Integrating and collecting values from like powers gives

$$\int_0^3 S(x) = \left[x + 1.46600\frac{x^2}{2} + 0.25228\frac{x^4}{4}\right]_0^1$$

$$+ \left[2.71828(x-1) + 2.22285\frac{(x-1)^2}{2} + 0.75685\frac{(x-1)^3}{3} + 1.69107\frac{(x-1)^4}{4}\right]_1^2$$

$$+ \left[7.38906(x-2) + 8.80977\frac{(x-2)^2}{2} + 5.83007\frac{(x-2)^3}{3} - 1.94336\frac{(x-2)^4}{4}\right]_2^3$$

$$= (1 + 2.71828 + 7.38906) + \frac{1}{2}(1.46600 + 2.22285 + 8.80977)$$

$$+ \frac{1}{3}(0.75685 + 5.83007) + \frac{1}{4}(0.25228 + 1.69107 - 1.94336)$$

$$= 19.55229.$$

Because the nodes are equally spaced in this example the integral approximation is simply

$$\int_0^3 S(x)\,dx = (a_0+a_1+a_2)+\frac{1}{2}(b_0+b_1+b_2)+\frac{1}{3}(c_0+c_1+c_2)+\frac{1}{4}(d_0+d_1+d_2). \quad (3.22)$$

□

If we create the natural spline using Maple as described after Example 2, we can then use Maple's integration command to find the value in the Illustration. Simply enter

$int(sn(t), t = 0 .. 3)$

19.55228648

Clamped Splines

Example 3 In Example 1 we found a natural spline S that passes through the points $(1,2)$, $(2,3)$, and $(3,5)$. Construct a clamped spline s through these points that has $s'(1) = 2$ and $s'(3) = 1$.

Solution Let

$$s_0(x) = a_0 + b_0(x-1) + c_0(x-1)^2 + d_0(x-1)^3,$$

be the cubic on $[1,2]$ and the cubic on $[2,3]$ be

$$s_1(x) = a_1 + b_1(x-2) + c_1(x-2)^2 + d_1(x-2)^3.$$

Then most of the conditions to determine the 8 constants are the same as those in Example 1. That is,

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

$$5 = f(3) = a_1 + b_1 + c_1 + d_1.$$

$$s_0'(2) = s_1'(2): \quad b_0 + 2c_0 + 3d_0 = b_1 \quad \text{and} \quad s_0''(2) = s_1''(2): \quad 2c_0 + 6d_0 = 2c_1$$

However, the boundary conditions are now

$$s_0'(1) = 2: \quad b_0 = 2 \quad \text{and} \quad s_1'(3) = 1: \quad b_1 + 2c_1 + 3d_1 = 1.$$

Solving this system of equations gives the spline as

$$s(x) = \begin{cases} 2 + 2(x-1) - \frac{5}{2}(x-1)^2 + \frac{3}{2}(x-1)^3, & \text{for } x \in [1,2] \\ 3 + \frac{3}{2}(x-2) + 2(x-2)^2 - \frac{3}{2}(x-2)^3, & \text{for } x \in [2,3] \end{cases}$$

■

In the case of general clamped boundary conditions we have a result that is similar to the theorem for natural boundary conditions described in Theorem 3.11.

Theorem 3.12 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 is, a spline interpolant that satisfies the clamped boundary conditions $S'(a) = f'(a)$ and $S'(b) = f'(b)$. ■

Proof Since $f'(a) = S'(a) = S'(x_0) = b_0$, Eq. (3.20) 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. (3.20) 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

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

Equations (3.21) 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 $A\mathbf{x} = \mathbf{b}$, where

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

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

This matrix A is also strictly diagonally dominant, so it satisfies the conditions of Theorem 6.21 in Section 6.6. Therefore, the linear system has a unique solution for $c_0, c_1, \ldots, c_n$. ■ ■ ■

The solution to the cubic spline problem with the boundary conditions $S'(x_0) = f'(x_0)$ and $S'(x_n) = f'(x_n)$ can be obtained by applying Algorithm 3.5.

