Interpolation & Polynomial Approximation

Cubic Spline Interpolation I

Numerical Analysis (9th Edition)

R L Burden & J D Faires

Beamer Presentation Slides
prepared by
John Carroll
Dublin City University

© 2011 Brooks/Cole, Cengage Learning

Outline

Outline

1. Piecewise-Polynomial Approximation

2. Conditions for a Cubic Spline Interpolant

Outline

Outline

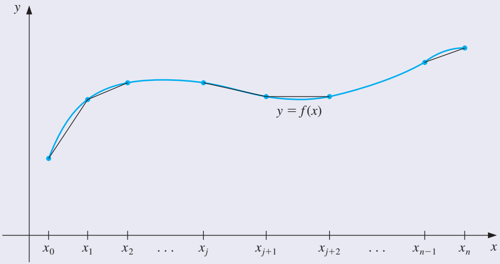
Piecewise-Polynomial Approximation

Piecewise-linear interpolation

This is the simplest piecewise-polynomial approximation and which consists of joining a set of data points

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

by a series of straight lines:



Piecewise-Polynomial Approximation

Disadvantage of piecewise-linear interpolation

Piecewise-Polynomial Approximation

Disadvantage of piecewise-linear interpolation

- There is likely no differentiability at the endpoints of the subintervals, which, in a geometrical context, means that the interpolating function is not "smooth."

Piecewise-Polynomial Approximation

Disadvantage of piecewise-linear interpolation

- There is likely no differentiability at the endpoints of the subintervals, which, in a geometrical context, means that the interpolating function is not “smooth.”
- Often it is clear from physical conditions that smoothness is required, so the approximating function must be continuously differentiable.

Piecewise-Polynomial Approximation

Disadvantage of piecewise-linear interpolation

- There is likely no differentiability at the endpoints of the subintervals, which, in a geometrical context, means that the interpolating function is not "smooth."
- Often it is clear from physical conditions that smoothness is required, so the approximating function must be continuously differentiable.
- We will next consider approximation using piecewise polynomials that require no specific derivative information, except perhaps at the endpoints of the interval on which the function is being approximated.

Piecewise-Polynomial Approximation

Differentiable piecewise-polynomial function

Piecewise-Polynomial Approximation

Differentiable piecewise-polynomial function

- The simplest type of differentiable piecewise-polynomial function on an entire interval $[x_0, x_n]$ is the function obtained by fitting one quadratic polynomial between each successive pair of nodes.

Piecewise-Polynomial Approximation

Differentiable piecewise-polynomial function

- The simplest type of differentiable piecewise-polynomial function on an entire interval $[x_0, x_n]$ is the function obtained by fitting one quadratic polynomial between each successive pair of nodes.
- This is done by constructing a quadratic on

$$[x_0, x_1] \text{ agreeing with the function at } x_0 \text{ and } x_1,$$

and another quadratic on

$$[x_1, x_2] \text{ agreeing with the function at } x_1 \text{ and } x_2,$$

and so on.

Piecewise-Polynomial Approximation

Differentiable piecewise-polynomial function (Cont'd)

- A general quadratic polynomial has 3 arbitrary constants—the constant term, the coefficient of x, and the coefficient of x^2—and only 2 conditions are required to fit the data at the endpoints of each subinterval.

Piecewise-Polynomial Approximation

Differentiable piecewise-polynomial function (Cont'd)

- A general quadratic polynomial has 3 arbitrary constants—the constant term, the coefficient of x, and the coefficient of x^2—and only 2 conditions are required to fit the data at the endpoints of each subinterval.
- So flexibility exists that permits the quadratics to be chosen so that the interpolant has a continuous derivative on $[x_0, x_n]$.

Piecewise-Polynomial Approximation

Differentiable piecewise-polynomial function (Cont'd)

- A general quadratic polynomial has 3 arbitrary constants—the constant term, the coefficient of x, and the coefficient of x^2—and only 2 conditions are required to fit the data at the endpoints of each subinterval.
- So flexibility exists that permits the quadratics to be chosen so that the interpolant has a continuous derivative on $[x_0, x_n]$.
- The difficulty arises because we generally need to specify conditions about the derivative of the interpolant at the endpoints x_0 and x_n.

