boundary Conditions (2 of 2)

Likewise if $S'(b) = S_n'(b) = f'(b) = b_n$ then

$$\begin{aligned} b_n &= b_{n-1} + h_{n-1}(c_{n-1} + c_n) \\ &= \frac{1}{h_{n-1}}(a_n - a_{n-1}) - \frac{h_{n-1}}{3}(2c_{n-1} + c_n) + h_{n-1}(c_{n-1} + c_n) \\ &= \frac{1}{h_{n-1}}(a_n - a_{n-1}) + \frac{h_{n-1}}{3}(c_{n-1} + 2c_n) \end{aligned}$$

which is equivalent to

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

This replaces the last equation in our system of n equations.

Clamped BC Linear System (1 of 2)

Theorem

If f is defined at $a = x_0 < x_1 < \cdots < x_n = b$ and differentiable at $x = a$ and at $x = b$, then f has a unique clamped cubic spline interpolant.

In matrix form the system of n equations has the form $A\mathbf{c} = \mathbf{y}$ where

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

Note: A is a tridiagonal matrix.

Clamped BC Linear System (2 of 2)

$$A \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_{n-2} \\ c_{n-1} \\ c_n \end{bmatrix} = \begin{bmatrix} \frac{3}{h_0}(a_1 - a_0) - 3f'(a) \\ \frac{3}{h_1}(a_2 - a_1) - \frac{3}{h_0}(a_1 - a_0) \\ \frac{3}{h_2}(a_3 - a_2) - \frac{3}{h_1}(a_2 - a_1) \\ \vdots \\ \frac{3}{h_{n-2}}(a_{n-1} - a_{n-2}) - \frac{3}{h_{n-3}}(a_{n-2} - a_{n-3}) \\ \frac{3}{h_{n-1}}(a_n - a_{n-1}) - \frac{3}{h_{n-2}}(a_{n-1} - a_{n-2}) \\ 3f'(b) - \frac{3}{h_{n-1}}(a_n - a_{n-1}) \end{bmatrix}$$

Coefficients of the Cubic Splines

Since a_j for $j = 0, 1, \ldots, n$ is known, once we solve the linear system for c_j (again for $j = 0, 1, \ldots, n$) then

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

$$d_j = \frac{1}{3h_j}(c_{j+1} - c_j)$$

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

Clamped Cubic Spline Algorithm (1 of 2)

INPUT $\{(x_0, f(x_0)), (x_1, f(x_1)), \ldots, (x_n, f(x_n))\}$, $f'(x_0)$, and $f'(x_n)$.

STEP 1 For $i = 0, 1, \ldots, n-1$ set $a_i = f(x_i)$; set $h_i = x_{i+1} - x_i$.

STEP 2 Set $\alpha_0 = \dfrac{3(a_1 - a_0)}{h_0} - 3f'(x_0)$;

$$\alpha_n = 3f'(x_n) - \frac{3(a_n - a_{n-1})}{h_{n-1}}.$$

STEP 3 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 4 Set $l_0 = 2h_0$; $\mu_0 = 0.5$; $z_0 = \dfrac{\alpha_0}{l_0}$.

Clamped Cubic Spline Algorithm (2 of 2)

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

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

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

STEP 8 For $j = 0, 1, \ldots, n-1$ OUTPUT a_j, b_j, c_j, d_j.

Example (1 of 4)

Construct the clamped cubic spline interpolant for $f(x) = \ln(e^x + 2)$ with nodal values:

x	$f(x)$
-1.0	0.86199480
-0.5	0.95802009
0.0	1.0986123
0.5	1.2943767

Calculate the absolute error in using the interpolant to approximate $f(0.25)$ and $f'(0.25)$.

Example (2 of 4)

In this case $n = 3$ and

$$h_0 = h_1 = h_2 = 0.5$$

with

$$a_0 = 0.86199480, \; a_1 = 0.95802009,$$
$$a_2 = 1.0986123, \; a_3 = 1.2943767.$$

Note that $f'(-1) \approx 0.155362$ and $f'(0.5) \approx 0.451863$.

The linear system resembles,

$$A\mathbf{c} = \begin{bmatrix} 1.0 & 0.5 & 0.0 & 0.0 \\ 0.5 & 2.0 & 0.5 & 0.0 \\ 0.0 & 0.5 & 2.0 & 0.5 \\ 0.0 & 0.0 & 0.5 & 1.0 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 0.110064 \\ 0.267402 \\ 0.331034 \\ 0.181001 \end{bmatrix} = \mathbf{y}.$$

Example (3 of 4)

The coefficients of the piecewise cubics:

i	a_i	b_i	c_i	d_i
0	0.861995	0.155362	0.0653748	0.0160031
1	0.95802	0.23274	0.0893795	0.0150207
2	1.09861	0.333384	0.11191	0.00875717

