Numerical Analysis

10th ed

R L Burden, J D Faires, and A M Burden

Beamer Presentation Slides
Prepared by
Dr. Annette M. Burden
Youngstown State University

September 15, 2015

Theorem (11.1)

Suppose the function f in the boundary-value problem

$$y'' = f(x, y, y'), \quad \text{for } a \le x \le b, \text{ with } y(a) = \alpha \text{ and } y(b) = \beta,$$

is continuous on the set

$$D = \{\, (x, y, y') \ \text{ for } a \le x \le b, \text{ with } -\infty < y < \infty \text{ and } -\infty < y' < \infty \,\},$$

and that the partial derivatives f_y and $f_{y'}$ are also continuous on D. If

(i) $f_y(x, y, y') > 0$, *for all* $(x, y, y') \in D$, *and*

(ii) *a constant M exists, with*

$$|f_{y'}(x, y, y')| \le M, \quad \text{for all } (x, y, y') \in D,$$

then the boundary-value problem has a unique solution.

Corollary (11.2)

Suppose the linear boundary-value problem

$$y'' = p(x)y' + q(x)y + r(x), \text{ for } a \leq x \leq b,$$

with

$$y(a) = \alpha \text{ and } y(b) = \beta, \text{ satisfies}$$

(i) $p(x)$, $q(x)$, *and* $r(x)$ *are continuous on* $[a, b]$,

(ii) $q(x) > 0$ *on* $[a, b]$.

Then the boundary-value problem has a unique solution.

ALGORITHM 11.1 LINEAR SHOOTING

To approximate the solution of the boundary-value problem

$$-y'' + p(x)y' + q(x)y + r(x) = 0, \text{ for } a \le x \le b, \text{ with } y(a) = \alpha \text{ and } y(b) = \beta,$$

INPUT endpoints a, b; boundary conditions α, β; number of subintervals N.

OUTPUT approximations $w_{1,i}$ to $y(x_i)$; $w_{2,i}$ to $y'(x_i)$ for each $i = 0, 1, \ldots, N$.

Step 1 Set $h = (b - a)/N$;

$u_{1,0} = \alpha$; $u_{2,0} = 0$; $v_{1,0} = 0$; $v_{2,0} = 1$.

Step 2 For $i = 0, \ldots, N - 1$ do Steps 3 and 4.

(*Runge-Kutta method for systems used in Steps* 3 & 4.)

Step 3 Set $x = a + ih$.

ALGORITHM 11.1 LINEAR SHOOTING

Step 4 Set

$$
\begin{aligned}
k_{1,1} &= hu_{2,i};\\
k_{1,2} &= h\left[p(x)u_{2,i} + q(x)u_{1,i} + r(x)\right];\\
k_{2,1} &= h\left[u_{2,i} + \tfrac{1}{2}k_{1,2}\right];\\
k_{2,2} &= h\Big[p(x+h/2)\left(u_{2,i} + \tfrac{1}{2}k_{1,2}\right)\\
&\qquad + q(x+h/2)\left(u_{1,i} + \tfrac{1}{2}k_{1,1}\right) + r(x+h/2)\Big];\\
k_{3,1} &= h\left[u_{2,i} + \tfrac{1}{2}k_{2,2}\right];\\
k_{3,2} &= h\Big[p(x+h/2)\left(u_{2,i} + \tfrac{1}{2}k_{2,2}\right)\\
&\qquad + q(x+h/2)(u_{1,i} + \tfrac{1}{2}k_{2,1}) + r(x+h/2)\Big];\\
k_{4,1} &= h\left[u_{2,i} + k_{3,2}\right];\\
&\qquad + q(x+h/2)(u_{1,i} + \tfrac{1}{2}k_{2,1}) + r(x+h/2)\Big];\\
k_{4,2} &= h\big[p(x+h)(u_{2,i} + k_{3,2})\\
&\qquad + q(x+h)(u_{1,i} + k_{3,1}) + r(x+h)\big];
\end{aligned}
$$