Piecewise-Polynomial Approximation

Differentiable piecewise-polynomial function (Cont'd)

- A general quadratic polynomial has 3 arbitrary constants—the constant term, the coefficient of x, and the coefficient of x^2—and only 2 conditions are required to fit the data at the endpoints of each subinterval.
- So flexibility exists that permits the quadratics to be chosen so that the interpolant has a continuous derivative on $[x_0, x_n]$.
- The difficulty arises because we generally need to specify conditions about the derivative of the interpolant at the endpoints x_0 and x_n.
- There is an insufficient number of constants to ensure that the conditions will be satisfied.

Outline

1 Piecewise-Polynomial Approximation

2 Conditions for a Cubic Spline Interpolant

3 Construction of a Cubic Spline

Cubic Splines: Establishing Conditions

Most common piecewise-polynomial approximation

- The most common piecewise-polynomial approximation uses cubic polynomials between each successive pair of nodes and is called cubic spline interpolation. ▸ Meaning of Spline

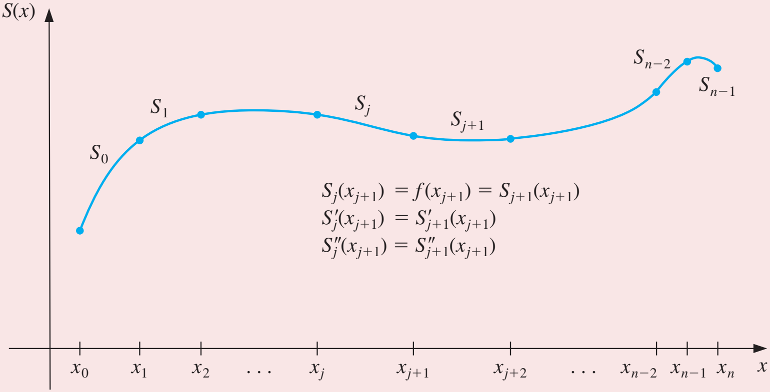
Cubic Splines: Establishing Conditions

Most common piecewise-polynomial approximation

- The most common piecewise-polynomial approximation uses cubic polynomials between each successive pair of nodes and is called cubic spline interpolation. ▸ Meaning of Spline
- A general cubic polynomial involves 4 constants, so there is sufficient flexibility in the cubic spline procedure to ensure that the interpolant is not only continuously differentiable on the interval, but also has a continuous second derivative.

Cubic Splines: Establishing Conditions

The construction of the cubic spline does not, however, assume that the derivatives of the interpolant agree with those of the function it is approximating, even at the nodes.



Cubic Spline Interpolant

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

Cubic Spline Interpolant

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

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

Cubic Spline Interpolant

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

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

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

Cubic Spline Interpolant

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

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

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

(c) $S_{j+1}(x_{j+1}) = S_j(x_{j+1})$ for each $j = 0, 1, \ldots, n-2$; (*Implied by* (b).)

Cubic Spline Interpolant

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

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

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

(c) $S_{j+1}(x_{j+1}) = S_j(x_{j+1})$ for each $j = 0, 1, \ldots, n-2$; (*Implied by* (b).)

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

Cubic Spline Interpolant

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

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

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

(c) $S_{j+1}(x_{j+1}) = S_j(x_{j+1})$ for each $j = 0, 1, \ldots, n-2$; (*Implied by* (b).)

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

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

Cubic Spline Interpolant

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

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

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

(c) $S_{j+1}(x_{j+1}) = S_j(x_{j+1})$ for each $j = 0, 1, \ldots, n-2$; (*Implied by* (b).)

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

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

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

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

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

Cubic Splines: Natural & Clamped Conditions

Natural & Clamped Boundary Conditions

Cubic Splines: Natural & Clamped Conditions

Natural & Clamped Boundary Conditions

- Although cubic splines are defined with other boundary conditions, the conditions given in (f) are sufficient for our purposes.

