Chapter 11

Boundary-Value Problems for Ordinary Differential Equations

11.1 Introduction

The differential equations in Chapter 5 are of first order and have one initial condition to satisfy. Later in the chapter we saw that the techniques could be extended to systems of equations and then to higher-order equations, but all the specified conditions must be on the same endpoint. These are initial-value problems. In this chapter we show how to approximate the solution to **two-point boundary-value** problems, differential equations where conditions are imposed at different points. For first-order differential equations only one condition is specified, so there is no distinction between initial-value and boundary-value problems.

The differential equations whose solutions we will approximate are of second order, specifically of the form

$$y'' = f(x, y, y'), \qquad \text{for } a \le x \le b,$$

with the *boundary conditions* on the solution prescribed by

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

for some constants α and β. Such a problem has a unique solution provided that:

- The function f and its partial derivatives with respect to y and y' are continuous;
- The partial derivative of f with respect to y is positive; and
- The partial derivative of f with respect to y' is bounded.

These are all reasonable conditions for boundary-value problems representing physical problems.

11.2 The Linear Shooting Method

A boundary-value problem is **linear** when the function f has the form

$$f(x, y, y') = p(x)y' + q(x)y + r(x).$$

Linear problems occur frequently in applications and are much easier to solve than nonlinear equations. This is because adding any solution to the **inhomogeneous** differential equation

$$y'' - p(x)y' - q(x)y = r(x)$$

to the complete solution of the **homogeneous** differential equation

$$y'' - p(x)y' - q(x)y = 0$$

gives all the solutions to the inhomogeneous problem. The solutions of the homogeneous problem are easier to determine than are those of the inhomogeneous. Moreover, to show that a linear problem has a unique solution, we need only show that p, q, and r are continuous and that the values of q are positive.

To approximate the unique solution to the linear boundary-value problem, let us first consider the two initial-value problems

$$y'' = p(x)y' + q(x)y + r(x), \text{ for } a \le x \le b, \text{ where } y(a) = \alpha \text{ and } y'(a) = 0, \quad (11.1)$$

and

$$y'' = p(x)y' + q(x)y, \quad \text{for } a \le x \le b, \text{ where } y(a) = 0 \text{ and } y'(a) = 1, \quad (11.2)$$

both of which have unique solutions. Let $y_1(x)$ denote the solution to Eq. (11.1), $y_2(x)$ denote the solution to Eq. (11.2), and assume that $y(b) \neq 0$. (The situation when $y_2(b) = 0$ is considered in Exercise 8.) Then,

$$y(x) = y_1(x) + \frac{\beta - y_1(b)}{y_2(b)} y_2(x) \quad (11.3)$$

is the unique solution to the linear boundary-value problem

$$y'' = p(x)y' + q(x)y + r(x), \text{ for } a \le x \le b, \text{ with } y(a) = \alpha \text{ and } y(b) = \beta. \quad (11.4)$$

To verify this, first note that

$$\begin{aligned} y'' - p(x)y' - q(x)y &= y_1'' - p(x)y_1' - q(x)y_1 + \frac{\beta - y_1(b)}{y_2(b)}[y_2'' - p(x)y_2' - q(x)y_2] \\ &= r(x) + \frac{\beta - y_1(b)}{y_2(b)} \cdot 0 = r(x). \end{aligned}$$

Moreover,

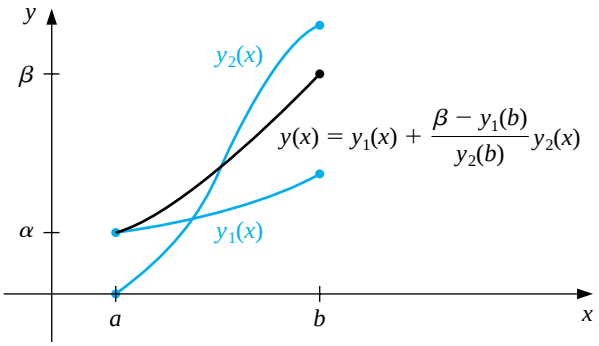
$$y(a) = y_1(a) + \frac{\beta - y_1(b)}{y_2(b)} y_2(a) = y_1(a) + \frac{\beta - y_1(b)}{y_2(b)} \cdot 0 = \alpha$$

and

$$y(b) = y_1(b) + \frac{\beta - y_1(b)}{y_2(b)} y_2(b) = y_1(b) + \beta - y_1(b) = \beta.$$

The Linear Shooting method is based on the replacement of the boundary-value problem by the two initial-value problems, (11.1) and (11.2). Numerous methods are available from Chapter 5 for approximating the solutions $y_1(x)$ and $y_2(x)$, and once these approximations are available, the solution to the boundary-value problem is approximated using the weighted sum in Eq. (11.3). Graphically, the method has the appearance shown in Figure 11.1.

Figure 11.1



The program LINST111 incorporates the Runge-Kutta method of order 4 to find the approximations to $y_1(x)$ and $y_2(x)$, but any technique for approximating the solutions to initial-value problems can be substituted. The program has the additional feature of obtaining approximations for the derivative of the solution to the boundary-value problem in addition to the solution of the problem itself.

EXAMPLE 1 The boundary-value problem

$$y'' = -\frac{2}{x}y' + \frac{2}{x^2}y + \frac{\sin(\ln x)}{x^2}, \quad \text{for } 1 \le x \le 2, \quad \text{where} \quad y(1) = 1 \quad \text{and} \quad y(2) = 2,$$

has the exact solution

$$y = c_1 x + \frac{c_2}{x^2} - \frac{3}{10}\sin(\ln x) - \frac{1}{10}\cos(\ln x),$$

where

$$c_2 = \frac{1}{70}(8 - 12\sin(\ln 2) - 4\cos(\ln 2)) \quad \text{and} \quad c_1 = \frac{11}{10} - c_2.$$

Applying the Linear Shooting method to this problem requires approximating the solutions to the initial-value problems

$$y_1'' = -\frac{2}{x}y_1' + \frac{2}{x^2}y_1 + \frac{\sin(\ln x)}{x^2}, \quad \text{for } 1 \le x \le 2, \quad \text{where} \quad y_1(1) = 1 \quad \text{and} \quad y_1'(1) = 0,$$

and

$$y_2'' = -\frac{2}{x}y_2' + \frac{2}{x^2}y_2, \qquad \text{for } 1 \le x \le 2, \quad \text{where} \quad y_2(1) = 0 \quad \text{and} \quad y_2'(1) = 1.$$

We will use the Runge-Kutta method of order 4 within Maple to solve both differential equations. The first second-order equation is written as a system of two first-order differential equations $u_1' = f_1(x, u_1, u_2)$ and $u_2' = f_2(x, u_1, u_2)$. We define the system and initial conditions with

```
>sys1:=D(u1)(x)=u2(x), D(u2)(x)=-2*u2(x)/x+2*u1(x)/x^2+sin(ln(x))/x^2;
>init1:=u1(1)=1,u2(1)=0;
```

The Runge-Kutta method of order 4 is invoked with the command

```
>g1:=dsolve({sys1,init1},numeric,method=classical[rk4],{u1(x),u2(x)},stepsize=0.1);
```

The next second-order differential equation is defined as a system of of two first-order differential equations by

```
>sys2:=D(u1)(x)=u2(x), D(u2)(x)=-2*u2(x)/x+2*u1(x)/x^2;
>init2:=u1(1)=0,u2(1)=2;
```

and the Runge-Kutta method of order 4 is invoked with the command

```
>g2:=dsolve({sys2,init2},numeric,method=classical[rk4],{u1(x),u2(x)},stepsize=0.1);
```

We form the combination

$$y(x) = y_1(x) + \frac{2 - y_1(2)}{y_2(2)}y_2(x)$$

using the Maple code

```
>c:=(2-rhs(g1(2)[2]))/rhs(g2(2)[2]));
>for i from 1 to 10 do
>x:=1+0.1*i;
> w[i]:=rhs(g1(x)[2])+c*rhs(g2(x)[2]);
>od;
```

This gives the results presented in the fourth column of Table 11.1. The value listed as $u_{1,i}$ approximates $y_1(x_i)$, the value of $v_{1,i}$ approximates $y_2(x_i)$, and w_i approximates $y(x_i)$. □

and gives the values of $u_{1,i}$ in the second column of Table 11.1.

Table 11.1

i	x_i	$u_{1,i}$	$v_{1,i}$	w_i	$y(x_i)$	$\lvert y(x_i) - w_i \rvert$
0	1.0	1.00000000	0.00000000	1.00000000	1.00000000	
1	1.1	1.00896058	0.09117986	1.09262916	1.09262930	1.43×10^{-7}
2	1.2	1.03245472	0.16851175	1.18708471	1.18708484	1.34×10^{-7}
3	1.3	1.06674375	0.23608704	1.28338227	1.28338236	9.78×10^{-8}
4	1.4	1.10928795	0.29659067	1.38144589	1.38144595	6.02×10^{-8}
5	1.5	1.15830000	0.35184379	1.48115939	1.48115942	3.06×10^{-8}
6	1.6	1.21248371	0.40311695	1.58239245	1.58239246	1.08×10^{-8}
7	1.7	1.27087454	0.45131840	1.68501396	1.68501396	5.43×10^{-10}
8	1.8	1.33273851	0.49711137	1.78889854	1.78889853	5.05×10^{-9}
9	1.9	1.39750618	0.54098928	1.89392951	1.89392951	4.41×10^{-9}
10	2.0	1.46472815	0.58332538	2.00000000	2.00000000	

The accurate results in this example are due to the fact that the Runge-Kutta method of order 4 gives $O(h^4)$ approximations to the solutions of the initial-value problems. Unfortunately, there can be round-off error problems hidden in this technique. If $y_1(x)$ rapidly increases as x goes from a to b, then $u_{1,N} \approx y_1(b)$ will be large. Should β be small in magnitude compared to $u_{1,N}$, the term $(\beta - u_{1,N})/v_{1,N}$ will be approximately $-u_{1,N}/v_{1,N}$. So the approximations

$$y(x_i) \approx w_i = u_{1,i} - \left(\frac{\beta - u_{1,N}}{v_{1,N}}\right) v_{1,i}, \approx u_{1,i} - \left(\frac{u_{1,N}}{v_{1,N}}\right) v_{1,i}$$

allow the possibility of a loss of significant digits due to cancellation. However, since $u_{1,i}$ is an approximation to $y_1(x_i)$, the behavior of y_1 can be easily monitored, and if $u_{1,i}$ increases rapidly from a to b, the shooting technique can be employed in the other direction—that is, solving instead the initial-value problems

$$y'' = p(x)y' + q(x)y + r(x), \qquad \text{for } a \le x \le b, \quad \text{where} \quad y(b) = \beta \quad \text{and} \quad y'(b) = 0,$$

and

$$y'' = p(x)y' + q(x)y, \qquad \text{for } a \le x \le b, \quad \text{where} \quad y(b) = 0 \quad \text{and} \quad y'(b) = 1.$$

If the reverse shooting technique still gives cancellation of significant digits and if increased precision does not yield greater accuracy, other techniques must be employed. In general, however, if $u_{1,i}$ and $v_{1,i}$ are $O(h^n)$ approximations to $y_1(x_i)$ and $y_2(x_i)$, respectively, for each $i = 0, 1, \ldots, N$, then $w_{1,i}$ will be an $O(h^n)$ approximation to $y(x_i)$.

EXERCISE SET 11.2

1. The boundary-value problem

$$y'' = 4(y - x), \qquad \text{for } 0 \le x \le 1 \quad \text{with } y(0) = 0 \text{ and } y(1) = 2$$

has the solution $y(x) = e^2(e^4 - 1)^{-1}(e^{2x} - e^{-2x}) + x$. Use the Linear Shooting method to approximate the solution and compare the results to the actual solution.

(a) With $h = \frac{1}{2}$

(b) With $h = \frac{1}{4}$

2. The boundary-value problem

$$y'' = y' + 2y + \cos x, \qquad \text{for } 0 \le x \le \frac{\pi}{2} \quad \text{with } y(0) = -0.3 \text{ and } y\left(\frac{\pi}{2}\right) = -0.1$$

has the solution $y(x) = -\frac{1}{10}(\sin x + 3\cos x)$. Use the Linear Shooting method to approximate the solution and compare the results to the actual solution.

(a) With $h = \dfrac{\pi}{4}$

(b) With $h = \dfrac{\pi}{8}$.

3. Use the Linear Shooting method to approximate the solution to the following boundary-value problems.

   (a) $y'' = -3y' + 2y + 2x + 3$, for $0 \le x \le 1$ with $y(0) = 2$ and $y(1) = 1$; use $h = 0.1$.

   (b) $y'' = -\dfrac{4}{x}y' + \dfrac{2}{x^2}y - \dfrac{2\ln x}{x^2}$, for $1 \le x \le 2$ with $y(1) = -\dfrac{1}{2}$ and $y(2) = \ln 2$; use $h = 0.05$.

   (c) $y'' = -(x+1)y' + 2y + (1 - x^2)e^{-x}$, for $0 \le x \le 1$ with $y(0) = -1$ and $y(1) = 0$; use $h = 0.1$.

   (d) $y'' = \dfrac{y'}{x} + \dfrac{3}{x^2}y + \dfrac{\ln x}{x} - 1$, for $1 \le x \le 2$ with $y(1) = y(2) = 0$; use $h = 0.1$.

4. Although $q(x) < 0$ in the following boundary-value problems, unique solutions exist and are given. Use the Linear Shooting method to approximate the solutions to the following problems and compare the results to the actual solutions.

   (a) $y'' + y = 0$, for $0 \le x \le \dfrac{\pi}{4}$ with $y(0) = 1$ and $y\left(\dfrac{\pi}{4}\right) = 1$; use $h = \dfrac{\pi}{20}$; actual solution $y(x) = \cos x + (\sqrt{2} - 1)\sin x$.