ALGORITHM 11.1 LINEAR SHOOTING

$$
\begin{aligned}
u_{1,i+1} &= u_{1,i} + \tfrac{1}{6}\left[k_{1,1} + 2k_{2,1} + 2k_{3,1} + k_{4,1}\right];\\
u_{2,i+1} &= u_{2,i} + \tfrac{1}{6}\left[k_{1,2} + 2k_{2,2} + 2k_{3,2} + k_{4,2}\right];\\
k'_{1,1} &= hv_{2,i};\\
k'_{1,2} &= h\left[p(x)v_{2,i} + q(x)v_{1,i}\right];\\
k'_{2,1} &= h\left[v_{2,i} + \tfrac{1}{2}k'_{1,2}\right];\\
k'_{2,2} &= h\left[p(x+h/2)\left(v_{2,i} + \tfrac{1}{2}k'_{1,2}\right)\right.\\
&\qquad \left. +q(x+h/2)\left(v_{1,i} + \tfrac{1}{2}k'_{1,1}\right)\right];\\
k'_{3,1} &= h\left[v_{2,i} + \tfrac{1}{2}k'_{2,2}\right];\\
k'_{3,2} &= h\left[p(x+h/2)\left(v_{2,i} + \tfrac{1}{2}k'_{2,2}\right)\right.\\
&\qquad \left. +q(x+h/2)\left(v_{1,i} + \tfrac{1}{2}k'_{2,1}\right)\right];
\end{aligned}
$$

ALGORITHM 11.1 LINEAR SHOOTING

$$k'_{4,1} = h\left[v_{2,i} + k'_{3,2}\right];$$

$$k'_{4,2} = h\left[p(x+h)(v_{2,i} + k'_{3,2}) + q(x+h)(v_{1,i} + k'_{3,1})\right];$$

$$v_{1,i+1} = v_{1,i} + \tfrac{1}{6}\left[k'_{1,1} + 2k'_{2,1} + 2k'_{3,1} + k'_{4,1}\right];$$

$$v_{2,i+1} = v_{2,i} + \tfrac{1}{6}\left[k'_{1,2} + 2k'_{2,2} + 2k'_{3,2} + k'_{4,2}\right].$$

Step 5 Set $w_{1,0} = \alpha$;

$w_{2,0} = \frac{\beta - u_{1,N}}{v_{1,N}}$;

OUTPUT $(a, w_{1,0}, w_{2,0})$.

Step 6 For $i = 1, \ldots, N$

set $W1 = u_{1,i} + w_{2,0}v_{1,i}$; $W2 = u_{2,i} + w_{2,0}v_{2,i}$; $x = a + ih$;

OUTPUT $(x, W1, W2)$. (*Output is* $x_i, w_{1,i}, w_{2,i}$.)

Step 7 STOP. (*The process is complete*.)

MOTIVATION

Shooting methods for nonlinear problems require iterations to approach the “target”.

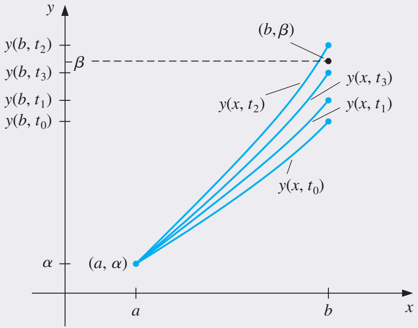

Figure: Figure 11.3

Algorithm 11.2: NONLINEAR SHOOTING NEWTON'S

To approximate the solution of the nonlinear boundary-value problem

$$y'' = f(x, y, y'), \quad \text{for } a \leq x \leq b, \text{ with } y(a) = \alpha \text{ and } y(b) = \beta :$$

INPUT endpoints a, b; boundary conditions α, β; number of subintervals $N \geq 2$; tolerance TOL; maximum number of iterations M.

OUTPUT approximations $w_{1,i}$ to $y(x_i)$; $w_{2,i}$ to $y'(x_i)$ for each $i = 0, 1, \ldots, N$ or a message that the maximum number of iterations was exceeded.