Example (3 of 4)

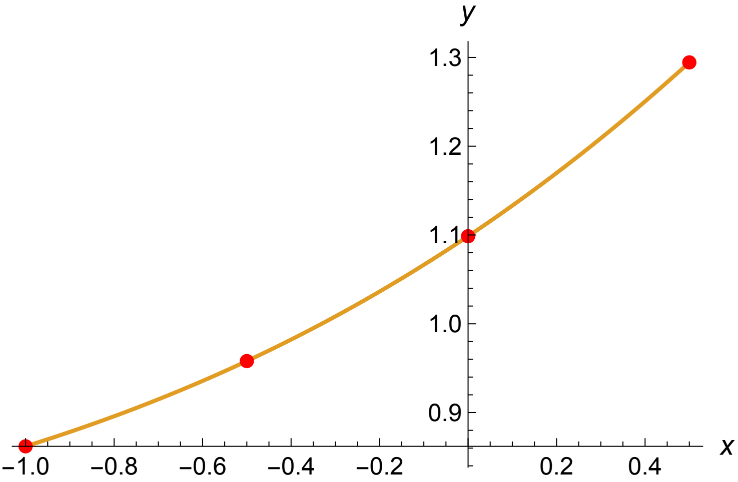
The coefficients of the piecewise cubics:

i	a_i	b_i	c_i	d_i
0	0.861995	0.155362	0.0653748	0.0160031
1	0.95802	0.23274	0.0893795	0.0150207
2	1.09861	0.333384	0.11191	0.00875717

The cubic spline:

$$S(x) = \begin{cases} 0.861995 + 0.155362(x+1) \\ \quad + 0.0653748(x+1)^2 & \text{if } -1 \leq x \leq -0.5 \\ \quad + 0.0160031(x+1)^3 \\ 0.95802 + 0.23274(x+0.5) \\ \quad + 0.0893795(x+0.5)^2 & \text{if } -0.5 \leq x \leq 0 \\ \quad + 0.0150207(x+0.5)^3 \\ 1.09861 + 0.333384x + 0.11191x^2 & \text{if } 0 \leq x \leq 0.5 \\ \quad + 0.00875717x^3 \end{cases}$$

Example (4 of 4)



$f(0.25)$	$S(0.25)$	Abs. Err.	$f'(0.25)$	$S'(0.25)$	Abs. Err.
1.18907	1.18991	1.97037×10^{-5}	0.390991	0.390982	9.67677×10^{-6}

Error Analysis

Theorem

Let $f \in \mathcal{C}^4[a,b]$ with $\max_{a \le x \le b} \left| f^{(4)}(x) \right| = M$. If S is the unique clamped cubic spline interpolant to f with respect to nodes $a = x_0 < x_1 < \cdots < x_n = b$, then for all $x \in [a,b]$,

$$|f(x) - S(x)| \le \frac{5M}{384} \max_{0 \le j \le n-1} (x_{j+1} - x_j)^4.$$

Example

Earlier we found the clamped cubic spline interpolant for $f(x) = \ln(e^x + 2)$. In this example $x_{j+1} - x_j = 0.5$ for all j.

Note that

$$\begin{aligned} f^{(4)}(x) &= \frac{2e^x(4 - 8e^x + e^{2x})}{(2 + e^x)^4} \\ \max_{-1 \leq x \leq 0.5} \left| f^{(4)}(x) \right| &\approx 0.120398 \\ |f(0.25) - S(0.25)| &= 1.97037 \times 10^{-5} \\ &\leq \frac{5(0.120398)}{384}(0.5)^4 \\ &\approx 9.798 \times 10^{-5}. \end{aligned}$$

Natural Cubic Spline Example (1 of 3)

Consider the following data:

x	$f(x)$
-0.5	-0.02475
-0.25	0.334938
0.0	1.101

The linear system takes the form

$$A\mathbf{c} = \mathbf{y}$$

$$\begin{bmatrix} 1.00 & 0.00 & 0.00 \\ 0.25 & 1.00 & 0.25 \\ 0.00 & 0.00 & 1.00 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} 0.00 \\ 4.8765 \\ 0.00 \end{bmatrix}$$

Natural Cubic Spline Example (2 of 3)