(b) $y'' + 4y = \cos x$, for $0 \le x \le \dfrac{\pi}{4}$ with $y(0) = 0$ and $y\left(\dfrac{\pi}{4}\right) = 0$; use $h = \dfrac{\pi}{20}$; actual solution $y(x) = -\dfrac{1}{3}\cos 2x - \dfrac{\sqrt{2}}{6}\sin 2x + \dfrac{1}{3}\cos x$.

(c) $y'' = -\dfrac{4}{x}y' - \dfrac{2}{x^2}y + \dfrac{2}{x^2}\ln x$, for $1 \le x \le 2$ with $y(1) = \dfrac{1}{2}$ and $y(2) = \ln 2$; use $h = 0.05$; actual solution $y(x) = \dfrac{4}{x} - \dfrac{2}{x^2} + \ln x - \dfrac{3}{2}$.

(d) $y'' = 2y' - y + xe^x - x$, for $0 \le x \le 2$ with $y(0) = 0$ and $y(2) = -4$; use $h = 0.2$; actual solution $y(x) = \dfrac{1}{6}x^3e^x - \dfrac{5}{3}xe^x + 2e^x - x - 2$.

5. Use the Linear Shooting method to approximate the solution $y = e^{-10x}$ to the boundary-value problem

$$y'' = 100y, \qquad \text{for } 0 \le x \le 1 \quad \text{with} \quad y(0) = 1 \quad \text{and} \quad y(1) = e^{-10}.$$

Use $h = 0.1$ and 0.05.

6. Write the second-order initial-value problems (11.1) and (11.2) as first-order systems, and derive the equations necessary to solve the systems using the fourth-order Runge-Kutta method for systems.

7. Let u represent the electrostatic potential between two concentric metal spheres of radii R_1 and R_2 ($R_1 < R_2$), such that the potential of the inner sphere is kept constant at V_1 volts and the potential of the outer sphere is 0 volts. The potential in the region between the two spheres is governed by Laplace's equation, which, in this particular application, reduces to

$$\frac{d^2u}{dr^2} + \frac{2}{r}\frac{du}{dr} = 0, \qquad \text{for } R_1 \le r \le R_2 \quad \text{with} \quad u(R_1) = V_1 \quad \text{and} \quad u(R_2) = 0.$$

Suppose $R_1 = 2$ in., $R_2 = 4$ in., and $V_1 = 110$ volts.

(a) Approximate $u(3)$ using the Linear Shooting method.

(b) Compare the results of part (a) with the actual potential $u(3)$, where

$$u(r) = \frac{V_1R_1}{r}\left(\frac{R_2 - r}{R_2 - R_1}\right).$$

8. Show that if y_2 is the solution to $y'' = p(x)y' + q(x)y$ and $y_2(a) = y_2(b) = 0$, then $y_2 \equiv 0$.

9. Consider the boundary-value problem

$$y'' + y = 0, \qquad \text{for } 0 \le x \le b \quad \text{with} \quad y(0) = 0 \quad \text{and} \quad y(b) = B.$$

Find choices for b and B so that the boundary-value problem has

(a) No solution;

(b) Exactly one solution;

(c) Infinitely many solutions.

10. Explain what happens when you attempt to apply the instructions in Exercise 9 to the boundary-value problem

$$y'' - y = 0, \qquad \text{for } 0 \le x \le b \quad \text{with} \quad y(0) = 0 \quad \text{and} \quad y(b) = B.$$

11.3 Linear Finite Difference Methods

The Shooting method discussed in Section 11.2 often has round-off error difficulties. The methods we present in this section have better rounding characteristics, but they generally require more computation to obtain a specified accuracy.

Methods involving finite differences for solving boundary-value problems replace each of the derivatives in the differential equation with an appropriate difference-quotient approximation of the type considered in Section 4.9. The particular difference quotient is chosen to maintain a specified order of error.

The finite-difference method for the linear second-order boundary-value problem,

$$y'' = p(x)y' + q(x)y + r(x), \qquad \text{for } a \le x \le b, \quad \text{where} \quad y(a) = \alpha \quad \text{and} \quad y(b) = \beta,$$

requires that difference-quotient approximations be used for approximating both y' and y''. First, we select an integer $N > 0$ and divide the interval $[a, b]$ into $(N + 1)$ equal subintervals whose endpoints are the mesh points $x_i = a + ih$, for $i = 0, 1, \ldots, N + 1$, where $h = (b - a)/(N + 1)$.

At the interior mesh points, x_i, for $i = 1, 2, \ldots, N$, the differential equation to be approximated is

$$y''(x_i) = p(x_i)y'(x_i) + q(x_i)y(x_i) + r(x_i). \tag{11.5}$$

Expanding $y(x)$ in a third-degree Taylor polynomial about x_i evaluated at x_{i+1} and x_{i-1}, we have, assuming that $y \in C^4[x_{i-1}, x_{i+1}]$,

$$y(x_{i+1}) = y(x_i + h) = y(x_i) + hy'(x_i) + \frac{h^2}{2}y''(x_i) + \frac{h^3}{6}y'''(x_i) + \frac{h^4}{24}y^{(4)}(\xi_i^+),$$

for some ξ_i^+ in (x_i, x_{i+1}), and

$$y(x_{i-1}) = y(x_i - h) = y(x_i) - hy'(x_i) + \frac{h^2}{2}y''(x_i) - \frac{h^3}{6}y'''(x_i) + \frac{h^4}{24}y^{(4)}(\xi_i^-),$$

for some ξ_i^- in (x_{i-1}, x_i). If these equations are added, we have

$$y(x_{i+1}) + y(x_{i-1}) = 2y(x_i) + h^2y''(x_i) + \frac{h^4}{24}\left[y^{(4)}\left(\xi_i^+\right) + y^{(4)}\left(\xi_i^-\right)\right],$$

and a simple algebraic manipulation gives

$$y''(x_i) = \frac{1}{h^2}\left[y(x_{i+1}) - 2y(x_i) + y(x_{i-1})\right] - \frac{h^2}{24}\left[y^{(4)}\left(\xi_i^+\right) + y^{(4)}\left(\xi_i^-\right)\right].$$

The Intermediate Value Theorem can be used to simplify this even further.

[Centered-Difference Formula for $y''(x_i)$]

$$y''(x_i) = \frac{1}{h^2}\left[y(x_{i+1}) - 2y(x_i) + y(x_{i-1})\right] - \frac{h^2}{12}y^{(4)}(\xi_i),$$

for some ξ_i in (x_{i-1}, x_{i+1}).

A centered-difference formula for $y'(x_i)$ is obtained in a similar manner.

[Centered-Difference Formula for $y'(x_i)$]

$$y'(x_i) = \frac{1}{2h}[y(x_{i+1}) - y(x_{i-1})] - \frac{h^2}{6}y'''(\eta_i),$$

for some η_i in (x_{i-1}, x_{i+1}).

The use of these centered-difference formulas in Eq. (11.5) results in the equation

$$\begin{aligned}\frac{y(x_{i+1}) - 2y(x_i) + y(x_{i-1})}{h^2} &= p(x_i)\left[\frac{y(x_{i+1}) - y(x_{i-1})}{2h}\right] + q(x_i)y(x_i) \\ &\quad + r(x_i) - \frac{h^2}{12}[2p(x_i)y'''(\eta_i) - y^{(4)}(\xi_i)].\end{aligned}$$

A Finite-Difference method with truncation error of order $O(h^2)$ results from using this equation together with the boundary conditions $y(a) = \alpha$ and $y(b) = \beta$ to define

$$w_0 = \alpha, \qquad w_{N+1} = \beta,$$

and

$$\left(\frac{2w_i - w_{i+1} - w_{i-1}}{h^2}\right) + p(x_i)\left(\frac{w_{i+1} - w_{i-1}}{2h}\right) + q(x_i)w_i = -r(x_i)$$

for each $i = 1, 2, \ldots, N$.

In the form we will consider, the equation is rewritten as

$$-\left(1 + \frac{h}{2}p(x_i)\right)w_{i-1} + \left(2 + h^2q(x_i)\right)w_i - \left(1 - \frac{h}{2}p(x_i)\right)w_{i+1} = -h^2r(x_i),$$

and the resulting system of equations is expressed in the tridiagonal $N \times N$ matrix

form $A\mathbf{w} = \mathbf{b}$,where

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

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

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

The program LINFD112 implements the Linear Finite-Difference method.

EXAMPLE 1 The Linear Finite-Difference method will be used to approximate the solution to the linear boundary-value problem

$$y'' = -\frac{2}{x}y' + \frac{2}{x^2}y + \frac{\sin(\ln x)}{x^2}, \quad \text{for } 1 \leq x \leq 2, \quad \text{where} \quad y(1) = 1 \quad \text{and} \quad y(2) = 2,$$

which was also approximated by the Shooting method in Example 1 of Section 11.2. For this example, we use the same spacing as in Example 1 of Section 11.2.

To use Maple to apply the Linear Finite-Difference method, first we need to access the linear algebra library with the command

```
>with(linalg);
```

Then we define the endpoints of the interval, the boundary conditions, N, and h.

```
>a:=1; b:=2; alpha:=1; beta:=2; N:=9; h:=(b-a)/(N+1);
```

The mesh points are defined in the following loop:

```
>for i from 1 to N do
>x[i]:=a+i*h;
>od;
```

The functions $p(x)$, $q(x)$,and $r(x)$ are defined by

```
>p:=x->-2/x;
>q:=x->2/x^2;
>r:=x->sin(ln(x))/x^2;
```

We initialize the 9×10 array A as the zero matrix.

```
>A:=matrix(9,10,0);
```

Then we generate the nonzero entries with the following statements:

```
>A[1,1]:=2+h*h*evalf(q(x[1]));
>A[1,2]:=-1+h*evalf(p(x[1]))/2;
>A[1,N+1]:=-h*h*evalf(r(x[1]))+(1+h*p(x[1])/2)*alpha;
>for i from 2 to N-1 do
>A[i,i-1]:=-1-h*evalf(p(x[i]))/2;
>A[i,i]:=2+h*h*evalf(q(x[i]));
>A[i,i+1]:=-1+h*evalf(p(x[i]))/2;
>A[i,N+1]:=-h*h*evalf(r(x[i]));
>od;
>A[N,N-1]:=-1-h*evalf(p(x[N]))/2;
>A[N,N]:=2+h*h*evalf(q(x[N]));
>A[N,N+1]:=-h*h*evalf(r(x[N]))+(1-h*p(x[N])/2)*beta;
```

We now apply Gaussian elimination to solve the 9×9 linear system for the values $w_1, w_2, \ldots, w_9$.

```
>C:=gausselim(A);
>w:=backsub(C);
```

The ith component, w_i, of the vector $\mathbf{w}$ gives the approximation to $y(t_i)$ for each $i = 1, 2, \ldots, n$. The complete results are presented in Table 11.2. □

Table 11.2

i	x_i	w_i	$y(x_i)$	$\lvert w_i - y(x_i)\rvert$
0	1.0	1.00000000	1.00000000	
1	1.1	1.09260052	1.09262930	2.88×10^{-5}
2	1.2	1.18704313	1.18708484	4.17×10^{-5}
3	1.3	1.28333687	1.28338236	4.55×10^{-5}
4	1.4	1.38140204	1.38144595	4.39×10^{-5}
5	1.5	1.48112026	1.48115942	3.92×10^{-5}
6	1.6	1.58235990	1.58239246	3.26×10^{-5}
7	1.7	1.68498902	1.68501396	2.49×10^{-5}
8	1.8	1.78888175	1.78889853	1.68×10^{-5}
9	1.9	1.89392110	1.89392951	8.41×10^{-6}
10	2.0	2.00000000	2.00000000	

Note that these results are considerably less accurate than those obtained in Example 1 of Section 11.2 and listed in Table 11.1. This is because the method used in Section 11.2 involved a Runge-Kutta technique with error of order $O(h^4)$, whereas the difference method used here has error of order $O(h^2)$.

To obtain a difference method with greater accuracy, we can proceed in a number of ways. Using fifth-order Taylor series for approximating $y''(x_i)$ and $y'(x_i)$ results in an error term involving h^4. However, this requires using multiples not only of $y(x_{i+1})$ and $y(x_{i-1})$, but also $y(x_{i+2})$ and $y(x_{i-2})$ in the approximation formulas for $y''(x_i)$ and $y'(x_i)$. This leads to difficulty at $i = 0$ and $i = N$. Moreover, the resulting system of equations is not in tridiagonal form, and the solution to the system requires many more calculations.

Instead of obtaining a difference method with a higher-order error term in this manner, it is generally more satisfactory to consider a reduction in step size. In addition, the Richardson's extrapolation technique can be used effectively for this method, since the error term is expressed in even powers of h with coefficients independent of h, provided y is sufficiently differentiable.