Step 1 Set $h = (b - a)/N$;
$k = 1$;
$TK = (\beta - \alpha)/(b - a)$. (*Note: TK could also be input.*)

Algorithm 11.2: NONLINEAR SHOOTING NEWTON'S

Step 2 While ($k \leq M$) do Steps 3–10.

Step 3 Set $w_{1,0} = \alpha$;
$w_{2,0} = TK$;
$u_1 = 0$;
$u_2 = 1$.

Step 4 For $i = 1, \ldots, N$ do Steps 5 and 6.
(*Runge-Kutta method for systems used in Steps* 5 & 6.)

Step 5 Set $x = a + (i - 1)h$.

Step 6 Set $k_{1,1} = hw_{2,i-1}$;

$$k_{1,2} = hf(x, w_{1,i-1} w_{2,i-1});$$

$$k_{2,1} = h\left(w_{2,i-1} + \frac{1}{2}k_{1,2}\right);$$

$$k_{2,2} = hf\left(x + \frac{h}{2}, w_{1,i-1} + \frac{1}{2}k_{1,1}, w_{2,i-1} + \frac{1}{2}k_{1,2}\right);$$

$$k_{3,1} = h\left(w_{2,i-1} + \frac{1}{2}k_{2,2}\right);$$

Algorithm 11.2: NONLINEAR SHOOTING NEWTON'S

$$k_{3,2} = hf\left(x + \tfrac{h}{2}, w_{1,i-1} + \tfrac{1}{2}k_{2,1}, w_{2,i-1} + \tfrac{1}{2}k_{2,2}\right);$$
$$k_{4,1} = h(w_{2,i-1} + k_{3,2});$$
$$k_{4,2} = hf(x + h, w_{1,i-1} + k_{3,1}, w_{2,i-1} + k_{3,2});$$
$$w_{1,i} = w_{1,i-1} + (k_{1,1} + 2k_{2,1} + 2k_{3,1} + k_{4,1})/6;$$
$$w_{2,i} = w_{2,i-1} + (k_{1,2} + 2k_{2,2} + 2k_{3,2} + k_{4,2})/6;$$
$$k'_{1,1} = hu_2;$$
$$k'_{1,2} = h[f_y(x, w_{1,i-1}, w_{2,i-1})u_1 + f_{y'}(x, w_{1,i-1}, w_{2,i-1})u_2]$$
$$k'_{2,1} = h\left[u_2 + \tfrac{1}{2}k'_{1,2}\right];$$
$$k'_{2,2} = h\left[f_y(x + h/2, w_{1,i-1}, w_{2,i-1})\left(u_1 + \tfrac{1}{2}k'_{1,1}\right)\right.$$
$$\left. + f_{y'}(x + h/2, w_{1,i-1}, w_{2,i-1})\left(u_2 + \tfrac{1}{2}k'_{1,2}\right)\right];$$

Algorithm 11.2: NONLINEAR SHOOTING NEWTON'S

$$k'_{3,1} = h\left(u_2 + \tfrac{1}{2}k'_{2,2}\right);$$
$$k'_{3,2} = h\Big[f_y(x + h/2, w_{1,i-1}, w_{2,i-1})\left(u_1 + \tfrac{1}{2}k'_{2,1}\right)$$
$$+ f_{y'}(x + h/2, w_{1,i-1}, w_{2,i-1})\left(u_2 + \tfrac{1}{2}k'_{2,2}\right)\Big];$$
$$k'_{4,1} = h(u_2 + k'_{3,2});$$
$$k'_{4,2} = h\Big[f_y(x + h, w_{1,i-1}, w_{2,i-1})\left(u_1 + k'_{3,1}\right)$$
$$+ f_{y'}(x + h, w_{1,i-1}, w_{2,i-1})\left(u_2 + k'_{3,2}\right)\Big];$$
$$u_1 = u_1 + \tfrac{1}{6}[k'_{1,1} + 2k'_{2,1} + 2k'_{3,1} + k'_{4,1}];$$
$$u_2 = u_2 + \tfrac{1}{6}[k'_{1,2} + 2k'_{2,2} + 2k'_{3,2} + k'_{4,2}].$$

Algorithm 11.2: NONLINEAR SHOOTING NEWTON'S

Step 7 If $|w_{1,N} - \beta| \leq TOL$ then do Steps 8 and 9.