Cubic Splines: Natural & Clamped Conditions

Natural & Clamped Boundary Conditions

- Although cubic splines are defined with other boundary conditions, the conditions given in (f) are sufficient for our purposes.
- When the free boundary conditions occur, the spline is called a natural spline, and its graph approximates the shape that a long flexible rod would assume if forced to go through the data points $\{(x_0, f(x_0)), (x_1, f(x_1)), \ldots, (x_n, f(x_n))\}$. See Natural Spline

Cubic Splines: Natural & Clamped Conditions

Natural & Clamped Boundary Conditions

- Although cubic splines are defined with other boundary conditions, the conditions given in (f) are sufficient for our purposes.
- When the free boundary conditions occur, the spline is called a natural spline, and its graph approximates the shape that a long flexible rod would assume if forced to go through the data points $\{(x_0, f(x_0)), (x_1, f(x_1)), \ldots, (x_n, f(x_n))\}$. See Natural Spline
- In general, clamped boundary conditions lead to more accurate approximations because they include more information about the function.

Cubic Splines: Natural & Clamped Conditions

Natural & Clamped Boundary Conditions

- Although cubic splines are defined with other boundary conditions, the conditions given in (f) are sufficient for our purposes.
- When the free boundary conditions occur, the spline is called a natural spline, and its graph approximates the shape that a long flexible rod would assume if forced to go through the data points $\{(x_0, f(x_0)), (x_1, f(x_1)), \ldots, (x_n, f(x_n))\}$. See Natural Spline
- In general, clamped boundary conditions lead to more accurate approximations because they include more information about the function.
- However, for this type of boundary condition to hold, it is necessary to have either the values of the derivative at the endpoints or an accurate approximation to those values.

Cubic Splines: Establishing Conditions

Example: 3 Data Values

Construct a natural cubic spline that passes through the points $(1, 2)$, $(2, 3)$, and $(3, 5)$.

Cubic Splines: Establishing Conditions

Example: 3 Data Values

Construct a natural cubic spline that passes through the points $(1, 2)$, $(2, 3)$, and $(3, 5)$.

Solution (1/4)

This spline consists of two cubics:

Cubic Splines: Establishing Conditions

Example: 3 Data Values

Construct a natural cubic spline that passes through the points $(1, 2)$, $(2, 3)$, and $(3, 5)$.

Solution (1/4)

This spline consists of two cubics: the first for the interval $[1, 2]$, denoted

$$S_0(x) = a_0 + b_0(x - 1) + c_0(x - 1)^2 + d_0(x - 1)^3,$$

Cubic Splines: Establishing Conditions

Example: 3 Data Values

Construct a natural cubic spline that passes through the points $(1, 2)$, $(2, 3)$, and $(3, 5)$.

Solution (1/4)

This spline consists of two cubics: the first for the interval $[1, 2]$, denoted

$$S_0(x) = a_0 + b_0(x - 1) + c_0(x - 1)^2 + d_0(x - 1)^3,$$

and the other for $[2, 3]$, denoted

$$S_1(x) = a_1 + b_1(x - 2) + c_1(x - 2)^2 + d_1(x - 2)^3.$$

Cubic Splines: Example with 3 Data Values

Solution (2/4)

There are 8 constants to be determined, which requires 8 conditions.

Cubic Splines: Example with 3 Data Values

Solution (2/4)

There are 8 constants to be determined, which requires 8 conditions. 4 conditions come from the fact that the splines must agree with the data at the nodes.