EXAMPLE 2 Richardson's extrapolation for approximating the solution to the boundary-value problem

$$y'' = -\frac{2}{x}y' + \frac{2}{x^2}y + \frac{\sin(\ln x)}{x^2}, \qquad \text{for } 1 \le x \le 2, \quad \text{where} \quad y(1) = 1 \quad \text{and} \quad y(2) = 2,$$

with $h = 0.1, 0.05$, and 0.025, gives the results listed in Table 11.3. The first extrapolation is

$$\text{Ext}_{1i} = \frac{4w_i(h = 0.05) - w_i(h = 0.1)}{3};$$

the second extrapolation is

$$\text{Ext}_{2i} = \frac{4w_i(h = 0.025) - w_i(h = 0.05)}{3};$$

Table 11.3

i	x_i	$w_i(h = 0.1)$	$w_i(h = 0.05)$	$w_i(h = 0.025)$	Ext_{1i}	Ext_{2i}	Ext_{3i}
0	1.0	1.00000000	1.00000000	1.00000000	1.00000000	1.00000000	1.00000000
1	1.1	1.09260052	1.09262207	1.09262749	1.09262925	1.09262930	1.09262930
2	1.2	1.18704313	1.18707436	1.18708222	1.18708477	1.18708484	1.18708484
3	1.3	1.28333687	1.28337094	1.28337950	1.28338230	1.28338236	1.28338236
4	1.4	1.38140204	1.38143493	1.38144319	1.38144589	1.38144595	1.38144595
5	1.5	1.48112026	1.48114959	1.48115696	1.48115937	1.48115941	1.48115942
6	1.6	1.58235990	1.58238429	1.58239042	1.58239242	1.58239246	1.58239246
7	1.7	1.68498902	1.68500770	1.68501240	1.68501393	1.68501396	1.68501396
8	1.8	1.78888175	1.78889432	1.78889748	1.78889852	1.78889853	1.78889853
9	1.9	1.89392110	1.89392740	1.89392898	1.89392950	1.89392951	1.89392951
10	2.0	2.00000000	2.00000000	2.00000000	2.00000000	2.00000000	2.00000000

and the final extrapolation is

$$\text{Ext}_{3i} = \frac{16\text{Ext}_{2i} - \text{Ext}_{1i}}{15}.$$

All the results of Ext_{3i} are correct to the decimal places listed. In fact, if sufficient digits are maintained, this approximation gives results that agree with the exact solution with a maximum error of 6.3×10^{-11}. □

EXERCISE SET 11.3

1. The boundary-value problem

$$y'' = 4(y - x), \qquad \text{for } 0 \le x \le 1 \quad \text{with} \quad y(0) = 0 \quad \text{and} \quad y(1) = 2$$

has the solution $y(x) = e^2(e^4 - 1)^{-1}(e^{2x} - e^{-2x}) + x$. Use the Linear Finite-Difference method to approximate the solution and compare the results to the actual solution.

(a) With $h = \frac{1}{2}$

(b) With $h = \frac{1}{4}$.

(c) Use extrapolation to approximate $y(1/2)$.

2. The boundary-value problem

$$y'' = y' + 2y + \cos x, \text{ for } 0 \le x \le \frac{\pi}{2} \text{ with } y(0) = -0.3 \text{ and } y\left(\frac{\pi}{2}\right) = -0.1$$

has the solution $y(x) = -\frac{1}{10}(\sin x + 3\cos x)$. Use the Linear Finite-Difference method to approximate the solution and compare the results to the actual solution.

(a) With $h = \dfrac{\pi}{4}$

(b) With $h = \dfrac{\pi}{8}$

(c) Use extrapolation to approximate $y(\pi/4)$.

3. Use the Linear Finite-Difference method to approximate the solution to the following boundary-value problems.

(a) $y'' = -3y' + 2y + 2x + 3$, for $0 \le x \le 1$ with $y(0) = 2$ and $y(1) = 1$; use $h = 0.1$.

(b) $y'' = -\dfrac{4}{x}y' + \dfrac{2}{x^2}y - \dfrac{2}{x^2}\ln x$, for $1 \le x \le 2$ with $y(1) = -\dfrac{1}{2}$ and $y(2) = \ln 2$; use $h = 0.05$.

(c) $y'' = -(x+1)y' + 2y + (1 - x^2)e^{-x}$, for $0 \le x \le 1$ with $y(0) = -1$ and $y(1) = 0$; use $h = 0.1$.

(d) $y'' = \dfrac{y'}{x} + \dfrac{3}{x^2}y + \dfrac{\ln x}{x} - 1$, for $1 \le x \le 2$ for $y(1) = y(2) = 0$; use $h = 0.1$.

4. Although $q(x) < 0$ in the following boundary-value problems, unique solutions exist and are given. Use the Linear Finite-Difference method to approximate the solutions and compare the results to the actual solutions.

(a) $y'' + y = 0$, for $0 \le x \le \dfrac{\pi}{4}$ with $y(0) = 1$ and $y\left(\dfrac{\pi}{4}\right) = 1$; use $h = \dfrac{\pi}{20}$; actual solution $y(x) = \cos x + (\sqrt{2} - 1)\sin x$.

(b) $y'' + 4y = \cos x$, for $0 \le x \le \dfrac{\pi}{4}$ with $y(0) = 0$ and $y\left(\dfrac{\pi}{4}\right) = 0$; use $h = \dfrac{\pi}{20}$; actual solution $y(x) = -\dfrac{1}{3}\cos 2x - \dfrac{\sqrt{2}}{6}\sin 2x + \dfrac{1}{3}\cos x$.

(c) $y'' = -\dfrac{4}{x}y' - \dfrac{2}{x^2}y + \dfrac{2\ln x}{x^2}$, for $1 \le x \le 2$ with $y(1) = \dfrac{1}{2}$ and $y(2) = \ln 2$; use $h = 0.05$; actual solution $y(x) = \dfrac{4}{x} - \dfrac{2}{x^2} + \ln x - \dfrac{3}{2}$.

(d) $y'' = 2y' - y + xe^x - x$, for $0 \le x \le 2$ with $y(0) = 0$ and $y(2) = -4$; use $h = 0.2$; actual solution $y(x) = \dfrac{1}{6}x^3e^x - \dfrac{5}{3}xe^x + 2e^x - x - 2$.

5. Use the Linear Finite-Difference method to approximate the solution $y = e^{-10x}$ to the boundary-value problem

$$y'' = 100y, \qquad \text{for } 0 \le x \le 1 \quad \text{with} \quad y(0) = 1 \quad \text{and} \quad y(1) = e^{-10}.$$

Use $h = 0.1$ and 0.05. Can you explain the consequences?

6. Repeat Exercise 3(a) and (b) using the extrapolation discussed in Example 2.

7. The deflection of a uniformly loaded, long rectangular plate under an axial tension force is governed by a second-order differential equation. Let S represent the axial force and q, the intensity of the uniform load. The deflection w along the elemental length is given by

$$w''(x) - \frac{S}{D}w(x) = \frac{-ql}{2D}x + \frac{q}{2D}x^2, \qquad \text{for } 0 \le x \le l \quad \text{with} \quad w(0) = w(l) = 0,$$

where l is the length of the plate and D is the flexual rigidity of the plate. Let $q = 200$ lb/in.^{2}, $S = 100$ lb/in., $D = 8.8 \times 10^7$ lb/in., and $l = 50$ in. Approximate the deflection at 1-in. intervals.

8. The boundary-value problem governing the deflection of a beam with supported ends subject to uniform loading is

$$w''(x) = \frac{S}{EI}w(x) + \frac{q}{2EI}x(x - l), \qquad \text{for } 0 < x < l \quad \text{with} \quad w(0) = 0 \quad \text{and} \quad w(l) = 0.$$

Suppose the beam is a W10-type steel I-beam with the following characteristics: length $l = 120$ in., intensity of uniform load $q = 100$ lb/ft, modulus of elasticity $E = 3.0 \times 10^7$ lb/in.^{2}, stress at ends $S = 1000$ lb, and central moment of inertia $I = 625$ in.^{4}.

(a) Approximate the deflection $w(x)$ of the beam every 6 in.

(b) The actual relationship is given by

$$w(x) = c_1 e^{ax} + c_2 e^{-ax} + b(x - l)x + c,$$

where $c_1 = 7.7042537 \times 10^4$, $c_2 = 7.9207462 \times 10^4$, $a = 2.3094010 \times 10^{-4}$, $b = -4.1666666 \times 10^{-3}$, and $c = -1.5625 \times 10^5$. Is the maximum error on the interval within 0.2 in.?

(c) State law requires that $\max_{0<x<l} w(x) < 1/300$. Does this beam meet state code?

11.4 The Nonlinear Shooting Method

The shooting technique for the nonlinear second-order boundary-value problem

$$y'' = f(x, y, y'), \qquad \text{for } a \le x \le b, \quad \text{where} \quad y(a) = \alpha \quad \text{and} \quad y(b) = \beta, \tag{11.6}$$

is similar to the linear shooting method, except that the solution to a nonlinear problem cannot be expressed as a linear combination of the solutions to two initial-value problems. Instead, we approximate the solution to the boundary-value problem by using the solutions to a sequence of initial-value problems involving a parameter t. These problems have the form

$$y'' = f(x, y, y'), \quad \text{for } a \le x \le b, \quad \text{where} \quad y(a) = \alpha \quad \text{and} \quad y'(a) = t. \tag{11.7}$$

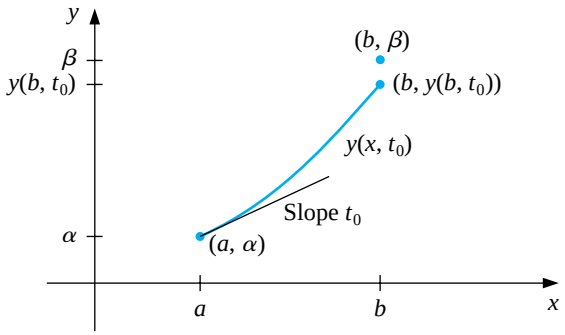
We do this by choosing the parameters $t = t_k$ in a manner to ensure that

$$\lim_{k \to \infty} y(b, t_k) = y(b) = \beta,$$

where $y(x, t_k)$ denotes the solution to the initial-value problem (11.7) with $t = t_k$ and $y(x)$ denotes the solution to the boundary-value problem (11.6).

This technique is called a *shooting* method, by analogy to the procedure of firing objects at a stationary target.(See Figure 11.2.)

Figure 11.2

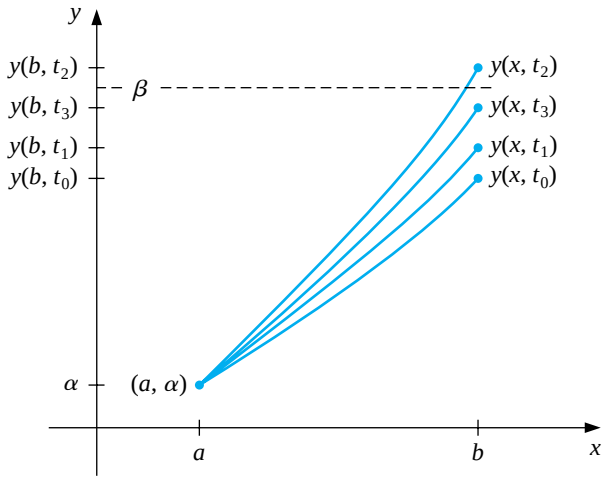


We start with a parameter t_0 that determines the initial elevation at which the object is fired from the point (a, α) along the curve described by the solution to the initial-value problem:

$$y'' = f(x, y, y'), \quad \text{for } a \le x \le b, \quad \text{where} \quad y(a) = \alpha \quad \text{and} \quad y'(a) = t_0.$$

If $y(b, t_0)$ is not sufficiently close to β, we correct our approximation by choosing elevations t_1, t_2, and so on, until $y(b, t_k)$ is sufficiently close to "hitting" β. (See Figure 11.3.)

Figure 11.3



The problem is to determine the parameter t in the initial-value problem so that

$$y(b,t) - \beta = 0.$$

Since this is a nonlinear equation of the type considered in Chapter 2, a number of methods are available. To employ the Secant method to solve the problem, we choose initial approximations t_0 and t_1 to t and then generate the remaining terms of the sequence by using the following procedure.

[Secant Method Solution] Suppose that t_0 and t_1 are initial approximations to the parameter t that solves the nonlinear equation $y(b,t) - \beta = 0$. For each successive $k = 2, 3, \ldots$, solve the initial-value problem

$$y'' = f(x, y, y'), \qquad \text{for } a \le x \le b, \quad \text{where} \quad y(a) = \alpha,$$

with $y'(a) = t_{k-2}$ to find $y(b, t_{k-2})$ and with $y'(a) = t_{k-1}$ to find $y(b, t_{k-1})$. Define

$$t_k = t_{k-1} - \frac{(y(b,t_{k-1}) - \beta)(t_{k-1} - t_{k-2})}{y(b,t_{k-1}) - y(b,t_{k-2})}.$$

Then repeat the process with t_{k-1} replacing t_{k-2} and t_k replacing t_{k-1}.

To use the more powerful Newton's method to generate the sequence $\{t_k\}$, only one initial approximation, t_0, is needed. However, the iteration has the form

$$t_k = t_{k-1} - \frac{y(b,t_{k-1}) - \beta}{(dy/dt)(b,t_{k-1})},$$

and requires the knowledge of $(dy/dt)(b, t_{k-1})$. This presents a difficulty, since an explicit representation for $y(b,t)$ is not known; we know only the values $y(b,t_0)$, $y(b,t_1), \ldots, y(b,t_{k-1})$.

To overcome this difficulty, first rewrite the initial-value problem, emphasizing that the solution depends on both x and t:

$$y''(x,t) = f(x, y(x,t), y'(x,t)), \quad \text{for } a \le x \le b, \text{ where } y(a,t) = \alpha \text{ and } y'(a,t) = t,$$

retaining the prime notation to indicate differentiation with respect to x. Since we are interested in determining $(dy/dt)(b,t)$ when $t = t_{k-1}$, we take the partial derivative with respect to t. This implies that

$$\begin{aligned}\frac{\partial y''}{\partial t}(x,t) &= \frac{\partial f}{\partial t}(x, y(x,t), y'(x,t)) \\ &= \frac{\partial f}{\partial x}(x, y(x,t), y'(x,t))\frac{\partial x}{\partial t} + \frac{\partial f}{\partial y}(x, y(x,t), y'(x,t))\frac{\partial y}{\partial t}(x,t) \\ &\quad + \frac{\partial f}{\partial y'}(x, y(x,t), y'(x,t))\frac{\partial y'}{\partial t}(x,t).\end{aligned}$$

But x and t are independent, so $\dfrac{\partial x}{\partial t} = 0$ and we have

$$\frac{\partial y''}{\partial t}(x,t) = \frac{\partial f}{\partial y}\left(x, y(x,t), y'(x,t)\right)\frac{\partial y}{\partial t}(x,t) + \frac{\partial f}{\partial y'}\left(x, y(x,t), y'(x,t)\right)\frac{\partial y'}{\partial t}(x,t) \tag{11.8}$$

for $a \le x \le b$. The initial conditions give

$$\frac{\partial y}{\partial t}(a,t) = 0 \quad \text{and} \quad \frac{\partial y'}{\partial t}(a,t) = 1.$$

If we simplify the notation by using $z(x,t)$ to denote $(\partial y/\partial t)(x,t)$ and assume that the order of differentiation of x and t can be reversed, Eq. (11.8) becomes the linear initial-value problem

$$z''(x,t) = \frac{\partial f}{\partial y}(x, y, y')z(x,t) + \frac{\partial f}{\partial y'}(x, y, y')z'(x,t), \qquad \text{for } a \le x \le b,$$

where $z(a,t) = 0$ and $z'(a,t) = 1$. Newton's method therefore requires that two initial-value problems be solved for each iteration.