Step 8 For $i = 0, 1, \ldots, N$

set $x = a + ih$; OUTPUT $(x, w_{1,i}, w_{2,i})$.

Step 9 (*The procedure is complete.*) STOP

Step 10 Set $TK = TK - \frac{w_{1,N} - \beta}{u_1}$;

(*Newton's method used for* TK.)

$k = k + 1$.

Step 11 OUTPUT ('Maximum number of iterations exceeded');

(*Procedure unsuccessful.*) STOP

Theorem (11.3)

Suppose that p, q, and r are continuous on $[a, b]$. *If* $q(x) \geq 0$ *on* $[a, b]$, *then the tridiagonal linear system below has a unique solution provided that* $h < 2/L$, *where* $L = \max_{a \leq x \leq b} |p(x)|$.

Tridiagonal System

The system of equations $A\mathbf{w} = \mathbf{b}$ is expressed in the tridiagonal $N \times N$ matrix form

$$A = \begin{bmatrix} 2 + h^2 q(x_1) & -1 + \frac{h}{2} p(x_1) & 0 & \cdots & 0 \\ -1 - \frac{h}{2} p(x_2) & 2 + h^2 q(x_2) & 1 + \frac{h}{2} p(x_2) & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & -1 + \frac{h}{2} p(x_{N-1}) \\ 0 & \cdots & 0 & 1 - \frac{h}{2} p(x_N) & 2 + h^2 q(x_N) \end{bmatrix}.$$

Tridiagonal System

where

$$\mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_{N-1} \\ w_N \end{bmatrix}, \quad \text{and} \quad \mathbf{b} = \begin{bmatrix} -h^2 r(x_1) + \left(1 + \frac{h}{2} p(x_1)\right) w_0 \\ -h^2 r(x_2) \\ \vdots \\ -h^2 r(x_{N-1}) \\ -h^2 r(x_N) + \left(1 - \frac{h}{2} p(x_N)\right) w_{N+1} \end{bmatrix}.$$

Algorithm 11.3: LINEAR FINITE-DIFFERENCE

To approximate the solution of the boundary-value problem

$y'' = p(x)y' + q(x)y + r(x),$ for $a \le x \le b$, with $y(a) = \alpha$ & $y(b) = \beta$:

INPUT endpoints a, b; boundary conditions α, β; integer $N \ge 2$.

OUTPUT approximations w_i to $y(x_i)$ for each $i = 0, 1, \ldots, N + 1$.

Step 1 Set $h = (b - a)/(N + 1)$;
$x = a + h$;
$a_1 = 2 + h^2 q(x)$;
$b_1 = -1 + (h/2)p(x)$;
$d_1 = -h^2 r(x) + (1 + (h/2)p(x))\alpha$.

Algorithm 11.3: LINEAR FINITE-DIFFERENCE

Step 2 For $i = 2, \ldots, N-1$
set $x = a + ih$;
$a_i = 2 + h^2 q(x)$;
$b_i = -1 + (h/2)p(x)$;
$c_i = -1 - (h/2)p(x)$;
$d_i = -h^2 r(x)$.

Step 3 Set $x = b - h$;
$a_N = 2 + h^2 q(x)$;
$c_N = -1 - (h/2)p(x)$;
$d_N = -h^2 r(x) + (1 - (h/2)p(x))\beta$.

Step 4 Set $l_1 = a_1$;
(*Steps* 4-8 *solve a tridiagonal linear system using Alg.* 6.7)
$u_1 = b_1/a_1$;
$z_1 = d_1/l_1$.

Algorithm 11.3: LINEAR FINITE-DIFFERENCE

Step 5 For $i = 2, \ldots, N-1$ set $l_i = a_i - c_i u_{i-1}$;
$u_i = b_i / l_i$;
$z_i = (d_i - c_i z_{i-1}) / l_i$.