Cubic Splines: Example with 3 Data Values

Solution (2/4)

There are 8 constants to be determined, which requires 8 conditions. 4 conditions come from the fact that the splines must agree with the data at the nodes. Hence

$$2 = f(1) = a_0, \quad 3 = f(2) = a_0 + b_0 + c_0 + d_0, \quad 3 = f(2) = a_1$$
$$\text{and} \quad 5 = f(3) = a_1 + b_1 + c_1 + d_1$$

Cubic Splines: Example with 3 Data Values

Solution (2/4)

There are 8 constants to be determined, which requires 8 conditions. 4 conditions come from the fact that the splines must agree with the data at the nodes. Hence

$$2 = f(1) = a_0, \quad 3 = f(2) = a_0 + b_0 + c_0 + d_0, \quad 3 = f(2) = a_1$$
$$\text{and} \quad 5 = f(3) = a_1 + b_1 + c_1 + d_1$$

2 more come from the fact that $S_0'(2) = S_1'(2)$ and $S_0''(2) = S_1''(2)$.

Cubic Splines: Example with 3 Data Values

Solution (2/4)

There are 8 constants to be determined, which requires 8 conditions. 4 conditions come from the fact that the splines must agree with the data at the nodes. Hence

$$2 = f(1) = a_0, \quad 3 = f(2) = a_0 + b_0 + c_0 + d_0, \quad 3 = f(2) = a_1$$
$$\text{and} \quad 5 = f(3) = a_1 + b_1 + c_1 + d_1$$

2 more come from the fact that $S_0'(2) = S_1'(2)$ and $S_0''(2) = S_1''(2)$. These are

$$S_0'(2) = S_1'(2): \quad b_0 + 2c_0 + 3d_0 = b_1$$
$$\text{and} \quad S_0''(2) = S_1''(2): \quad 2c_0 + 6d_0 = 2c_1$$

Cubic Splines: Example with 3 Data Values

(1)	$2 = a_0$	(2)	$3 = a_0 + b_0 + c_0 + d_0$
(3)	$3 = a_1$	(4)	$5 = a_1 + b_1 + c_1 + d_1$
(5)	$b_0 + 2c_0 + 3d_0 = b_1$	(6)	$2c_0 + 6d_0 = 2c_1$

Cubic Splines: Example with 3 Data Values

(1)	$2 = a_0$	(2)	$3 = a_0 + b_0 + c_0 + d_0$
(3)	$3 = a_1$	(4)	$5 = a_1 + b_1 + c_1 + d_1$
(5)	$b_0 + 2c_0 + 3d_0 = b_1$	(6)	$2c_0 + 6d_0 = 2c_1$

Solution (3/4)

The final 2 come from the natural boundary conditions:

$$S_0''(1) = 0: \quad 2c_0 = 0 \qquad \text{and} \qquad S_1''(3) = 0: \quad 2c_1 + 6d_1 = 0.$$

Cubic Splines: Example with 3 Data Values

(1)	$2 = a_0$	(2)	$3 = a_0 + b_0 + c_0 + d_0$
(3)	$3 = a_1$	(4)	$5 = a_1 + b_1 + c_1 + d_1$
(5)	$b_0 + 2c_0 + 3d_0 = b_1$	(6)	$2c_0 + 6d_0 = 2c_1$
(7)	$2c_0 = 0$	(8)	$2c_1 + 6d_1 = 0$

Cubic Splines: Example with 3 Data Values

(1)	$2 = a_0$	(2)	$3 = a_0 + b_0 + c_0 + d_0$
(3)	$3 = a_1$	(4)	$5 = a_1 + b_1 + c_1 + d_1$
(5)	$b_0 + 2c_0 + 3d_0 = b_1$	(6)	$2c_0 + 6d_0 = 2c_1$
(7)	$2c_0 = 0$	(8)	$2c_1 + 6d_1 = 0$

Solution (4/4)

Solving this system of equations gives the spline

$$S(x) = \begin{cases} 2 + \frac{3}{4}(x-1) + \frac{1}{4}(x-1)^3, & \text{for } x \in [1,2] \\ 3 + \frac{3}{2}(x-2) + \frac{3}{4}(x-2)^2 - \frac{1}{4}(x-2)^3, & \text{for } x \in [2,3] \end{cases}$$

Outline

Basic Approach

Basic Approach

- A spline defined on an interval that is divided into n subintervals will require determining $4n$ constants.