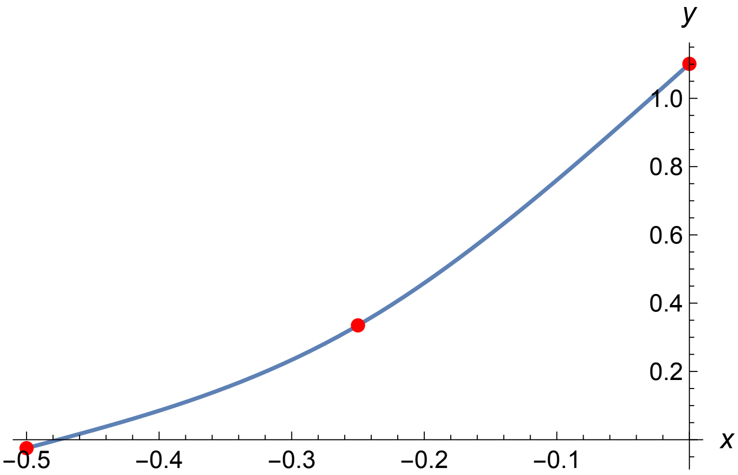
The coefficients of the natural cubic spline interpolant are

a_i	b_i	c_i	d_i
-0.02475	1.03238	0.0	6.502
0.334938	2.2515	4.8765	-6.502

and the piecewise cubic is

$$S(x) = \begin{cases} -0.02475 + 1.03238(x+0.5) + 6.502(x+0.05)^3 & \text{if } -0.5 \leq x \leq -0.25 \\ 0.334938 + 2.2515(x+0.25) + 4.8765(x+0.25)^2 - 6.502(x+0.25)^3 & \text{if } -0.25 \leq x \leq 0. \end{cases}$$

Natural Cubic Spline Example (3 of 3)



Clamped Cubic Spline Example (1 of 4)

Here we will find the clamped cubic spline interpolant to the function $f(x) = J_0(\sqrt{x})$ at the nodes $x_i = 5i$ for $i = 0, 1, \ldots, 10$.

x	$f(x)$
0.0	1.0
5.0	0.0904053
10.0	-0.310045
$\vdots$	$\vdots$
50.0	0.299655

Note: $f'(0) = -0.25$ and $f'(50) = -0.00117217$.

Clamped Cubic Spline Example (2 of 4)

The tridiagonal linear system takes the following form

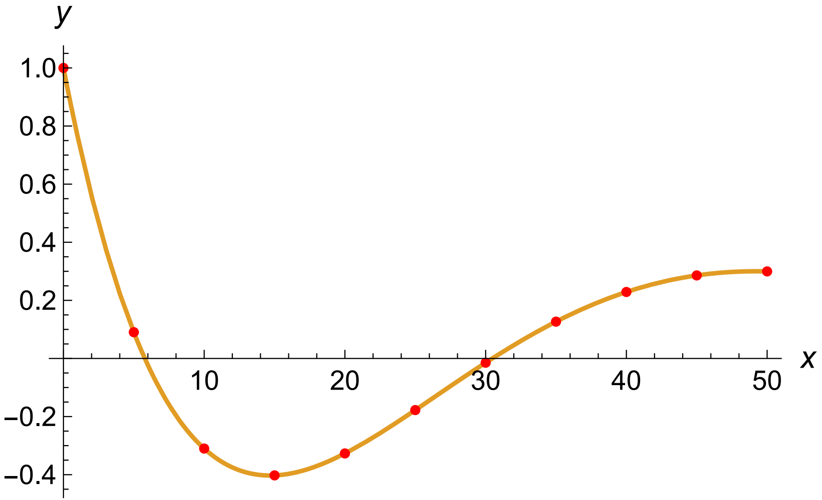
$$\begin{bmatrix} 10 & 5 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 5 & 20 & 5 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 5 & 20 & 5 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & & & & \ddots & & & & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 5 & 20 & 5 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 5 & 20 & 5 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 5 & 10 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_8 \\ c_9 \\ c_{10} \end{bmatrix} = \begin{bmatrix} 0.204243 \\ 0.305487 \\ 0.184846 \\ 0.100749 \\ 0.044242 \\ 0.008211 \\ -0.012944 \\ -0.023582 \\ -0.027056 \\ -0.025905 \\ -0.011808 \end{bmatrix}.$$

Clamped Cubic Spline Example (3 of 4)

The coefficients of the clamped cubic spline interpolant are

a_i	b_i	c_i	d_i
1	-0.25	0.0154655	-0.00036986
0.09040533	-0.1230843	0.009917643	-0.0002637577
-0.3100448	-0.0436897	0.005961278	-0.0001836499
-0.4024176	0.00214934	0.003206529	-0.0001229411
-0.3268753	0.02499404	0.001362412	-0.0000780158
-0.1775968	0.03276697	0.000192174	-0.0000454083
-0.0146336	0.03128308	-0.00048895	-0.0000224102
0.12675676	0.02471281	-0.00082510	-6.79522×10^{-6}
0.22884382	0.01595213	-0.00092703	3.265389×10^{-6}
0.28583684	0.00692671	-0.00087805	9.088463×10^{-6}

Clamped Cubic Spline Example (4 of 4)



Homework

- Read Section 3.5
- Exercises: 1, 3d, 5d, 7d, 25