Step 6 Set $l_N = a_N - c_N u_{N-1}$;
$z_N = (d_N - c_N z_{N-1}) / l_N$.

Step 7 Set $w_0 = \alpha$;
$w_{N+1} = \beta$.
$w_N = z_N$.

Step 8 For $i = N-1, \ldots, 1$ set $w_i = z_i - u_i w_{i+1}$.

Step 9 For $i = 0, \ldots, N+1$ set $x = a + ih$;
OUTPUT (x, w_i).

Step 10 STOP. (*The procedure is complete.*)

Algorithm 11.4: NONLINEAR FINITE-DIFFERENCE

To approximate the solution to the nonlinear boundary-value problem

$$y'' = f(x, y, y'), \text{ for } a \le x \le b, \text{ with } y(a) = \alpha \text{ \& } y(b) = \beta :$$

NPUT endpoints a, b; boundary conditions α, β; integer $N \ge 2$; tolerance TOL; maximum number of iterations M.

OUTPUT approximations w_i to $y(x_i)$ for each $i = 0, 1, \ldots, N+1$ or a message that the maximum number of iterations was exceeded.

Step 1 Set $h = (b-a)/(N+1)$;
$w_0 = \alpha$;
$w_{N+1} = \beta$.

Step 2 For $i = 1, \ldots, N$ set $w_i = \alpha + i\left(\frac{\beta-\alpha}{b-a}\right)h$.

Algorithm 11.4: NONLINEAR FINITE-DIFFERENCE

Step 3 Set $k = 1$.

Step 4 While $k \leq M$ do Steps 5–l6.

Step 5 Set $x = a + h$;

$t = (w_2 - \alpha)/(2h)$;

$a_1 = 2 + h^2 f_y(x, w_1, t)$;

$b_1 = -1 + (h/2) f_{y'}(x, w_1, t)$;

$d_1 = -\left(2w_1 - w_2 - \alpha + h^2 f(x, w_1, t)\right)$.

Step 6 For $i = 2, \ldots, N - 1$

set $x = a + ih$;

$t = (w_{i+1} - w_{i-1})/(2h)$;

$a_i = 2 + h^2 f_y(x, w_i, t)$;

$b_i = -1 + (h/2) f_{y'}(x, w_i, t)$;

$c_i = -1 - (h/2) f_{y'}(x, w_i, t)$;

$d_i = -\left(2w_i - w_{i+1} - w_{i-1} + h^2 f(x, w_i, t)\right)$.

Algorithm 11.4: NONLINEAR FINITE-DIFFERENCE

Step 7 Set $x = b - h$;
$t = (\beta - w_{N-1})/(2h)$;
$a_N = 2 + h^2 f_y(x, w_N, t)$;
$c_N = -1 - (h/2) f_{y'}(x, w_N, t)$;
$d_N = -\left(2w_N - w_{N-1} - \beta + h^2 f(x, w_N, t)\right)$.

Step 8 Set $l_1 = a_1$; (*Steps* 8-12 *solve tridiagonal linear system using Algorithm* 6.7.)
$u_1 = b_1/a_1$;
$z_1 = d_1/l_1$.

Step 9 For $i = 2, \ldots, N-1$ set $l_i = a_i - c_i u_{i-1}$;
$u_i = b_i/l_i$;
$z_i = (d_i - c_i z_{i-1})/l_i$.

Step 10 Set $l_N = a_N - c_N u_{N-1}$;
$z_N = (d_N - c_N z_{N-1})/l_N$.

Algorithm 11.4: NONLINEAR FINITE-DIFFERENCE

Step 11 Set $v_N = z_N$;
$w_N = w_N + v_N$.

Step 12 For $i = N-1, \ldots, 1$ set $v_i = z_i - u_i v_{i+1}$;
$w_i = w_i + v_i$.

Step 13 If $\|\mathbf{v}\| \leq TOL$ then do Steps 14 and 15.

Step 14 For $i = 0, \ldots, N+1$ set $x = a + ih$;
OUTPUT (x, w_i).