Basic Approach

- 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

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

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

Basic Approach

- 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

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

for each $j = 0, 1, \ldots, n-1$. Since $S_j(x_j) = a_j = f(x_j)$, condition (c), namely $S_{j+1}(x_{j+1}) = S_j(x_{j+1})$, can be applied to obtain

$$\begin{aligned} a_{j+1} &= S_{j+1}(x_{j+1}) = S_j(x_{j+1}) \\ &= a_j + b_j(x_{j+1} - x_j) + c_j(x_{j+1} - x_j)^2 + d_j(x_{j+1} - x_j)^3 \end{aligned}$$

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

Cubic Splines: Construction

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

Basic Approach (Cont'd)

Cubic Splines: Construction

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

Basic Approach (Cont'd)

The terms $x_{j+1} - x_j$ are used repeatedly in this development, so it is convenient to introduce the simpler notation

$$h_j = x_{j+1} - x_j,$$

for each $j = 0, 1, \ldots, n-1$. If we also define $a_n = f(x_n)$,

Cubic Splines: Construction

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

Basic Approach (Cont'd)

The terms $x_{j+1} - x_j$ are used repeatedly in this development, so it is convenient to introduce the simpler notation

$$h_j = x_{j+1} - x_j,$$

for each $j = 0, 1, \ldots, n-1$. If we also define $a_n = f(x_n)$, then the equation

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

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

Cubic Splines: Construction

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

Basic Approach (Cont'd)

Cubic Splines: Construction

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

Basic Approach (Cont'd)

In a similar manner, 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 $S_j'(x_j) = b_j$, for each $j = 0, 1, \ldots, n-1$.

Cubic Splines: Construction

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

Basic Approach (Cont'd)

In a similar manner, 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 $S_j'(x_j) = b_j$, for each $j = 0, 1, \ldots, n-1$. Applying condition (d), namely $S_{j+1}'(x_{j+1}) = S_j'(x_{j+1})$, gives

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

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

Cubic Splines: Construction

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

Basic Approach (Cont'd)

Cubic Splines: Construction

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

Basic Approach (Cont'd)

Another relationship between the coefficients of S_j is obtained by defining $c_n = S''(x_n)/2$ and applying condition (e), namely $S''_{j+1}(x_{j+1}) = S''_j(x_{j+1})$.

Cubic Splines: Construction

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

Basic Approach (Cont'd)

Another relationship between the coefficients of S_j is obtained by defining $c_n = S''(x_n)/2$ and applying condition (e), namely $S''_{j+1}(x_{j+1}) = S''_j(x_{j+1})$. Then, for each $j = 0, 1, \ldots, n-1$,

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

Cubic Splines: Construction

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

Basic Approach (Cont'd)

Cubic Splines: Construction

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

Basic Approach (Cont'd)

Solving for d_j in the third equation and substituting this value into the other two gives, for each $j = 0, 1, \ldots, n-1$, the new equations

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

Cubic Splines: Construction

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

Basic Approach (Cont'd)

Cubic Splines: Construction

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

Basic Approach (Cont'd)

The final relationship involving the coefficients is obtained by solving the appropriate equation in the form of the equation for a_{j+1} above, first for b_j:

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

and then, with a reduction of the index, for b_{j-1}:

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

Cubic Splines: Construction

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

Basic Approach (Cont'd)

Cubic Splines: Construction

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

Basic Approach (Cont'd)

Substituting these values into the equation derived from

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

with the index reduced by one, gives the linear system of equations

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

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

Cubic Splines: Construction

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

Basic Approach (Cont'd)

Cubic Splines: Construction

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

Basic Approach (Cont'd)

- This system involves only the $\{c_j\}_{j=0}^{n}$ as unknowns.

Cubic Splines: Construction

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

Basic Approach (Cont'd)

- This system involves only the $\{c_j\}_{j=0}^{n}$ as unknowns.
- The values of $\{h_j\}_{j=0}^{n-1}$ and $\{a_j\}_{j=0}^{n}$ are given, respectively, by the spacing of the nodes $\{x_j\}_{j=0}^{n}$ and the values of f at the nodes.