[Newton's Method Solution] Suppose that t_0 is an initial approximation to the parameter t that solves the nonlinear equation $y(b,t) - \beta = 0$. For each successive $k = 1, 2, \ldots$, solve the initial-value problems

$$y'' = f(x, y, y'), \qquad \text{for } a \le x \le b, \quad \text{where} \quad y(a) = \alpha \quad \text{and} \quad y'(a) = t_{k-1}$$

and

$$\begin{aligned} z'' &= f_y(x, y, y')z + f_{y'}(x, y, y')z', \quad \text{for } a \le x \le b, \\ &\text{where} \quad z(a,t) = 0 \quad \text{and} z'(a,t) = 1. \end{aligned}$$

Then define

$$t_k = t_{k-1} - \frac{y(b, t_{k-1}) - \beta}{z(b, t_{k-1})}$$

and repeat the process with t_k replacing t_{k-1}.

In practice, none of these initial-value problems is likely to be solved exactly; instead the solutions are approximated by one of the methods discussed in Chapter 5. The program NLINS113 uses the Runge-Kutta method of order 4 to approximate both solutions required for Newton's method.

EXAMPLE 1 Consider the boundary-value problem

$$y'' = \frac{1}{8}(32 + 2x^3 - yy'), \qquad \text{for } 1 \le x \le 3, \quad \text{where} \quad y(1) = 17 \quad \text{and} \quad y(3) = \frac{43}{3},$$

which has the exact solution $y(x) = x^2 + 16/x$.

Applying the Shooting method to this problem requires approximating the solutions to the initial-value problems

$$y'' = \frac{1}{8}(32 + 2x^3 - yy'), \quad \text{for } 1 \le x \le 3, \quad \text{where} \quad y(1) = 17 \quad \text{and} \quad y'(1) = t_k,$$

and

$$z'' = \frac{\partial f}{\partial y}z + \frac{\partial f}{\partial y'}z' = -\frac{1}{8}(y'z + yz'), \quad \text{for } 1 \le x \le 3, \quad \text{where } z(1) = 0 \text{ and } z'(1) = 1,$$

at each step in the iteration.

We will use the Runge-Kutta method of order 4 within Maple to solve both differential equations. We write the second-order equation as a system of two first-order differential equations, with

```
>sys1:=D(u1)(x)-u2(x),Du2(x)=(32+2*x^3-u1(x)*u2(x))/8;
```

giving

$$sys1 := D(u1)(x) = u2(x), D(u2)(x) = 4 + \frac{1}{4}x^3 - \frac{1}{8}U1(x)U2(x)$$

Define a, b, N, α, and β by

```
>a:=1; b:=3; N:=20; h:=(b-a)/N; alpha:=17; beta:=43/3;
```

For the initial value of t_0 we use

```
>tk:=(beta-alpha)/(b-a);
```

and define the initial conditions by

```
>init1:=u1(1)=alpha,u2(1)=tk;
```

The Runge-Kutta method of order 4 for solving this system is invoked by

```
>g1:=dsolve({sys1,init1},numeric, method=classical[rk4],{u1(x),u2(x)},stepsize=h);
```

We use the values from the Runge-Kutta method to solve the second initial-value problem written as a system of two first-order differential equations, and set up the system as follows:

```
>fy:=(x,v1,v2)->-v2/8;
>fyp:=(x,v1,v2)->-v1/8;
>u1:=0;
>u2:=1;
```

Since we need to use the approximations to y and y_1 at t_i, the Runge-Kutta method cannot be called from Maple. We must generate our own code for the computation. We will include only the values from $z(x_i, t_0)$.

```
> for i from 1 to N do
> x:=a+(i-1)*h;
> k11:=h*u2;
> k12:=h*fy(x,rhs(g1(x)[2]),rhs(g1(x)[3]))*u1+h*fyp(x,rhs(g1(x)[2]),rhs(g1(x)[3]))*u2;
> k21:=h*(u2+k12/2);
> k22:=h*fy(x+h/2,rhs(g1(x)[2]),rhs(g1(x)[3]))*(u1+k11/2)+h*fyp(x+h/2,rhs(g1(x)[2]),rhs
> k31:=h*(u2+k22/2);
> k32:=h*fy(x+h/2,rhs(g1(x)[2]),rhs(g1(x)[3]))*(u1+k21/2)+h*fyp(x+h/2,rhs(g1(x)[2]),rhs
> k41:=h*(u2+k32);
> k42:=h*fy(x+h,rhs(g1(x)[2]),rhs(g1(x)[3]))*(u1+k31)+h*fyp(x+h,rhs(g1(x)[2]),rhs(g1(x
> uu1:=u1+(k11+2*k21+2*k31+k41)/6;
> uu2:=u2+(k12+2*k22+2*k32+k42)/6;
> u1:=uu1;
> u2:=uu2;
> od;
```

We make our test for convergence on $|y(b, t_0) - \beta|$ with

```
>abs(rhs(g1(b)[2])-beta);
```

which gives 6.14586912. This is not sufficient, and we need at least one more iteration. So we compute $t_1 = -16.20583517$ with

```
>tk1:=tk-(rhs(g1(b)[2])-beta)/u1;
```

and repeat the entire process.

If the stopping technique requires $|w_{1,N}(t_k) - y(3)| \leq 10^{-5}$, this problem takes four iterations and uses $t_4 = -14.000203$. The results obtained for this value of t are shown in Table 11.4. □

Table 11.4

x_i	$w_{1,i}$	$y(x_i)$	$\lvert w_{1,i} - y(x_i)\rvert$	x_i	$w_{1,i}$	$y(x_i)$	$\lvert w_{1,i} - y(x_i)\rvert$
1.0	17.000000	17.000000		2.0	12.000023	12.000000	2.32×10^{-5}
1.1	15.755495	15.755455	4.06×10^{-5}	2.1	12.029066	12.029048	1.84×10^{-5}
1.2	14.773389	14.773333	5.60×10^{-5}	2.2	12.112741	12.112727	1.40×10^{-5}
1.3	13.997752	13.997692	5.94×10^{-5}	2.3	12.246532	12.246522	1.01×10^{-5}
1.4	13.388629	13.388571	5.71×10^{-5}	2.4	12.426673	12.426667	6.68×10^{-6}
1.5	12.916719	12.916667	5.23×10^{-5}	2.5	12.650004	12.650000	3.61×10^{-6}
1.6	12.560046	12.560000	4.64×10^{-5}	2.6	12.913847	12.913845	9.17×10^{-7}
1.7	12.301805	12.301765	4.02×10^{-5}	2.7	13.215924	13.215926	1.43×10^{-6}
1.8	12.128923	12.128889	3.14×10^{-5}	2.8	13.554282	13.554286	3.46×10^{-6}
1.9	12.031081	12.031053	2.84×10^{-5}	2.9	13.927236	13.927241	5.21×10^{-6}
				3.0	14.333327	14.333333	6.69×10^{-6}

Although Newton's method used with the shooting technique requires the solution of an additional initial-value problem, it will generally be faster than the Secant method. Both methods are only locally convergent, since they require good initial approximations.

EXERCISE SET 11.4

1. Use the Nonlinear Shooting method with $h = 0.5$ to approximate the solution to the boundary-value problem

   $$y'' = -(y')^2 - y + \ln x, \qquad \text{for } 1 \le x \le 2 \quad \text{with} \quad y(1) = 0 \quad \text{and} \quad y(2) = \ln 2.$$

   Compare your results to the actual solution $y = \ln x$.

2. Use the Nonlinear Shooting method with $h = 0.25$ to approximate the solution to the boundary-value problem

   $$y'' = 2y^3, \qquad \text{for } -1 \le x \le 0 \quad \text{with } y(-1) = \frac{1}{2} \text{ and } y(0) = \frac{1}{3}.$$

   Compare your results to the actual solution $y(x) = 1/(x + 3)$.

3. Use the Nonlinear Shooting method to approximate the solution to the following boundary-value problems, iterating until $|w_{1,n} - \beta| \le 10^{-4}$. The actual solution is given for comparison to your results.

   (a) $y'' = y^3 - yy'$, for $1 \le x \le 2$ with $y(1) = \frac{1}{2}$ and $y(2) = \frac{1}{3}$; use $h = 0.1$ and compare the results to $y(x) = (x + 1)^{-1}$.

   (b) $y'' = 2y^3 - 6y - 2x^3$, for $1 \le x \le 2$ with $y(1) = 2$ and $y(2) = \frac{5}{2}$; use $h = 0.1$ and compare the results to $y(x) = x + x^{-1}$.

   (c) $y'' = y' + 2(y - \ln x)^3 - x^{-1}$, for $2 \le x \le 3$ with $y(2) = \frac{1}{2} + \ln 2$ and $y(3) = \frac{1}{3} + \ln 3$; use $h = 0.1$ and compare the results to $y(x) = x^{-1} + \ln x$.

   (d) $y'' = \left[x^2(y')^2 - 9y^2 + 4x^6\right]/x^5$, for $1 \le x \le 2$ with $y(1) = 0$ and $y(2) = \ln 256$; use $h = 0.05$ and compare the results to $y(x) = x^3 \ln x$.

4. Use the Secant method with $t_0 = (\beta - \alpha)/(b - a)$ and $t_1 = t_0 + (\beta - y(b, t_0))/(b - a)$ to solve the problems in Exercises 3(a) and 3(c) and compare the number of iterations required with that of Newton's method.

5. The Van der Pol equation,

   $$y'' - \mu(y^2 - 1)y' + y = 0, \qquad \text{for } \mu > 0,$$

   governs the flow of current in a vacuum tube with three internal elements. Let $\mu = \frac{1}{2}$, $y(0) = 0$, and $y(2) = 1$. Approximate the solution $y(t)$ for $t = 0.2i$, where $1 \le i \le 9$.

11.5 Nonlinear Finite-Difference Methods

The difference method for the general nonlinear boundary-value problem

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

is similar to the method applied to linear problems in Section 11.3. Here, however, the system of equations will not be linear, so an iterative process is required to solve it.

As in the linear case, we divide $[a, b]$ into $(N + 1)$ equal subintervals whose endpoints are at $x_i = a + ih$ for $i = 0, 1, \ldots, N + 1$. Assuming that the exact solution has a bounded fourth derivative allows us to replace $y''(x_i)$ and $y'(x_i)$ in each of the equations by the appropriate centered-difference formula to obtain, for each $i = 1, 2, \ldots, N$,

$$\frac{y(x_{i+1}) - 2y(x_i) + y(x_{i-1})}{h^2} = f\left(x_i, y(x_i), \frac{y(x_{i+1}) - y(x_{i-1})}{2h} - \frac{h^2}{6}y'''(\eta_i)\right) + \frac{h^2}{12}y^{(4)}(\xi_i),$$

for some ξ_i and η_i in the interval (x_{i-i}, x_{i+1}).

The difference method results when the error terms are deleted and the boundary conditions are added. This produces the $N \times N$ nonlinear system

$$\begin{aligned}
2w_1 - w_2 + h^2 f\left(x_1, w_1, \frac{w_2 - \alpha}{2h}\right) - \alpha &= 0,\\
-w_1 + 2w_2 - w_3 + h^2 f\left(x_2, w_2, \frac{w_3 - w_1}{2h}\right) &= 0,\\
&\vdots\\
-w_{N-2} + 2w_{N-1} - w_N + h^2 f\left(x_{N-1}, w_{N-1}, \frac{w_N - w_{N-2}}{2h}\right) &= 0,\\
-w_{N-1} + 2w_N + h^2 f\left(x_N, w_N, \frac{\beta - w_{N-1}}{2h}\right) - \beta &= 0.
\end{aligned}$$

To approximate the solution to this system, we use Newton's method for nonlinear systems, as discussed in Section 10.2. A sequence of iterates $\{(w_1^{(k)}, w_2^{(k)}, \ldots, w_N^{(k)})^t\}$ is generated that converges to the solution of system, provided that the initial approximation $(w_1^{(0)}, w_2^{(0)}, \ldots, w_N^{(0)})^t$ is sufficiently close to the true solution, $(w_1, w_2, \ldots, w_N)^t$.

Newton's method for nonlinear systems requires solving, at each iteration, an $N \times N$ linear system involving the Jacobian matrix. In our case, the Jacobian matrix is tridiagonal, and Crout factorization can be applied. The initial approximations $w_i^{(0)}$ to w_i for each $i = 1, 2, \ldots, N$, are obtained by passing a straight line through (a, α) and (b, β) and evaluating at x_i.

Since a good initial approximation may be required, an upper bound for k should be specified and, if exceeded, a new initial approximation or a reduction in step size considered.

The program NLFDM114 can be used to employ the Nonlinear Finite-Difference method.