ALGORITHM 3.5

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; $x_0, x_1, \ldots, x_n$; $a_0 = f(x_0)$, $a_1 = f(x_1), \ldots, a_n = f(x_n)$; $FPO = f'(x_0)$; $FPN = f'(x_n)$.

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

(Note: $S(x) = S_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3$ for $x_j \le x \le x_{j+1}$.)

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 $l_i = 2(x_{i+1} - x_{i-1}) - h_{i-1}\mu_{i-1}$;
$\mu_i = h_i/l_i$;
$z_i = (\alpha_i - h_{i-1}z_{i-1})/l_i$.

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.

■

Example 4 Example 2 used a natural spline and the data points $(0, 1)$, $(1, e)$, $(2, e^2)$, and $(3, e^3)$ to form a new approximating function $S(x)$. Determine the clamped spline $s(x)$ that uses this data and the additional information that, since $f'(x) = e^x$, so $f'(0) = 1$ and $f'(3) = e^3$.

Solution As in Example 2, we have $n = 3$, $h_0 = h_1 = h_2 = 1$, $a_0 = 0$, $a_1 = e$, $a_2 = e^2$, and $a_3 = e^3$. This together with the information that $f'(0) = 1$ and $f'(3) = e^3$ gives the the matrix A and the vectors **b** and **x** with the forms

$$A = \begin{bmatrix} 2 & 1 & 0 & 0 \\ 1 & 4 & 1 & 0 \\ 0 & 1 & 4 & 1 \\ 0 & 0 & 1 & 2 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 3(e-2) \\ 3(e^2 - 2e + 1) \\ 3(e^3 - 2e^2 + e) \\ 3e^2 \end{bmatrix}, \quad \text{and} \quad \mathbf{x} = \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}.$$

The vector-matrix equation $A\mathbf{x} = \mathbf{b}$ is equivalent to the system of equations

$$2c_0 + c_1 = 3(e-2),$$

$$c_0 + 4c_1 + c_2 = 3(e^2 - 2e + 1),$$

$$c_1 + 4c_2 + c_3 = 3(e^3 - 2e^2 + e),$$

$$c_2 + 2c_3 = 3e^2.$$

Solving this system simultaneously for c_0, c_1, c_2 and c_3 gives, to 5 decimal places,

$$c_0 = \frac{1}{15}(2e^3 - 12e^2 + 42e - 59) = 0.44468,$$

$$c_1 = \frac{1}{15}(-4e^3 + 24e^2 - 39e + 28) = 1.26548,$$

$$c_2 = \frac{1}{15}(14e^3 - 39e^2 + 24e - 8) = 3.35087,$$

$$c_3 = \frac{1}{15}(-7e^3 + 42e^2 - 12e + 4) = 9.40815.$$

Solving for the remaining constants in the same manner as Example 2 gives

$$b_0 = 1.00000, \quad b_1 = 2.71016, \quad b_2 = 7.32652,$$

and

$$d_0 = 0.27360, \quad d_1 = 0.69513, \quad d_2 = 2.01909.$$

This gives the clamped cubic spine

$$s(x) = \begin{cases} 1 + x + 0.44468x^2 + 0.27360x^3, & \text{if } 0 \le x < 1, \\ 2.71828 + 2.71016(x-1) + 1.26548(x-1)^2 + 0.69513(x-1)^3, & \text{if } 1 \le x < 2, \\ 7.38906 + 7.32652(x-2) + 3.35087(x-2)^2 + 2.01909(x-2)^3, & \text{if } 2 \le x \le 3. \end{cases}$$

The graph of the clamped spline and $f(x) = e^x$ are so similar that no difference can be seen. ■

We can create the clamped cubic spline in Example 4 with the same commands we used for the natural spline, the only change that is needed is to specify the derivative at the endpoints. In this case we use

$sn := t \to Spline\,([[0., 1.0], [1.0, f(1.0)], [2.0, f(2.0)], [3.0, f(3.0)]], t, degree = 3,$
$endpoints = \left[1.0, e^{3.0}\right])$

giving essentially the same results as in the example.