Cubic Splines: Construction

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

Basic Approach (Cont'd)

- This system involves only the $\{c_j\}_{j=0}^{n}$ as unknowns.
- The values of $\{h_j\}_{j=0}^{n-1}$ and $\{a_j\}_{j=0}^{n}$ are given, respectively, by the spacing of the nodes $\{x_j\}_{j=0}^{n}$ and the values of f at the nodes.
- So once the values of $\{c_j\}_{j=0}^{n}$ are determined, it is a simple matter to find the remainder of the constants

  $\{b_j\}_{j=0}^{n-1}$ from $b_j = \frac{1}{h_j}(a_{j+1} - a_j) - \frac{h_j}{3}(2c_j + c_{j+1})$ and

  $\{d_j\}_{j=0}^{n-1}$ from $c_{j+1} = c_j + 3d_j h_j$

Cubic Splines: Construction

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

Basic Approach (Cont'd)

- This system involves only the $\{c_j\}_{j=0}^{n}$ as unknowns.
- The values of $\{h_j\}_{j=0}^{n-1}$ and $\{a_j\}_{j=0}^{n}$ are given, respectively, by the spacing of the nodes $\{x_j\}_{j=0}^{n}$ and the values of f at the nodes.
- So once the values of $\{c_j\}_{j=0}^{n}$ are determined, it is a simple matter to find the remainder of the constants

  $\{b_j\}_{j=0}^{n-1}$ from $b_j = \dfrac{1}{h_j}(a_{j+1} - a_j) - \dfrac{h_j}{3}(2c_j + c_{j+1})$ and

  $\{d_j\}_{j=0}^{n-1}$ from $c_{j+1} = c_j + 3d_j h_j$

  Then we can construct the cubic polynomials $\{S_j(x)\}_{j=0}^{n-1}$.

Cubic Splines: Construction

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

Major Question

Cubic Splines: Construction

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

Major Question

- The major question that arises in connection with this construction is whether the values of $\{c_j\}_{j=0}^{n}$ can be found using the system of equations given above and, if so, whether these values are unique.

Cubic Splines: Construction

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

Major Question

- The major question that arises in connection with this construction is whether the values of $\{c_j\}_{j=0}^{n}$ can be found using the system of equations given above and, if so, whether these values are unique.
- We will answer this question using theorems which indicate that this is the case when either of the boundary conditions given in part (f) of the Definition are imposed.

Questions?

Reference Material

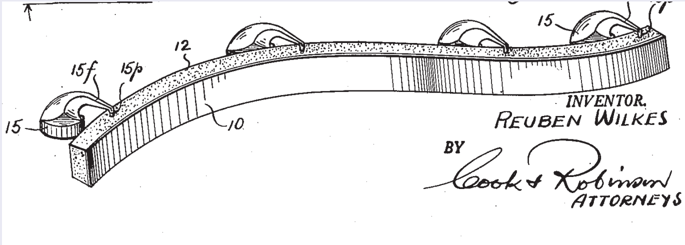
Spline

- The root of the word “spline” is the same as that of splint.
- It was originally a small strip of wood that could be used to join two boards.
- Later, the word was use to refer to a long flexible strip, generally of metal, that could be used to draw continuous smooth curves by forcing the strip to pass through specified points and tracing along the curve.

◂ Return to Cubic Spline Conditions

Natural Spline

- A natural spline has no conditions imposed for the direction at its endpoints, so the curve takes the shape of a straight line after it passes through the interpolation points nearest its endpoints.



- The name derives from the fact that this is the natural shape a flexible strip assumes if forced to pass through specified interpolation points with no additional constraints.

Cubic Spline Interpolant

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

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

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

(c) $S_{j+1}(x_{j+1}) = S_j(x_{j+1})$ for each $j = 0, 1, \ldots, n-2$; (*Implied by* (b).)

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

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

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

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

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

Return to Cubic Spline Construction: Basic Approach

Return to Cubic Spline Construction: Major Question