Step 15 STOP. (*The procedure was successful.*)

Step 16 Set $k = k + 1$.

Step 17 OUTPUT ('Maximum number of iterations exceeded');
(*The procedure was unsuccessful.*)
STOP.

Theorem (11.4)

Let $p \in C^1[0,1]$, *q*, $f \in C[0,1]$, *and*

$$p(x) \geq \delta > 0, \quad q(x) \geq 0, \quad \text{for } 0 \leq x \leq 1.$$

The function $y \in C_0^2[0,1]$ *is the unique solution to the differential equation*

$$-\frac{d}{dx}\left(p(x)\frac{dy}{dx}\right) + q(x)y = f(x), \quad \text{for } 0 \leq x \leq 1,$$

if and only if y *is the unique function in* $C_0^2[0,1]$ *that minimizes the integral*

$$I[u] = \int_0^1 \{p(x)[u'(x)]^2 + q(x)[u(x)]^2 - 2f(x)u(x)\}\, dx.$$

Algorithm 10.5 PIECEWISE LINEAR RAYLEIGH-RITZ

To approximate the solution to the boundary-value problem

$$-\frac{d}{dx}\left(p(x)\frac{dy}{dx}\right)+q(x)y = f(x),\ \text{for } 0 \leq x \leq 1,\quad y(0) = 0 \ \&\ y(1) = 0:$$

with the piecewise linear function

$$\phi(x) = \sum_{i=1}^{n} c_i\phi_i(x):$$

INPUT integer $n \geq 1$; points $x_0 = 0 < x_1 < \cdots < x_n < x_{n+1} = 1$.

OUTPUT coefficients $c_1, \ldots, c_n$.

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

Algorithm 10.5 PIECEWISE LINEAR RAYLEIGH-RITZ

Step 2 For $i = 1, \ldots, n$ define the piecewise linear basis ϕ_i by

$$\phi_i(x) = \begin{cases} 0, & 0 \le x \le x_{i-1}, \\ \frac{x - x_{i-1}}{h_{i-1}}, & x_{i-1} < x \le x_i, \\ \frac{x_{i+1} - x}{h_i}, & x_i < x \le x_{i+1}, \\ 0, & x_{i+1} < x \le 1. \end{cases}$$

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

$Q_{1,i}, Q_{2,i}, Q_{3,i}, Q_{4,i}, Q_{5,i}, Q_{6,i}$;

Compute $Q_{2,n}, Q_{3,n}, Q_{4,n}, Q_{4,n+1}, Q_{5,n}, Q_{6,n}$.

Step 4 For each $i = 1, 2, \ldots, n-1$, set

$\alpha_i = Q_{4,i} + Q_{4,i+1} + Q_{2,i} + Q_{3,i}$;

$\beta_i = Q_{1,i} - Q_{4,i+1}$;

$b_i = Q_{5,i} + Q_{6,i}$.

Algorithm 10.5 PIECEWISE LINEAR RAYLEIGH-RITZ

Step 5 Set $\alpha_n = Q_{4,n} + Q_{4,n+1} + Q_{2,n} + Q_{3,n}$;
$b_n = Q_{5,n} + Q_{6,n}$.

Step 6 Set $a_1 = \alpha_1$; (*Steps* 6-10 *solve symmetric tridiagonal linear system using Algorithm* 6.7.)
$\zeta_1 = \beta_1/\alpha_1$;
$z_1 = b_1/a_1$.

Step 7 For $i = 2, \ldots, n-1$ set $a_i = \alpha_i - \beta_{i-1}\zeta_{i-1}$;
$\zeta_i = \beta_i/a_i$;
$z_i = (b_i - \beta_{i-1}z_{i-1})/a_i$.

Step 8 Set $a_n = \alpha_n - \beta_{n-1}\zeta_{n-1}$;
$z_n = (b_n - \beta_{n-1}z_{n-1})/a_n$.

Step 9 Set $c_n = z_n$; OUTPUT (c_n).

Step 10 For $i = n-1, \ldots, 1$ set $c_i = z_i - \zeta_i c_{i+1}$; OUTPUT ($c_i$).