We can also approximate the integral of f on $[0, 3]$, by integrating the clamped spline. The exact value of the integral is

$$\int_0^3 e^x\,dx = e^3 - 1 \approx 20.08554 - 1 = 19.08554.$$

Because the data is equally spaced, piecewise integrating the clamped spline results in the same formula as in (3.22), that is,

$$\int_0^3 s(x)\,dx = (a_0 + a_1 + a_2) + \frac{1}{2}(b_0 + b_1 + b_2) + \frac{1}{3}(c_0 + c_1 + c_2) + \frac{1}{4}(d_0 + d_1 + d_2).$$

Hence the integral approximation is

$$\int_0^3 s(x)\,dx = (1 + 2.71828 + 7.38906) + \frac{1}{2}(1 + 2.71016 + 7.32652) + \frac{1}{3}(0.44468 + 1.26548 + 3.35087) + \frac{1}{4}(0.27360 + 0.69513 + 2.01909) = 19.05965.$$

The absolute error in the integral approximation using the clamped and natural splines are

$$\text{Natural} : |19.08554 - 19.55229| = 0.46675$$

and

$$\text{Clamped} : |19.08554 - 19.05965| = 0.02589.$$

For integration purposes the clamped spline is vastly superior. This should be no surprise since the boundary conditions for the clamped spline are exact, whereas for the natural spline we are essentially assuming that, since $f''(x) = e^x$,

$$0 = S''(0) \approx f''(0) = e^1 = 1 \quad \text{and} \quad 0 = S''(3) \approx f''(3) = e^3 \approx 20.$$

The next illustration uses a spine to approximate a curve that has no given functional representation.

Illustration Figure 3.11 shows a ruddy duck in flight. To approximate the top profile of the duck, we have chosen points along the curve through which we want the approximating curve to pass. Table 3.18 lists the coordinates of 21 data points relative to the superimposed coordinate system shown in Figure 3.12. Notice that more points are used when the curve is changing rapidly than when it is changing more slowly.

Figure 3.11

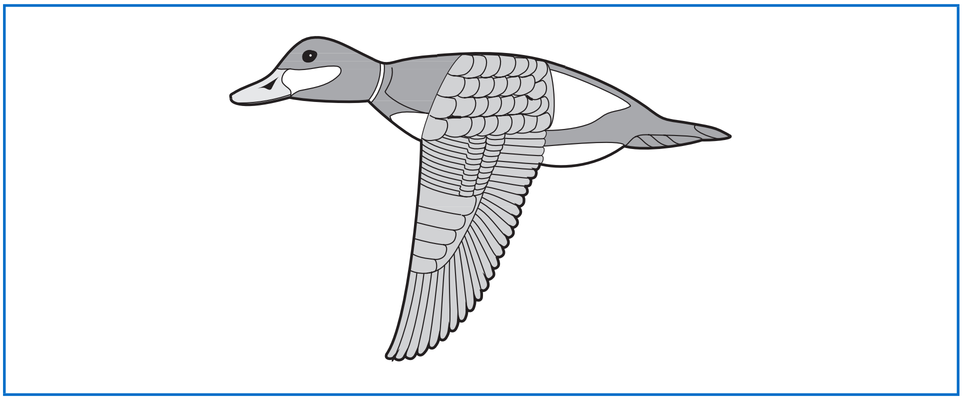
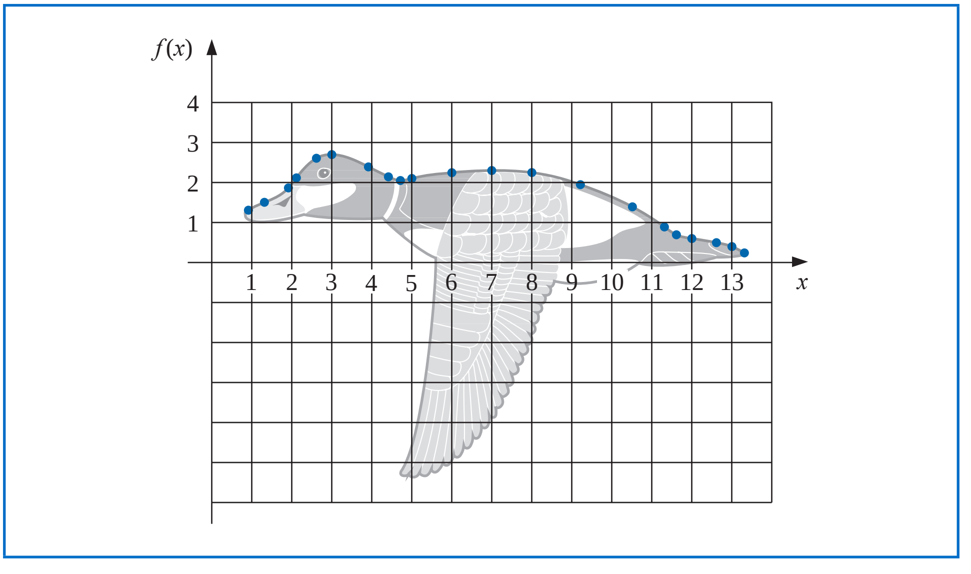