EXAMPLE 1 We apply the Nonlinear Finite-Difference method, with $h = 0.1$, to the nonlinear boundary-value problem

$$y'' = \frac{1}{8}\left(32 + 2x^3 - yy'\right), \qquad \text{for } 1 \le x \le 3, \quad \text{where } y(1) = 17 \text{ and } y(3) = \frac{43}{3}.$$

To use Maple, we first access the linear algebra library

```
>with(linalg);
```

and then define a, b, α, β, N, h, and $\mathbf{w} = \mathbf{0}$ by

```
>a:=1; b:=3; alpha:=17; beta:=43/3; N:=19; h:=(b-a)/(N+1);
>w:=vector(19,0);
```

We define x_i and initialize w_i by passing a straight line through (a, α) and (b, β) and evaluating at x_i. We also define and initialize the 19×19 matrix A as the zero matrix and $\mathbf{u}$ as the 19-dimensional zero vector.

```
>for i from 1 to N do
>x[i]:=a+i*h;
>w[i]:=alpha+i*(beta-alpha)/(b-a)*h;
>od;
>A:=matrix(19,19,0);
>u:=vector(19,0);
```

The functions $f(x, y, y')$, $f_y(x, y, y')$, and $f_{y'}(x, y, y')$ are defined by

```
>f:=(x,y,yp)->(32+2*x^3-y*yp)/8;
>fy:=(x,y,yp)->-yp/8;
>fyp:=(x,y,yp)->-y/8;
```

The nonzero entries of A and the right-hand side, u, of the linear system are generated as follows:

```
>A[1,1]:=2+h*h*evalf(fy(x[1],w[1],(w[2]-alpha)/(2*h)));
>A[1,2]:=-1+h*evalf(fyp(x[1],w[1],(w[2]-alpha)/(2*h)))/2;
>u[1]:=-(2*w[1]-w[2]-alpha+h*h*evalf(f(x[1],w[1],
 (w[2]-alpha)/(2*h))));
>for i from 2 to N-1 do
>A[i,i-1]:=-1-h*evalf(fyp(x[i],w[i],(w[i+1]-w[i-1])/(2*h)))/2;
>A[i,i+1]:=-1+h*evalf(fyp(x[i],w[i],(w[i+1]-w[i-1])/(2*h)))/2;
>A[i,i]:=2+h*h*evalf(fy(x[i],w[i],(w[i+1]-w[i-1])/(2*h)));

>u[i]:=-(-w[i-1]+2*w[i]-w[i+1]+h*h*evalf(f(x[i],w[i],
 (w[i+1]-w[i-1])/(2*h))));
>od;
```

```
>A[N,N-1]:=-1-h*evalf(fyp(x[N],w[N],(beta-w[N-1])/(2*h)))/2;
>A[N,N]:=2+h*h*evalf(fy(x[N],w[N],(beta-w[N-1])/(2*h)));
>u[N]:=-(-w[N-1]+2*w[N]-beta+h*h*evalf(f(x[N],w[N],
 (beta-w[N-1])/(2*h))));
```

The augmented matrix for the linear system is formed by

```
>B:=augment(A,u);
```

Then we use the Gaussian elimination of Maple with the command

```
>M:=gausselim(B);
```

and obtain the solution to the linear system with the command

```
>v:=backsub(M);
```

The first iteration for $w(x_i)$ is given by

```
>z:=w+v;
>w:=evalm(z);
```

We continue to iterate the process until values of successive iterates differ by less than 10^{-8}. This is accomplished with four iterations and produces the results in Table 11.5. The problem in this example is the same as that considered for the Nonlinear Shooting method, Example 1 of Section 11.4. □

Table 11.5

x_i	w_i	$y(x_i)$	$\lvert w_i - y(x_i)\rvert$	x_i	w_i	$y(x_i)$	$\lvert w_i - y(x_i)\rvert$
1.0	17.000000	17.000000		2.0	11.997915	12.000000	2.085×10^{-3}
1.1	15.754503	15.755455	9.520×10^{-4}	2.1	12.027142	12.029048	1.905×10^{-3}
1.2	14.771740	14.773333	1.594×10^{-3}	2.2	12.111020	12.112727	1.707×10^{-3}
1.3	13.995677	13.997692	2.015×10^{-3}	2.3	12.245025	12.246522	1.497×10^{-3}
1.4	13.386297	13.388571	2.275×10^{-3}	2.4	12.425388	12.426667	1.278×10^{-3}
1.5	12.914252	12.916667	2.414×10^{-3}	2.5	12.648944	12.650000	1.056×10^{-3}
1.6	12.557538	12.560000	2.462×10^{-3}	2.6	12.913013	12.913846	8.335×10^{-4}
1.7	12.299326	12.301765	2.438×10^{-3}	2.7	13.215312	13.215926	6.142×10^{-4}
1.8	12.126529	12.128889	2.360×10^{-3}	2.8	13.553885	13.554286	4.006×10^{-4}
1.9	12.028814	12.031053	2.239×10^{-3}	2.9	13.927046	13.927241	1.953×10^{-4}
				3.0	14.333333	14.333333	

Richardson's extrapolation can also be used for the Nonlinear Finite-Difference method. Table 11.6 lists the results when this method is applied to Example 1

using $h = 0.1, 0.05$, and 0.025, with four iterations in each case. The notation is the same as in Example 2 of Section 11.3, and the values of Ext_{3i} are all accurate to the places listed, with an actual maximum error of 3.68×10^{-10}. The values of $w_i(h = 0.1)$ are omitted from the table since they were listed previously.

Table 11.6

x_i	$w_i(h = 0.05)$	$w_i(h = 0.025)$	Ext_{1i}	Ext_{2i}	Ext_{3i}
1.0	17.00000000	17.00000000	17.00000000	17.00000000	17.00000000
1.1	15.75521721	15.75539525	15.75545543	15.75545460	15.75545455
1.2	14.77293601	14.77323407	14.77333479	14.77333342	14.77333333
1.3	13.99718996	13.99756690	13.99769413	13.99769242	13.99769231
1.4	13.38800424	13.38842973	13.38857346	13.38857156	13.38857143
1.5	12.91606471	12.91651628	12.91666881	12.91666680	12.91666667
1.6	12.55938618	12.55984665	12.56000217	12.56000014	12.56000000
1.7	12.30115670	12.30161280	12.30176684	12.30176484	12.30176471
1.8	12.12830042	12.12874287	12.12899094	12.12888902	12.12888889
1.9	12.03049438	12.03091316	12.03105457	12.03105275	12.03105263
2.0	11.99948020	11.99987013	12.00000179	12.00000011	12.00000000
2.1	12.02857252	12.02892892	12.02902924	12.02904772	12.02904762
2.2	12.11230149	12.11262089	12.11272872	12.11272736	12.11272727
2.3	12.24614846	12.24642848	12.24652299	12.24652182	12.24652174
2.4	12.42634789	12.42658702	12.42666773	12.42666673	12.42666667
2.5	12.64973666	12.64993420	12.65000086	12.65000005	12.65000000
2.6	12.91362828	12.91379422	12.91384683	12.91384620	12.91384615
2.7	13.21577275	13.21588765	13.21592641	13.21592596	13.21592593
2.8	13.55418579	13.55426075	13.55428603	13.55428573	13.55428571
2.9	13.92719268	13.92722921	13.92724153	13.92724139	13.92724138
3.0	14.33333333	14.33333333	14.33333333	14.33333333	14.33333333

EXERCISE SET 11.5

1. Use the Nonlinear Finite-Difference method with $h = 0.5$ to approximate the solution to the boundary-value problem

   $$y'' = -(y')^2 - y + \ln x, \qquad \text{for } 1 \le x \le 2 \quad \text{with} \quad y(1) = 0 \quad \text{and} \quad y(2) = \ln 2.$$

   Compare your results to the actual solution $y = \ln x$.

2. Use the Nonlinear Finite-Difference method with $h = 0.25$ to approximate the solution to the boundary-value problem

   $$y'' = 2y^3, \qquad \text{for } -1 \le x \le 0 \quad \text{with} \quad y(-1) = \tfrac{1}{2} \quad \text{and} \quad y(0) = \tfrac{1}{3}.$$

   Compare your results to the actual solution $y(x) = 1/(x + 3)$.

3. Use the Nonlinear Finite-Difference method to approximate the solution to the following boundary-value problems, iterating until successive iterations differ by less than 10^{-4}. The actual solution is given for comparison to your results.

   (a) $y'' = y^3 - yy'$, for $1 \le x \le 2$ with $y(1) = \frac{1}{2}$ and $y(2) = \frac{1}{3}$; use $h = 0.1$ and compare the results to $y(x) = (x + 1)^{-1}$.

   (b) $y'' = 2y^3 - 6y - 2x^3$, for $1 \le x \le 2$ with $y(1) = 2$ and $y(2) = \frac{5}{2}$; use $h = 0.1$ and compare the results to $y(x) = x + x^{-1}$.

   (c) $y'' = y' + 2(y - \ln x)^3 - x^{-1}$, for $2 \le x \le 3$ with $y(2) = \frac{1}{2} + \ln 2$ and $y(3) = \frac{1}{3} + \ln 3$; use $h = 0.1$ and compare the results to $y(x) = x^{-1} + \ln x$.

   (d) $y'' = (x^2(y')^2 - 9y^2 + 4x^6)/x^5$, for $1 \le x \le 2$ with $y(1) = 0$ and $y(2) = \ln 256$; use $h = 0.05$ and compare the results to $y(x) = x^3 \ln x$.

4. Repeat Exercise 3(a) and (b) using extrapolation.

5. In Exercise 8 of Section 11.3 the deflection of beam with supported ends subject to uniform loading was approximated. Using a more appropriate representation of curvature gives the differential equation

   $$\left[1 + (w'(x))^2\right]^{-3/2} w''(x) = \frac{S}{EI} w(x) + \frac{q}{2EI} x(x - l), \qquad \text{for } 0 < x < l.$$

   Approximate the deflection $w(x)$ of the beam every 6 in. and compare the results to those of Exercise 8 of Section 11.3.

11.6 Variational Techniques

The Shooting method for approximating the solution to a boundary-value problem replaced it with initial-value problems. The finite-difference method replaced the continuous operation of differentiation with the discrete operation of finite differences. The Rayleigh-Ritz method is a variational technique that attacks the problem from a third approach. The boundary-value problem is first reformulated as a problem of choosing, from the set of all sufficiently differentiable functions satisfying the boundary conditions, the function to minimize a certain integral. Then the set of feasible functions is reduced in size and a function in this reduced set is found that minimizes the integral. This gives an approximation to the solution to the original minimization problem and, as a consequence, an approximation to the solution to the boundary-value problem.

To describe the Rayleigh-Ritz method, we consider approximating the solution to a linear two-point boundary-value problem from beam-stress analysis. This boundary-value problem is described by the differential equation

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

with the boundary conditions

$$y(0) = y(1) = 0.$$

This differential equation describes the deflection $y(x)$ of a beam of length 1 with variable cross section given by $q(x)$. The deflection is due to the added stresses $p(x)$ and $f(x)$.

As is the case with many boundary-value problems that describe physical phenomena, the solution to the beam equation satisfies a *variational property*. The solution to the beam equation is the function that minimizes a certain integral over all functions in the set $C_0^2[0,1]$, where we define

$$C_0^2[0,1] = \left\{u \in C^2[0,1] \;\mid\; u(0) = u(1) = 0.\right\}$$

Details concerning this connection can be found in *Spline Analysis* by Schultz [Schu], pp. 88–89.

[Variational Property for the Beam Equation] The function $y \in C_0^2[0,1]$ is the unique solution to the boundary-value problem

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

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

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

The Rayleigh-Ritz method approximates the solution y by minimizing the integral, not over all the functions in $C_0^2[0,1]$, but over a smaller set of functions consisting of linear combinations of certain basis functions $\phi_1, \phi_2, \ldots, \phi_n$. The basis functions are chosen to be linearly independent and to satisfy

$$\phi_i(0) = \phi_i(1) = 0, \qquad \text{for each } i = 1, 2, \ldots, n.$$

An approximation $\phi(x) = \sum_{i=1}^{n} c_i\phi_i(x)$ to the solution $y(x)$ is obtained by finding constants $c_1, c_2, \ldots, c_n$ to minimize $I[\phi(x)] = I[\sum_{i=1}^{n} c_i\phi_i(x)]$.

From the variational property,

$$\begin{aligned} I[\phi(x)] &= I\left[\sum_{i=1}^{n} c_i\phi_i(x)\right] \\ &= \int_0^1 \left[p(x)\left[\sum_{i=1}^{n} c_i\phi_i'(x)\right]^2 + q(x)\left[\sum_{i=1}^{n} c_i\phi_i(x)\right]^2 - 2f(x)\sum_{i=1}^{n} c_i\phi_i(x)\right] dx, \end{aligned}$$

and, for a minimum to occur, it is necessary to have

$$\frac{\partial I}{\partial c_j}[\phi(x)] = 0, \qquad \text{for each } j = 1, 2, \ldots, n.$$

Differentiating with respect to the coefficients gives

$$\frac{\partial I}{\partial c_j}[\phi(x)] = \int_0^1 \left[2p(x)\sum_{i=1}^{n} c_i\phi_i'(x)\phi_j'(x) + 2q(x)\sum_{i=1}^{n} c_i\phi_i(x)\phi_j(x) - 2f(x)\phi_j(x)\right] dx,$$

so

$$0 = \sum_{i=1}^{n}\left[\int_0^1 \{p(x)\phi_i'(x)\phi_j'(x) + q(x)\phi_i(x)\phi_j(x)\}\, dx\right] c_i - \int_0^1 f(x)\phi_j(x)\, dx,$$

for each $j = 1, 2, \ldots, n$. These *normal equations* produce an $n \times n$ linear system $A\mathbf{c} = \mathbf{b}$ in the variables $c_1, c_2, \ldots, c_n$, where the symmetric matrix A is given by

$$a_{ij} = \int_0^1 [p(x)\phi_i'(x)\phi_j'(x) + q(x)\phi_i(x)\phi_j(x)]\, dx,$$

and the vector $\mathbf{b}$ has the coordinates

$$b_i = \int_0^1 f(x)\phi_i(x)\, dx.$$

The most elementary choice of basis functions involves piecewise linear polynomials. The first step is to form a partition of $[0,1]$ by choosing points $x_0, x_1, \ldots, x_{n+1}$ with

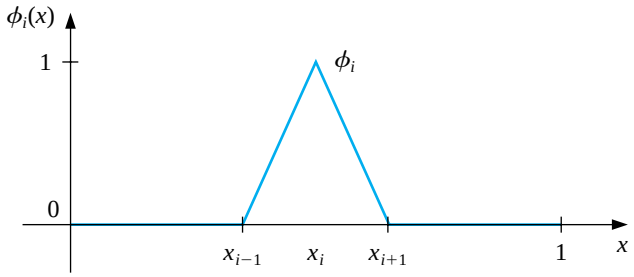
$$0 = x_0 < x_1 < \cdots < x_n < x_{n+1} = 1.$$

Let $h_i = x_{i+1} - x_i$ for each $i = 0, 1, \ldots, n$, and define the basis functions $\phi_1(x), \phi_2(x), \ldots, \phi_n(x)$ by

$$\phi_i(x) = \begin{cases} 0, & \text{for } 0 \le x \le x_{i-1}, \\ \dfrac{1}{h_{i-1}}(x - x_{i-1}), & \text{for } x_{i-1} < x \le x_i, \\ \dfrac{1}{h_i}(x_{i+1} - x), & \text{for } x_i < x \le x_{i+1}, \\ 0, & \text{for } x_{i+1} < x \le 1, \end{cases} \tag{11.9}$$

for each $i = 1, 2, \ldots, n$. (See Figure 11.4.)