Step 11 STOP. (*The procedure is complete.*)

Algorithm 10.6 CUBIC SPLINE RAYLEIGH-RITZ

To approximate the solution to the boundary-value problem

$$-\frac{d}{dx}\left(p(x)\frac{dy}{dx}\right)+q(x)y=f(x),\ 0\leq x\leq 1,\ y(0)=0\,y(1)=0$$

with the sum of cubic splines

$$\phi(x)=\sum_{i=0}^{n+1}c_i\phi_i(x):$$

INPUT integer $n \geq 1$.

OUTPUT coefficients $c_0, \ldots, c_{n+1}$.

Step 1 Set $h = 1/(n+1)$.

Algorithm 10.6 CUBIC SPLINE RAYLEIGH-RITZ

Step 2 For $i = 0, \ldots, n+1$ set $x_i = ih$.
Set $x_{-2} = x_{-1} = 0;\ x_{n+2} = x_{n+3} = 1$.

Step 3 Define the function S by

$$S(x) = \begin{cases} 0, & x \leq -2, \\ \frac{1}{4}(2+x)^3, & -2 < x \leq -1, \\ \frac{1}{4}\left[(2+x)^3 - 4(1+x)^3\right], & -1 < x \leq 0, \\ \frac{1}{4}\left[(2-x)^3 - 4(1-x)^3\right], & 0 < x \leq 1, \\ \frac{1}{4}(2-x)^3, & 1 < x \leq 2, \\ 0, & 2 < x \end{cases}$$

Algorithm 10.6 CUBIC SPLINE RAYLEIGH-RITZ

Step 4 Define the cubic spline basis $\{\phi_i\}_{i=0}^{n+1}$ by

$$\phi_0(x) = S\left(\frac{x}{h}\right) - 4S\left(\frac{x+h}{h}\right),$$
$$\phi_1(x) = S\left(\frac{x-x_1}{h}\right) - S\left(\frac{x+h}{h}\right),$$
$$\phi_i(x) = S\left(\frac{x-x_i}{h}\right), \text{ for } i = 2, \ldots, n-1,$$
$$\phi_n(x) = S\left(\frac{x-x_n}{h}\right) - S\left(\frac{x-(n+2)h}{h}\right),$$
$$\phi_{n+1}(x) = S\left(\frac{x-x_{n+1}}{h}\right) - 4S\left(\frac{x-(n+2)h}{h}\right).$$

Step 5 For $i = 0, \ldots, n+1$ do Steps 6–9.

(*Note: The integrals in Steps* 6 *and* 9 *can be evaluated* using a numerical *integration procedure*.)

Step 6 For $j = i, i+1, \ldots, \min\{i+3, n+1\}$

set $L = \max\{x_{j-2}, 0\};\ U = \min\{x_{i+2}, 1\};$

$$a_{ij} = \int_L^U \left[p(x)\phi_i'(x)\phi_j'(x) + q(x)\phi_i(x)\phi_j(x)\right] dx;$$

if $i \neq j$, then set $a_{ji} = a_{ij}$. (*Since A is symmetric.*)

Algorithm 10.6 CUBIC SPLINE RAYLEIGH-RITZ

Step 7 If $i \geq 4$ then for $j = 0, \ldots, i-4$ set $a_{ij} = 0$.
Step 8 If $i \leq n-3$ then for $j = i+4, \ldots, n+1$ set $a_{ij} = 0$.
Step 9 Set $L = \max\{x_{i-2}, 0\}$;
$U = \min\{x_{i+2}, 1\}$;
$b_i = \int_L^U f(x)\phi_i(x)\, dx$.
Step 10 Solve the linear system $A\mathbf{c} = \mathbf{b}$, where
$A = (a_{ij}), \mathbf{b} = (b_0, \ldots, b_{n+1})^t$ and $\mathbf{c} = (c_0, \ldots, c_{n+1})^t$.
Step 11 For $i = 0, \ldots, n+1$
OUTPUT (c_i).
Step 12 STOP. (*The procedure is complete.*)