Table 3.18

x	0.9	1.3	1.9	2.1	2.6	3.0	3.9	4.4	4.7	5.0	6.0	7.0	8.0	9.2	10.5	11.3	11.6	12.0	12.6	13.0	13.3
$f(x)$	1.3	1.5	1.85	2.1	2.6	2.7	2.4	2.15	2.05	2.1	2.25	2.3	2.25	1.95	1.4	0.9	0.7	0.6	0.5	0.4	0.25

Figure 3.12

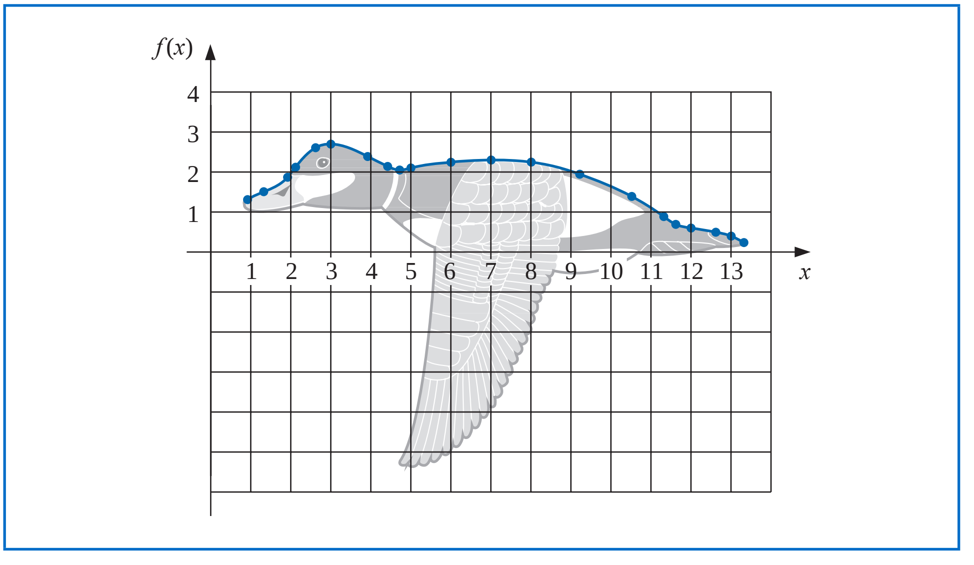


Using Algorithm 3.4 to generate the natural cubic spline for this data produces the coefficients shown in Table 3.19. This spline curve is nearly identical to the profile, as shown in Figure 3.13.