Figure 11.4



Since the functions ϕ_i are piecewise linear, the derivatives ϕ_i', while not continuous, are constant on the open subinterval (x_j, x_{j+1}) for each $j = 0, 1, \ldots, n$. Thus, we have

$$\phi_i'(x) = \begin{cases} 0, & \text{for } 0 < x < x_{i-1}, \\ \dfrac{1}{h_{i-1}}, & \text{for } x_{i-1} < x < x_i, \\ -\dfrac{1}{h_i}, & \text{for } x_i < x < x_{i+1}, \\ 0, & \text{for } x_{i+1} < x < 1, \end{cases}$$

for each $i = 1, 2, \ldots, n$. Because ϕ_i and ϕ_i' are nonzero only on (x_{i-1}, x_{i+1}),

$$\phi_i(x)\phi_j(x) \equiv 0 \quad \text{and} \quad \phi_i'(x)\phi_j'(x) \equiv 0,$$

except when j is $i-1$, i, or $i+1$. As a consequence, the linear system reduces to an $n \times n$ tridiagonal linear system. The nonzero entries in A are

$$\begin{aligned} a_{ii} &= \int_0^1 \{p(x)[\phi_i'(x)]^2 + q(x)[\phi_i(x)]^2\}\,dx \\ &= \int_{x_{i-1}}^{x_i} \left(\frac{1}{h_{i-1}}\right)^2 p(x)\,dx + \int_{x_i}^{x_{i+1}} \left(\frac{-1}{h_i}\right)^2 p(x)\,dx \\ &\quad + \int_{x_{i-1}}^{x_i} \left(\frac{1}{h_{i-1}}\right)^2 (x - x_{i-1})^2 q(x)\,dx + \int_{x_i}^{x_{i+1}} \left(\frac{1}{h_i}\right)^2 (x_{i+1} - x)^2 q(x)\,dx, \end{aligned}$$

for each $i = 1, 2, \ldots, n$;

$$\begin{aligned} a_{i,i+1} &= \int_0^1 \{p(x)\phi_i'(x)\phi_{i+1}'(x) + q(x)\phi_i(x)\phi_{i+1}(x)\}\,dx \\ &= \int_{x_i}^{x_{i+1}} -\left(\frac{1}{h_i}\right)^2 p(x)\,dx + \int_{x_i}^{x_{i+1}} \left(\frac{1}{h_i}\right)^2 (x_{i+1} - x)(x - x_i)q(x)\,dx, \end{aligned}$$

for each $i = 1, 2, \ldots, n-1$; and

$$\begin{aligned} a_{i,i-1} &= \int_0^1 \{p(x)\phi_i'(x)\phi_{i-1}'(x) + q(x)\phi_i(x)\phi_{i-1}(x)\}\,dx \\ &= \int_{x_{i-1}}^{x_i} -\left(\frac{1}{h_{i-1}}\right)^2 p(x)\,dx + \int_{x_{i-1}}^{x_i} \left(\frac{1}{h_{i-1}}\right)^2 (x_i - x)(x - x_{i-1})q(x)\,dx, \end{aligned}$$

for each $i = 2, \ldots, n$. The entries in $\mathbf{b}$ are

$$\begin{aligned} b_i &= \int_0^1 f(x)\phi_i(x)\,dx \\ &= \int_{x_{i-1}}^{x_i} \frac{1}{h_{i-1}}(x - x_{i-1})f(x)\,dx + \int_{x_i}^{x_{i+1}} \frac{1}{h_i}(x_{i+1} - x)f(x)\,dx, \end{aligned}$$

for each $i = 1, 2, \ldots, n$.

There are six types of integrals to be evaluated

$$Q_{1,i} = \left(\frac{1}{h_i}\right)^2 \int_{x_i}^{x_{i+1}} (x_{i+1} - x)(x - x_i)q(x)\,dx, \qquad \text{for each } i = 1, 2, \ldots, n-1,$$

$$Q_{2,i} = \left(\frac{1}{h_{i-1}}\right)^2 \int_{x_{i-1}}^{x_i} (x - x_{i-1})^2 q(x)\,dx, \qquad \text{for each } i = 1, 2, \ldots, n,$$

$$Q_{3,i} = \left(\frac{1}{h_i}\right)^2 \int_{x_i}^{x_{i+1}} (x_{i+1} - x)^2 q(x)\,dx, \qquad \text{for each } i = 1, 2, \ldots, n,$$

$$Q_{4,i} = \left(\frac{1}{h_{i-1}}\right)^2 \int_{x_{i-1}}^{x_i} p(x)\,dx, \qquad \text{for each } i = 1, 2, \ldots, n+1,$$

$$Q_{5,i} = \frac{1}{h_{i-1}} \int_{x_{i-1}}^{x_i} (x - x_{i-1})f(x)\,dx, \qquad \text{for each } i = 1, 2, \ldots, n,$$

and

$$Q_{6,i} = \frac{1}{h_i} \int_{x_i}^{x_{i+1}} (x_{i+1} - x)f(x)\,dx, \qquad \text{for each } i = 1, 2, \ldots, n.$$

Once the values of these integrals have been determined we have

$$\begin{aligned} a_{i,i} &= Q_{4,i} + Q_{4,i+1} + Q_{2,i} + Q_{3,i}, \qquad \text{for each } i = 1, 2, \ldots, n, \\ a_{i,i+1} &= -Q_{4,i+1} + Q_{1,i}, \qquad \text{for each } i = 1, 2, \ldots, n-1, \\ a_{i,i-1} &= -Q_{4,i} + Q_{1,i-1}, \qquad \text{for each } i = 2, 3, \ldots, n, \end{aligned}$$

and

$$b_i = Q_{5,i} + Q_{6,i}, \qquad \text{for each } i = 1, 2, \ldots, n.$$

The entries in $\mathbf{c}$ are the unknown coefficients $c_1, c_2, \ldots, c_n$, from which the Rayleigh-Ritz approximation $\phi(x) = \sum_{i=1}^{n} c_i \phi_i(x)$ is constructed.

A practical difficulty with this method is the necessity of evaluating $6n$ integrals. The integrals can be evaluated either directly or by a quadrature formula such as Simpson's method. An alternative approach for the integral evaluation is to approximate each of the functions p, q, and f with its piecewise linear interpolating polynomial and then integrate the approximation. Consider, for example, the integral $Q_{1,i}$. The piecewise linear interpolation of q is

$$P_q(x) = \sum_{i=0}^{n+1} q(x_i)\phi_i(x),$$

where $\phi_1, \ldots, \phi_n$ are defined in Eq. (11.9) and

$$\phi_0(x) = \begin{cases} \dfrac{x_1 - x}{x_1}, & \text{if } 0 \le x \le x_1, \\ 0, & \text{elsewhere,} \end{cases} \quad \text{and} \quad \phi_{n+1}(x) = \begin{cases} \dfrac{x - x_n}{1 - x_n}, & \text{if } x_n \le x \le 1, \\ 0, & \text{elsewhere.} \end{cases}$$

Since the interval of integration is $[x_i, x_{i+1}]$, the piecewise polynomial $P_q(x)$ reduces to

$$P_q(x) = q(x_i)\phi_i(x) + q(x_{i+1})\phi_{i+1}(x).$$

This is the first-degree interpolating polynomial studied in Section 3.2, with error

$$|q(x) - P_q(x)| = O(h_i^2), \qquad \text{when } x_i \le x \le x_{i+1},$$

if $q \in C^2[x_i, x_{i+1}]$. For $i = 1, 2, \ldots, n-1$, the approximation to $Q_{1,i}$ is obtained by integrating the approximation to the integrand

$$\begin{aligned} Q_{1,i} &= \left(\frac{1}{h_i}\right)^2 \int_{x_i}^{x_{i+1}} (x_{i+1} - x)(x - x_i) q(x)\, dx \\ &\approx \left(\frac{1}{h_i}\right)^2 \int_{x_i}^{x_{i+1}} (x_{i+1} - x)(x - x_i) \left[\frac{q(x_i)(x_{i+1} - x)}{h_i} + \frac{q(x_{i+1})(x - x_i)}{h_i}\right] dx \\ &= \frac{h_i}{12}[q(x_i) + q(x_{i+1})], \end{aligned}$$

with

$$\left| Q_{1,i} - \frac{h_i}{12}\left[q(x_i) + q(x_{i+1})\right] \right| = O(h_i^3).$$

Approximations to the other integrals are derived in a similar manner and given by

$$Q_{2,i} \approx \frac{h_{i-1}}{12}\left[3q(x_i) + q(x_{i-1})\right], \qquad Q_{3,i} \approx \frac{h_i}{12}\left[3q(x_i) + q(x_{i+1})\right],$$

$$Q_{4,i} \approx \frac{h_{i-1}}{2}\left[p(x_i) + p(x_{i-1})\right], \qquad Q_{5,i} \approx \frac{h_{i-1}}{6}\left[2f(x_i) + f(x_{i-1})\right],$$

and

$$Q_{6,i} \approx \frac{h_i}{6}[2f(x_i) + f(x_{i+1})].$$

The program PLRRG115 sets up the tridiagonal linear system and incorporates Crout factorization for tridiagonal systems to solve the system. The integrals $Q_{1,i}, \ldots, Q_{6,i}$ can be computed by one of the methods just discussed. Because of the elementary nature of the following example, the integrals were found directly.

EXAMPLE 1 Consider the boundary-value problem

$$-y'' + \pi^2 y = 2\pi^2 \sin(\pi x), \quad \text{for } 0 \le x \le 1, \quad \text{where } y(0) = y(1) = 0.$$

Let $h_i = h = 0.1$, so that $x_i = 0.1i$ for each $i = 0, 1, \ldots, 9$. The integrals are

$$\begin{aligned}
Q_{1,i} &= 100 \int_{0.1i}^{0.1i+0.1} (0.1i + 0.1 - x)(x - 0.1i)\pi^2 \, dx = \frac{\pi^2}{60}, \\
Q_{2,i} &= 100 \int_{0.1i-0.1}^{0.1i} (x - 0.1i + 0.1)^2 \pi^2 \, dx = \frac{\pi^2}{30}, \\
Q_{3,i} &= 100 \int_{0.1i}^{0.1i+0.1} (0.1i + 0.1 - x)^2 \pi^2 \, dx = \frac{\pi^2}{30}, \\
Q_{4,i} &= 100 \int_{0.1i-0.1}^{0.1i} dx = 10, \\
Q_{5,i} &= 10 \int_{0.1i-0.1}^{0.1i} (x - 0.1i + 0.1) 2\pi^2 \sin \pi x \, dx \\
&= -2\pi \cos 0.1\pi i + 20 \left[\sin(0.1\pi i) - \sin\left((0.1i - 0.1)\pi\right)\right],
\end{aligned}$$

and

$$\begin{aligned}
Q_{6,i} &= 10 \int_{0.1i}^{0.1i+0.1} (0.1i + 0.1 - x) 2\pi^2 \sin \pi x \, dx \\
&= 2\pi \cos 0.1\pi i - 20 \left[\sin\left((0.1i + 0.1)\pi\right) - \sin(0.1\pi i)\right].
\end{aligned}$$

The linear system $A\mathbf{c} = \mathbf{b}$ has

$$\begin{aligned}
a_{i,i} &= 20 + \frac{\pi^2}{15}, \qquad \text{for each } i = 1, 2, \ldots, 9, \\
a_{i,i+1} &= -10 + \frac{\pi^2}{60}, \qquad \text{for each } i = 1, 2, \ldots, 8, \\
a_{i,i-1} &= -10 + \frac{\pi^2}{60}, \qquad \text{for each } i = 2, 3, \ldots, 9,
\end{aligned}$$

and

$$b_i = 40 \sin(0.1\pi i)[1 - \cos 0.1\pi], \qquad \text{for each } i = 1, 2, \ldots, 9.$$

The solution to the tridiagonal linear system is

$$\begin{array}{lllllllll} c_9 &=& 0.3102866742, & c_6 &=& 0.9549641893, & c_3 &=& 0.8123410598, \\ c_8 &=& 0.5902003271, & c_5 &=& 1.004108771, & c_2 &=& 0.5902003271, \\ c_7 &=& 0.8123410598, & c_4 &=& 0.9549641893, & c_1 &=& 0.3102866742, \end{array}$$

and the piecewise linear approximation is

$$\phi(x) = \sum_{i=1}^{9} c_i \phi_i(x).$$

The actual solution to the boundary-value problem is

$$y(x) = \sin \pi x.$$

Table 11.7 lists the error in the approximation at x_i for each $i = 1, \ldots, 9$. □

Table 11.7

i	x_i	$\phi(x_i)$	$y(x_i)$	$\lvert\phi(x_i) - y(x_i)\rvert$
1	0.1	0.3102866742	0.3090169943	0.00127
2	0.2	0.5902003271	0.5877852522	0.00242
3	0.3	0.8123410598	0.8090169943	0.00332
4	0.4	0.9549641896	0.9510565162	0.00391
5	0.5	1.0041087710	1.0000000000	0.00411
6	0.6	0.9549641893	0.9510565162	0.00391
7	0.7	0.8123410598	0.8090169943	0.00332
8	0.8	0.5902003271	0.5877852522	0.00242
9	0.9	0.3102866742	0.3090169943	0.00127

The tridiagonal matrix A given by the piecewise linear basis functions is positive definite, so the linear system is stable with respect to round-off error and

$$|\phi(x) - y(x)| = O(h^2), \qquad \text{when } 0 \le x \le 1.$$

The use of piecewise-linear basis functions results in an approximate solution that is continuous but not differentiable on $[0, 1]$. A more complicated set of basis functions is required to construct an approximation that has two continuous derivatives. These basis functions are similar to the cubic interpolatory splines discussed in Section 3.5.