Table 3.19

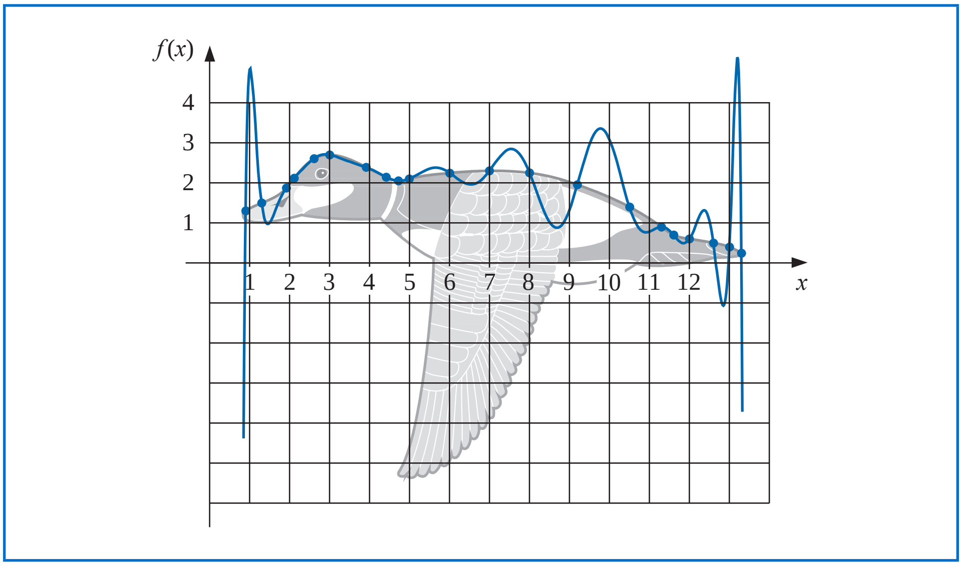
j	x_j	a_j	b_j	c_j	d_j
0	0.9	1.3	5.40	0.00	−0.25
1	1.3	1.5	0.42	−0.30	0.95
2	1.9	1.85	1.09	1.41	−2.96
3	2.1	2.1	1.29	−0.37	−0.45
4	2.6	2.6	0.59	−1.04	0.45
5	3.0	2.7	−0.02	−0.50	0.17
6	3.9	2.4	−0.50	−0.03	0.08
7	4.4	2.15	−0.48	0.08	1.31
8	4.7	2.05	−0.07	1.27	−1.58
9	5.0	2.1	0.26	−0.16	0.04
10	6.0	2.25	0.08	−0.03	0.00
11	7.0	2.3	0.01	−0.04	−0.02
12	8.0	2.25	−0.14	−0.11	0.02
13	9.2	1.95	−0.34	−0.05	−0.01
14	10.5	1.4	−0.53	−0.10	−0.02
15	11.3	0.9	−0.73	−0.15	1.21
16	11.6	0.7	−0.49	0.94	−0.84
17	12.0	0.6	−0.14	−0.06	0.04
18	12.6	0.5	−0.18	0.00	−0.45
19	13.0	0.4	−0.39	−0.54	0.60
20	13.3	0.25			

Figure 3.13



For comparison purposes, Figure 3.14 gives an illustration of the curve that is generated using a Lagrange interpolating polynomial to fit the data given in Table 3.18. The interpolating polynomial in this case is of degree 20 and oscillates wildly. It produces a very strange illustration of the back of a duck, in flight or otherwise.

Figure 3.14



To use a clamped spline to approximate this curve we would need derivative approximations for the endpoints. Even if these approximations were available, we could expect little improvement because of the close agreement of the natural cubic spline to the curve of the top profile. □

Constructing a cubic spline to approximate the lower profile of the ruddy duck would be more difficult since the curve for this portion cannot be expressed as a function of x, and at certain points the curve does not appear to be smooth. These problems can be resolved by using separate splines to represent various portions of the curve, but a more effective approach to approximating curves of this type is considered in the next section.

The clamped boundary conditions are generally preferred when approximating functions by cubic splines, so the derivative of the function must be known or approximated at the endpoints of the interval. When the nodes are equally spaced near both endpoints, approximations can be obtained by any of the appropriate formulas given in Sections 4.1 and 4.2. When the nodes are unequally spaced, the problem is considerably more difficult.

To conclude this section, we list an error-bound formula for the cubic spline with clamped boundary conditions. The proof of this result can be found in [Schul], pp. 57–58.

Theorem 3.13 Let $f \in C^4[a,b]$ with $\max_{a\le x\le b} |f^{(4)}(x)| = M$. If S is the unique clamped cubic spline interpolant to f with respect to the nodes $a = x_0 < x_1 < \cdots < x_n = b$, then for all x in $[a,b]$,

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

A fourth-order error-bound result also holds in the case of natural boundary conditions, but it is more difficult to express. (See [BD], pp. 827–835.)

The natural boundary conditions will generally give less accurate results than the clamped conditions near the ends of the interval $[x_0, x_n]$ unless the function f happens to nearly satisfy $f''(x_0) = f''(x_n) = 0$. An alternative to the natural boundary condition that does not require knowledge of the derivative of f is the *not-a-knot* condition, (see [Deb2], pp. 55–56). This condition requires that $S'''(x)$ be continuous at x_1 and at x_{n-1}.

EXERCISE SET 3.5

1. Determine the natural cubic spline S that interpolates the data $f(0) = 0$, $f(1) = 1$, and $f(2) = 2$.
2. Determine the clamped cubic spline s that interpolates the data $f(0) = 0$, $f(1) = 1$, $f(2) = 2$ and satisfies $s'(0) = s'(2) = 1$.
3. Construct the natural cubic spline for the following data.

a.

x	$f(x)$
8.3	17.56492
8.6	18.50515

b.

x	$f(x)$
0.8	0.22363362
1.0	0.65809197

c.

x	$f(x)$
−0.5	−0.0247500
−0.25	0.3349375
0	1.1010000

d.

x	$f(x)$
0.1	−0.62049958
0.2	−0.28398668
0.3	0.00660095
0.4	0.24842440

4. Construct the natural cubic spline for the following data.

a.

x	$f(x)$
0	1.00000
0.5	2.71828

b.

x	$f(x)$
−0.25	1.33203
0.25	0.800781

c.

x	$f(x)$
0.1	−0.29004996
0.2	−0.56079734
0.3	−0.81401972

d.

x	$f(x)$
−1	0.86199480
−0.5	0.95802009
0	1.0986123
0.5	1.2943767

5. The data in Exercise 3 were generated using the following functions. Use the cubic splines constructed in Exercise 3 for the given value of x to approximate $f(x)$ and $f'(x)$, and calculate the actual error.
   a. $f(x) = x \ln x$; approximate $f(8.4)$ and $f'(8.4)$.
   b. $f(x) = \sin(e^x - 2)$; approximate $f(0.9)$ and $f'(0.9)$.
   c. $f(x) = x^3 + 4.001x^2 + 4.002x + 1.101$; approximate $f(-\frac{1}{3})$ and $f'(-\frac{1}{3})$.
   d. $f(x) = x \cos x - 2x^2 + 3x - 1$; approximate $f(0.25)$ and $f'(0.25)$.
6. The data in Exercise 4 were generated using the following functions. Use the cubic splines constructed in Exercise 4 for the given value of x to approximate $f(x)$ and $f'(x)$, and calculate the actual error.
   a. $f(x) = e^{2x}$; approximate $f(0.43)$ and $f'(0.43)$.
   b. $f(x) = x^4 - x^3 + x^2 - x + 1$; approximate $f(0)$ and $f'(0)$.
   c. $f(x) = x^2 \cos x - 3x$; approximate $f(0.18)$ and $f'(0.18)$.
   d. $f(x) = \ln(e^x + 2)$; approximate $f(0.25)$ and $f'(0.25)$.
7. Construct the clamped cubic spline using the data of Exercise 3 and the fact that
   a. $f'(8.3) = 3.116256$ and $f'(8.6) = 3.151762$
   b. $f'(0.8) = 2.1691753$ and $f'(1.0) = 2.0466965$
   c. $f'(-0.5) = 0.7510000$ and $f'(0) = 4.0020000$
   d. $f'(0.1) = 3.58502082$ and $f'(0.4) = 2.16529366$
8. Construct the clamped cubic spline using the data of Exercise 4 and the fact that
   a. $f'(0) = 2$ and $f'(0.5) = 5.43656$
   b. $f'(-0.25) = 0.437500$ and $f'(0.25) = -0.625000$