Recall that the cubic *interpolatory* spline S on the five nodes x_0, x_1, x_2, x_3, and x_4 for a function f is defined as follows:

(a) S is a cubic polynomial, denoted by S_j, on $[x_j, x_{j+1}]$, for $j = 0$, 1, 2, 3. (*This gives* 16 *selectable constants for* S, 4 *for each cubic.*)

(b) $S(x_j) = f(x_j)$, for $j = 0, 1, 2, 3, 4$ (5 *specified conditions*).

(c) $S_{j+1}(x_{j+1}) = S_j(x_{j+1})$, for $j = 0, 1, 2$ (3 *specified conditions*).

(d) $S'_{j+1}(x_{j+1}) = S'_j(x_{j+1})$, for $j = 0, 1, 2$ (3 *specified conditions*).

(e) $S''_{j+1}(x_{j+1}) = S''_j(x_{j+1})$, for $j = 0, 1, 2$ (3 *specified conditions*).

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

(i) Free: $S''(x_0) = S''(x_4) = 0$ (2 *specified conditions*).

(ii) Clamped: $S'(x_0) = f'(x_0)$ and $S'(x_4) = f'(x_4)$ (2 *specified conditions*).

Since uniqueness of solution requires the number of constants in (a), 16, to equal the number of conditions in (b) through (f), only one of the boundary conditions in (f) can be specified for the interpolatory cubic splines.

The cubic spline functions we will use for our basis functions are called **B-splines**, or *bell-shaped splines*. These differ from interpolatorysplines in that both sets of boundary conditions in (f) are satisfied. This requires the relaxation of two of the conditions in (b) through (e). Since the spline must have two continuous derivatives on $[x_0, x_4]$, we must delete two of the interpolation conditions from the description of the interpolatory splines. In particular, we modify condition (b) to

(b') $S(x_j) = f(x_j)$ for $j = 0, 2, 4$.

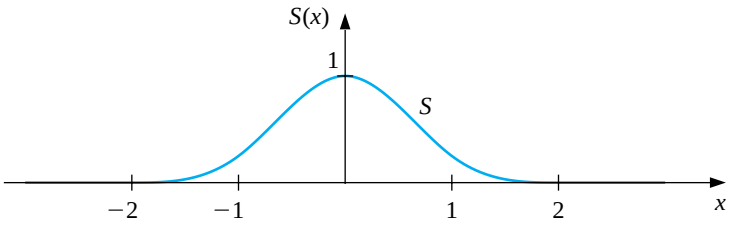
The basic B-spline S shown in Figure 11.5 uses the equally spaced nodes $x_0 = -2$, $x_1 = -1$, $x_2 = 0$, $x_3 = 1$, and $x_4 = 2$. It satisfies the interpolatory conditions

(b'). $S(x_0) = 0$, $S(x_2) = 1$, $S(x_4) = 0$;

as well as both sets of conditions

(i) $S''(x_0) = S''(x_4) = 0$ and (ii) $S'(x_0) = S'(x_4) = 0$.

Figure 11.5



As a consequence, $S \in C^2(-\infty, \infty)$.

$$S(x) = \begin{cases} 0, & \text{for } x \le -2, \\ \frac{1}{4}(2+x)^3, & \text{for } -2 \le x \le -1, \\ \frac{1}{4}\left[(2+x)^3 - 4(1+x)^3\right], & \text{for } -1 < x \le 0, \\ \frac{1}{4}\left[(2-x)^3 - 4(1-x)^3\right], & \text{for } 0 < x \le 1, \\ \frac{1}{4}(2-x)^3, & \text{for } 1 < x \le 2, \\ 0, & \text{for } 2 < x. \end{cases} \tag{11.10}$$

To construct the basis functions ϕ_i in $C_0^2[0,1]$ we first partition $[0,1]$ by choosing a positive integer n and defining $h = 1/(n+1)$. This produces the equally-spaced nodes $x_i = ih$, for each $i = 0, 1, \ldots, n+1$.We then define the basis functions $\{\phi_i\}_{i=0}^{n+1}$ as

$$\phi_i(x) = \begin{cases} S\left(\frac{x}{h}\right) - 4S\left(\frac{x+h}{h}\right), & \text{for } i = 0, \\ S\left(\frac{x-h}{h}\right) - S\left(\frac{x+h}{h}\right), & \text{for } i = 1, \\ S\left(\frac{x-ih}{h}\right), & \text{for } 2 \le i \le n-1, \\ S\left(\frac{x-nh}{h}\right) - S\left(\frac{x-(n+2)h}{h}\right), & \text{for } i = n, \\ S\left(\frac{x-(n+1)h}{h}\right) - 4S\left(\frac{x-(n+2)h}{h}\right), & \text{for } i = n+1. \end{cases}$$

It is not difficult to show that $\{\phi_i\}_{i=0}^{n+1}$ is a linearly independent set of cubic splines satisfying $\phi_i(0) = \phi_i(1) = 0$, for each $i = 0, 1, \ldots, n, n+1$. The graphs of ϕ_i, for $2 \le i \le n-1$, are shown in Figure 11.6 and the graphs of ϕ_0, ϕ_1, ϕ_n, and ϕ_{n+1} are in Figure 11.7.

Figure 11.6

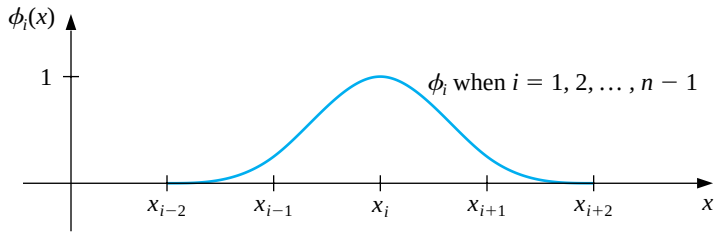
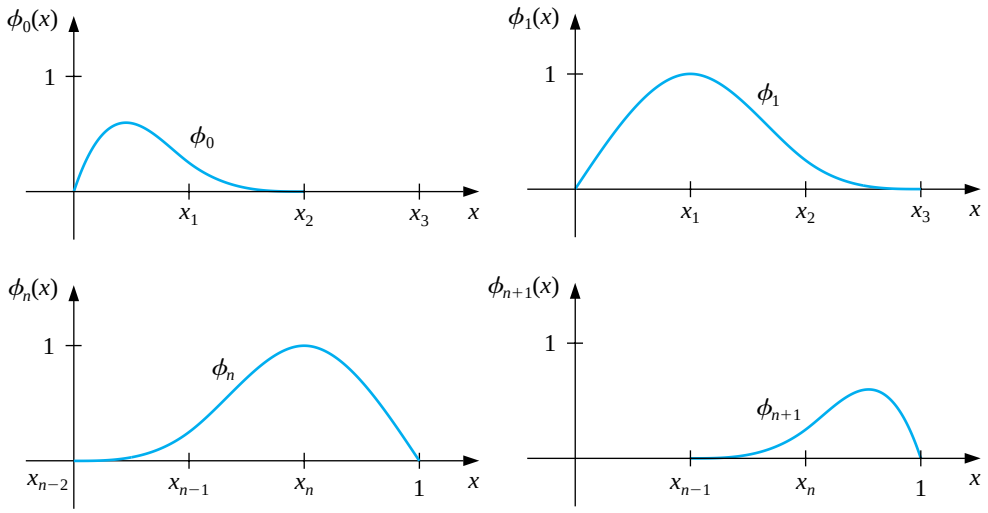


Figure 11.7



Since $\phi_i(x)$ and $\phi_i'(x)$ are nonzero only for $x_{i-2} \le x \le x_{i+2}$, the matrix in the Rayleigh-Ritz approximation is a band matrix with bandwidth at most seven:

$$A = \begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} & 0 & & & & 0 \\ a_{10} & a_{11} & a_{12} & a_{13} & a_{14} & & & & \\ a_{20} & a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & & & \\ a_{30} & a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & a_{36} & & \\ 0 & & & & & & & 0 & \\ & & & & & & & a_{n-2,n+1} & \\ & & & & & & & a_{n-1,n+1} & \\ & & & & & & & a_{n,n+1} & \\ 0 & & & & 0 & a_{n+1,n-2} & a_{n+1,n-1} & a_{n+1,n} & a_{n+1,n+1} \end{bmatrix}, \tag{11.11}$$

where

$$a_{ij} = \int_0^1 \{p(x)\phi_i'(x)\phi_j'(x) + q(x)\phi_i(x)\phi_j(x)\}\,dx,$$

for each $i = 0, 1, \ldots, n+1$ and $j = 0, 1, \ldots, n+1$. The vector $\mathbf{b}$ has the entries

$$b_i = \int_0^1 f(x)\phi_i(x)\,dx.$$

The matrix A is positive definite, so the linear system $A\mathbf{c} = \mathbf{b}$ can be quickly and stably solved by Choleski's method or by Gaussian elimination. The program CSRRG116 constructs the cubic spline approximation described by the Rayleigh-Ritz technique.

EXAMPLE 2 Consider the boundary-value problem

$$-y'' + \pi^2 y = 2\pi^2 \sin(\pi x), \qquad \text{for } 0 \le x \le 1, \quad \text{where } y(0) = y(1) = 0.$$

In Example 1 we let $h = 0.1$ and generated approximations using piecewise-linear basis functions. Here we let $n = 3$, so that $h = 0.25$. The cubic spline basis functions are

$$\phi_0(x) = \begin{cases} 12x - 72x^2 + 112x^3, & \text{for } 0 \le x \le 0.25 \\ 2 - 12x + 24x^2 - 16x^3, & \text{for } 0.25 < x \le 0.5 \\ 0, & \text{otherwise} \end{cases}$$

$$\phi_1(x) = \begin{cases} 6x - 32x^3, & \text{for } 0 \le x \le 0.25 \\ -\frac{5}{4} + 21x - 60x^2 + 48x^3, & \text{for } 0.25 < x \le 0.5 \\ \frac{27}{4} - 27x + 36x^2 - 16x^3, & \text{for } 0.5 < x \le 0.75 \\ 0, & \text{otherwise} \end{cases}$$

$$\phi_2(x) = \begin{cases} 16x^3, & \text{for } 0 \le x \le 0.25 \\ 1 - 12x + 48x^2 - 48x^3, & \text{for } 0 < x \le 0.5 \\ -11 + 60x - 96x^2 + 48x^3, & \text{for } 0.5 < x \le 0.75 \\ 16 - 48x + 48x^2 - 16x^3, & \text{for } 0.75 < x \le 1 \\ 0, & \text{otherwise} \end{cases}$$

$$\phi_3(x) = \begin{cases} -\frac{1}{4} + 3x - 12x^2 + 16x^3, & \text{for } 0.25 < x \le 0.5 \\ \frac{31}{4} - 45x + 84x^2 - 48x^3, & \text{for } 0.5 < x \le 0.75 \\ -26 + 90x - 96x^2 + 32x^3, & \text{for } 0.75 < x \le 1 \\ 0, & \text{otherwise} \end{cases}$$

and

$$\phi_4(x) = \begin{cases} -2 + 12x - 24x^2 + 16x^3, & \text{for } 0.5 < x \le 0.75 \\ 52 - 204x + 264x^2 - 112x^3, & \text{for } 0.75 \le x \le 1 \\ 0, & \text{otherwise} \end{cases}$$

The entries in the matrix A are generated as follows

$$
\begin{aligned}
a_{00} &= \int_0^1 \left\{ p(x) \left[\phi_0'(x)\right]^2 + q(x) \left[\phi_0(x)\right]^2 \right\} dx \\
&= \int_0^{0.5} \left\{ \left[\phi_0'(x)\right]^2 + \pi^2 \left[\phi_0(x)\right]^2 \right\} dx = 6.5463531 \\
a_{01} = a_{10} &= \int_0^1 \left\{ p(x)\phi_0'(x)\phi_1'(x) + q(x)\phi_0(x)\phi_1(x) \right\} dx \\
&= \int_0^{0.5} \left\{ \phi_0'(x)\phi_1'(x) + \pi^2 \phi_0(x)\phi_1(x) \right\} dx = 4.0764737 \\
a_{02} = a_{20} &= \int_0^1 \left\{ p(x)\phi_0'(x)\phi_2'(x) + q(x)\phi_0(x)\phi_2(x) \right\} dx \\
&= \int_0^{0.5} \left\{ \phi_0'(x)\phi_2'(x) + \pi^2 \phi_0(x)\phi_2(x) \right\} dx = -1.3722239 \\
a_{03} = a_{30} &= \int_0^1 \left\{ p(x)\phi_0'(x)\phi_3'(x) + q(x)\phi_0(x)\phi_3(x) \right\} dx \\
&= \int_{0.25}^{0.5} \left\{ \phi_0'(x)\phi_3'(x) + \pi^2 \phi_0(x)\phi_3(x) \right\} dx = -0.73898482 \\
a_{11} &= \int_0^1 \left\{ p(x) \left[\phi_1'(x)\right]^2 + q(x) \left[\phi_1(x)\right]^2 \right\} dx \\
&= \int_0^{0.75} \left\{ \left[\phi_1'(x)\right]^2 + \pi^2 \left[\phi_1(x)\right]^2 \right\} dx = 10.329086 \\
a_{12} = a_{21} &= \int_0^1 \left\{ p(x)\phi_1'(x)\phi_2'(x) + q(x)\phi_1(x)\phi_2(x) \right\} dx \\
&= \int_0^{0.75} \left\{ \phi_1'(x)\phi_2'(x) + \pi^2 \phi_1(x)\phi_2(x) \right\} dx = 0.26080684 \\
a_{13} = a_{31} &= \int_0^1 \left\{ p(x)\phi_1'(x)\phi_3'(x) + q(x)\phi_1(x)\phi_3(x) \right\} dx \\
&= \int_{0.25}^{0.75} \left\{ \phi_1'(x)\phi_3'(x) + \pi^2 \phi_1(x)\phi_3(x) \right\} dx = -1.6678178 \\
a_{14} = a_{41} &= \int_0^1 \left\{ p(x)\phi_1'(x)\phi_4'(x) + q(x)\phi_1(x)\phi_4(x) \right\} dx \\
&= \int_{0.5}^{0.75} \left\{ \phi_1'(x)\phi_4'(x) + \pi^2 \phi_1(x)\phi_4(x) \right\} dx = -0.73898482 \\
a_{22} &= \int_0^1 \left\{ p(x) \left[\phi_2'(x)\right]^2 + q(x) \left[\phi_2(x)\right]^2 \right\} dx \\
&= \int_0^1 \left\{ \left[\phi_2'(x)\right]^2 + \pi^2 \left[\phi_2(x)\right]^2 \right\} dx = 8.6612683 \\
a_{23} = a_{32} &= \int_0^1 \left\{ p(x)\phi_2'(x)\phi_3'(x) + q(x)\phi_2(x)\phi_3(x) \right\} dx \\
&= \int_{0.25}^1 \left\{ \phi_2'(x)\phi_3'(x) + \pi^2 \phi_2(x)\phi_3(x) \right\} dx = 0.26080684
\end{aligned}
$$

$$
\begin{aligned}
a_{24} = a_{42} &= \int_0^1 \{p(x)\phi_2'(x)\phi_4'(x) + q(x)\phi_2(x)\phi_4(x)\}\, dx \\
&= \int_{0.5}^1 \{\phi_2'(x)\phi_4'(x) + \pi^2\phi_2(x)\phi_4(x)\}\, dx = -1.3722239 \\
a_{33} &= \int_0^1 \left\{p(x)\left[\phi_3'(x)\right]^2 + q(x)\left[\phi_3(x)\right]^2\right\} dx \\
&= \int_{0.25}^1 \left\{\left[\phi_3'(x)\right]^2 + \pi^2\left[\phi_3(x)\right]^2\right\} dx = 10.329086 \\
a_{34} = a_{43} &= \int_0^1 \{p(x)\phi_3'(x)\phi_4'(x) + q(x)\phi_3(x)\phi_4(x)\}\, dx \\
&= \int_{0.5}^1 \{\phi_3'(x)\phi_4'(x) + \pi^2\phi_3(x)\phi_4(x)\}\, dx = 4.0764737 \\
a_{44} &= \int_0^1 \left\{p(x)\left[\phi_4'(x)\right]^2 + q(x)\left[\phi_4(x)\right]^2\right\} dx \\
&= \int_{0.5}^1 \left\{\left[\phi_4'(x)\right]^2 + \pi^2\left[\phi_4(x)\right]^2\right\} dx = 6.5463531
\end{aligned}
$$

and

$$
\begin{aligned}
b_0 &= \int_0^1 f(x)\phi_0(x)\, dx = \int_0^{0.5} \left(2\pi^2 \sin \pi x\right)\phi_0(x)\, dx = 1.0803542 \\
b_1 &= \int_0^1 f(x)\phi_1(x)\, dx = \int_0^{0.75} \left(2\pi^2 \sin \pi x\right)\phi_1(x)\, dx = 4.7202512 \\
b_2 &= \int_0^1 f(x)\phi_2(x)\, dx = \int_0^1 \left(2\pi^2 \sin \pi x\right)\phi_2(x)\, dx = 6.6754433 \\
b_3 &= \int_0^1 f(x)\phi_3(x)\, dx = \int_{0.25}^1 \left(2\pi^2 \sin \pi x\right)\phi_3(x)\, dx = 4.7202512 \\
b_4 &= \int_0^1 f(x)\phi_4(x)\, dx = \int_{0.5}^1 \left(2\pi^2 \sin \pi x\right)\phi_4(x)\, dx = 1.08035418
\end{aligned}
$$

The solution to the system $A\mathbf{c} = \mathbf{b}$ is $c_0 = 0.00060266150$, $c_1 = 0.52243908$, $c_2 = 0.73945127$, $c_3 = 0.52243908$, and $c_4 = 0.00060264906$. Evaluating $\phi(x) = \sum_{i=0}^4 c_i\phi_i(x)$ at each x_j, for $0 \le j \le 4$, gives the results in Table 11.8. Notice that these results using $h = 0.25$ are superior to the piecewise-linear results given in Example 1, where we used $h = 0.1$. □

Table 11.8

i	x_i	$\phi(x_i)$	$y(x_i)$	$\lvert\phi(x_i) - y(x_i)\rvert$
0	0	0	0	0
1	0.25	0.70745256	0.70710678	0.00034578
2	0.5	1.0006708	1	0.0006708
3	0.75	0.70745256	0.70710678	0.00034578
4	1	0	0	0

The integrations should be performed in two steps, as was done in the Piecewise Linear method. First, construct cubic spline interpolatory polynomials for p, q, and f using the methods presented in Section 3.5. Then approximate the integrands by products of cubic splines or derivatives of cubic splines. Since these integrands are piecewise polynomials, they can be integrated exactly on each subinterval and then summed.

In general, this technique produces approximations $\phi(x)$ to $y(x)$ that satisfy

$$\left[\int_0^1 |y(x) - \phi(x)|^2 dx\right]^{1/2} = O(h^4), \qquad 0 \le x \le 1.$$

EXERCISE SET 11.6

1. Use the Piecewise Linear method to approximate the solution to the boundary-value problem

$$y'' + \frac{\pi^2}{4}y = \frac{\pi^2}{16}\cos\frac{\pi}{4}x, \qquad \text{for } 0 \le x \le 1 \quad \text{with } y(0) = y(1) = 0$$

using $x_0 = 0, x_1 = 0.3, x_2 = 0.7, x_3 = 1$ and compare the results to the actual solution $y(x) = -\frac{1}{3}\cos\frac{\pi}{2}x - \frac{\sqrt{2}}{6}\sin\frac{\pi}{2}x + \frac{1}{3}\cos\frac{\pi}{4}x$.

2. Use the Piecewise Linear method to approximate the solution to the boundary-value problem

$$-\frac{d}{dx}(xy') + 4y = 4x^2 - 8x + 1, \qquad \text{for } 0 \le x \le 1 \quad \text{with } y(0) = y(1) = 0$$

using $x_0 = 0$, $x_1 = 0.4$, $x_2 = 0.8$, $x_3 = 1$ and compare the results to the actual solution $y(x) = x^2 - x$.

3. Use the Piecewise Linear method to approximate the solutions to the following boundary-value problems and compare the results to the actual solution:

   (a) $-x^2y'' - 2xy' + 2y = -4x^2$, for $0 \le x \le 1$ with $y(0) = y(1) = 0$; use $h = 0.1$; actual solution $y(x) = x^2 - x$.

   (b) $-\frac{d}{dx}(e^x y') + e^x y = x + (2 - x)e^x$, for $0 \le x \le 1$ with $y(0) = y(1) = 0$; use $h = 0.1$; actual solution $y(x) = (x - 1)(e^{-x} - 1)$.

   (c) $-\frac{d}{dx}(e^{-x}y') + e^{-x}y = (x - 1) - (x + 1)e^{-(x-1)}$, for $0 \le x \le 1$ with $y(0) = y(1) = 0$; use $h = 0.05$; actual solution $y(x) = x(e^x - e)$.

   (d) $-(x + 1)y'' - y' + (x + 2)y = [2 - (x + 1)^2]e \ln 2 - 2e^x$, for $0 \le x \le 1$ with $y(0) = y(1) = 0$; use $h = 0.05$; actual solution $y(x) = e^x \ln(x + 1) - (e \ln 2)x$.

4. Use the Cubic Spline method with $n = 3$ to approximate the solution to each of the following boundary-value problems and compare the results to the actual solutions.

   (a) $y'' + \frac{\pi^2}{4}y = \frac{\pi^2}{16}\cos\frac{\pi}{4}x$, for $0 \le x \le 1$ with $y(0) = 0$ and $y(1) = 0$

   (b) $-\frac{d}{dx}(xy') + 4y = 4x^2 - 8x + 1$, for $0 \le x \le 1$ with $y(0) = 0$ and $y(1) = 0$.

5. Repeat Exercise 3 using the Cubic Spline method.

6. Show that the boundary-value problem

$$-\frac{d}{dx}(p(x)y')+q(x)y = f(x), \qquad \text{for } 0 \le x \le 1 \quad \text{with } y(0) = \alpha \text{ and } y(1) = \beta,$$

can be transformed by the change of variable

$$z = y - \beta x - (1 - x)\alpha$$

into the form

$$-\frac{d}{dx}(p(x)z') + q(x)z = F(x), \quad 0 \le x \le 1, \qquad z(0) = 0, \qquad z(1) = 0.$$

7. Use Exercise 6 and the Piecewise Linear method with $n = 9$ to approximate the solution to the boundary-value problem

$$-y'' + y = x, \quad \text{for } 0 \le x \le 1 \quad \text{with} \quad y(0) = 1 \quad \text{and} \quad y(1) = 1 + e^{-1}.$$

8. Repeat Exercise 7 using the Cubic Spline method.

9. Show that the boundary-value problem

$$-\frac{d}{dx}(p(x)y')+q(x)y = f(x), \qquad \text{for } a \le x \le b \quad \text{with } y(a) = \alpha \text{ and } y(b) = \beta,$$

can be transformed into the form

$$-\frac{d}{dw}(p(w)z')+q(w)z = F(w), \qquad \text{for } 0 \le w \le 1 \quad \text{with } z(0) = 0 \text{ and } z(1) = 0,$$

by a method similar to that given in Exercise 6.

10. Show that the set of piecewise linear basis functions is linearly independent on $[0, 1]$.

11. Use the definition of positive definite to show that the matrix given by the piecewise linear basis functions satisfies this condition.

11.7 Survey of Methods and Software

In this chapter we discussed methods for approximating solutions to boundary-value problems. For the linear boundary-value problem

$$y'' = p(x)y' + q(x)y + r(x), \quad a \le x \le b, \quad y(a) = \alpha, \quad y(b) = \beta,$$

we considered both a linear shooting method and a finite-difference method to approximate the solution. The shooting method uses an initial-value technique to solve the problems

$$y'' = p(x)y' + q(x)y + r(x), \quad a \le x \le b, \quad y(a) = \alpha, \quad y'(a) = 0,$$

and

$$y'' = p(x)y' + q(x)y, \quad a \le x \le b, \quad y(a) = 0, \quad y'(a) = 1.$$

A weighted average of these solutions produces a solution to the linear boundary-value problem.

In the finite-difference method, we replaced y'' and y' with difference approximations and solved a linear system. Although the approximations may not be as accurate as the shooting method, there is less sensitivity to roundoff error. Higher-order difference methods are available, or extrapolation can be used to improve accuracy.

For the nonlinear boundary problem

$$y'' = f(x, y, y'), \quad a \le x \le b, \quad y(a) = \alpha, \quad y(b) = \beta,$$

we also presented two methods. The nonlinear shooting method requires the solution of the initial-value problem

$$y'' = f(x, y, y'), \quad a \le x \le b, \quad y(a) = \alpha, \quad y'(a) = t,$$

for an initial choice of t. We improved the choice by using Newton's method to approximate the solution, t, to $y(b, t) = \beta$. This method required solving two initial-value problems at each iteration. The accuracy is dependent on the choice of method for solving the initial-value problems.

The finite-difference method for the nonlinear equation requires the replacement of y'' and y' by difference quotients, which results in a nonlinear system. This system is solved using Newton's method. Higher-order differences or extrapolation can be used to improve accuracy. Finite-difference methods tend to be less sensitive to roundoff error than shooting methods.

The Rayleigh-Ritz-Galerkin method was illustrated by approximating the solution to the boundary-value problem

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

A piecewise-linear approximation or a cubic spline approximation can be obtained.

Most of the material concerning second-order boundary-value problems can be extended to problems with boundary conditions of the form

$$\alpha_1 y(a) + \beta_1 y'(a) = \alpha \quad \text{and} \quad \alpha_2 y(b) + \beta_2 y'(b) = \beta,$$

where $|\alpha_1| + |\beta_1| \neq 0$ and $|\alpha_2| + |\beta_2| \neq 0$, but some of the techniques become quite complicated. The reader who is interested in problems of this type is advised to consider a book specializing in boundary-value problems, such as [K,H].

The IMSL and NAG libraries contain methods for boundary-value problems. There are Finite-Difference methods and Shooting methods that are based on their adaptations of variable-step-size Runge-Kutta methods.

The subroutines MUSL and MUSN in the ODE package contained in the Netlib library solve the linear and nonlinear two-point boundary-value problems, respectively. Both routines are based on multiple shooting methods.

Further information on the general problems involved with the numerical solution to two-point boundary-value problems can be found in Keller [K,H] and Bailey, Shampine and Waltman [BSW]. Roberts and Shipman [RS] focuses on the shooting methods for the two-point boundary-value problem, and Pryce [Pr] restricts attention to Sturm-Liouville problems. The book by Ascher, Mattheij, and Russell [AMR] has a comprehensive presentation of multiple shooting and parallel shooting methods.