Write the second-order initial-value problems (11.3) and (11.4) 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 \( R_2 \), with \( R_1 < R_2 \), 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

\[
\frac{d^2 u}{dr^2} + \frac{2}{r} \frac{du}{dr} = 0, \quad R_1 \leq r \leq R_2, \quad u(R_1) = V_1, \quad u(R_2) = 0.
\]

Suppose \( R_1 = 2 \text{ in.}, \ R_2 = 4 \text{ in.}, \) and \( V_1 = 110 \text{ volts} \).

a. Approximate \( u(3) \) using the Linear Shooting Algorithm.

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

\[
u(r) = \frac{V_1 R_1}{r} \left( \frac{R_2 - r}{R_2 - R_1} \right).
\]

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 \( y_2 = 0 \).

9. Consider the boundary-value problem

\[
y'' + y = 0, \quad 0 \leq x \leq b, \quad y(0) = 0, \quad y(b) = B.
\]

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. Attempt to apply Exercise 9 to the boundary-value problem

\[
y'' - y = 0, \quad 0 \leq x \leq b, \quad y(0) = 0, \quad y(b) = B.
\]

What happens? How do both problems relate to Corollary 11.2?

11.2 The Shooting Method for Nonlinear Problems

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

\[
y'' = f(x, y, y'), \quad a \leq x \leq b, \quad y(a) = \alpha, \quad y(b) = \beta,
\]

is similar to the linear case, 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 need to use the solutions to a sequence of initial-value problems of the form

\[
y'' = f(x, y, y'), \quad a \leq x \leq b, \quad y(a) = \alpha, \quad y'(a) = t,
\]

involving a parameter \( t \), to approximate the solution to the boundary-value problem. We do this by choosing the parameters \( t = t_k \) so 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.) We start with a parameter \( t_0 \) that determines the initial elevation at which the object is fired from the point \((a, \alpha)\) and along the curve described by the solution to the initial-value problem:

\[
y'' = f(x, y, y'), \quad a \leq x \leq b, \quad y(a) = \alpha, \quad y'(a) = t_0.
\]

If \( y(b, t_0) \) is not sufficiently close to \( \beta \), we correct our approximation by choosing elevations \( t_1, t_2, \) and so on, until \( y(b, t_k) \) is sufficiently close to "hitting" \( \beta \). (See Figure 11.3.)
To determine the parameters $t_k$, suppose a boundary-value problem of the form (11.6) satisfies the hypotheses of Theorem 11.1. If $y(x, t)$ denotes the solution to the initial-value problem (11.7), the problem is to determine $t$ so that

\[(11.8)\quad y(b, t) - \beta = 0.\]

This is a nonlinear equation of the type considered in Chapter 2, so a number of methods are available.

To use the Secant method to solve the problem, we need to choose initial approximations $t_0$ and $t_1$ and then generate the remaining terms of the sequence by

\[t_k = t_{k-1} - \frac{(y(b, t_{k-1}) - \beta)(t_{k-1} - t_{k-2})}{y(b, t_{k-1}) - y(b, t_{k-2})}, \quad k = 2, 3, \ldots.\]

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

\[(11.9)\quad t_k = t_{k-1} - \frac{y(b, t_{k-1}) - \beta}{\frac{dy}{dt}(b, t_{k-1})},\]

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 $y(b, t_0), y(b, t_1), \ldots, y(b, t_{k-1}).$

Suppose we rewrite the initial-value problem (11.7), emphasizing that the solution depends on both $x$ and $t$ as

\[(11.10)\quad y''(x, t) = f(x, y(x, t), y'(x, t)), \quad a \leq x \leq b, \quad y(a, t) = \alpha, \quad y'(a, t) = t.\]

We have retained the prime notation to indicate differentiation with respect to $x$. Since we need to determine $(dy/dt)(b, t)$ when $t = t_{k-1}$, we first take the partial derivative of (11.10) with respect to $t$. This implies that

\[\frac{\partial y''}{\partial t}(x, t) = \frac{\partial f}{\partial t}(x, y(x, t), y'(x, t))\]

\[= \frac{\partial f}{\partial x}(x, y(x, t), y'(x, t)) \frac{\partial x}{\partial t} + \frac{\partial f}{\partial y}(x, y(x, t), y'(x, t)) \frac{\partial y}{\partial t}(x, t)\]

\[+ \frac{\partial f}{\partial y'}(x, y(x, t), y'(x, t)) \frac{\partial y'}{\partial t}(x, t).\]

Since $x$ and $t$ are independent, $\partial x/\partial t = 0$ and

\[(11.11)\quad \frac{\partial y''}{\partial t}(x, t) = \frac{\partial f}{\partial y}(x, y(x, t), y'(x, t)) \frac{\partial y}{\partial t}(x, t) + \frac{\partial f}{\partial y'}(x, y(x, t), y'(x, t)) \frac{\partial y'}{\partial t}(x, t)\]

for $a \leq x \leq b$. 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, (11.11) with the initial conditions becomes the initial-value problem

\[
z''(x, t) = \frac{\partial f}{\partial y}(x, y', z(x, t)) + \frac{\partial f}{\partial y'}(x, y', z'(x, t)), \quad a \leq x \leq b, \quad z(a, t) = 0, \quad z'(a, t) = 1.
\]

(11.12)

Newton’s method therefore requires that two initial-value problems be solved for each iteration, (11.10) and (11.12). Then from Eq. (11.9),

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

(11.13)

Of course, none of these initial-value problems is solved exactly; the solutions are approximated by one of the methods discussed in Chapter 5. Algorithm 11.2 uses the fourth-order Runge-Kutta method to approximate both solutions required by Newton’s method. A similar procedure for the Secant method is considered in Exercise 4.

**Algorithm 11.2**

To approximate the solution of the nonlinear boundary-value problem

\[
y'' = f(x, y, y'), \quad a \leq x \leq b, \quad y(a) = \alpha, \quad y(b) = \beta.
\]

(Note: Equations (11.10) and (11.12) are written as first-order systems and solved.)

**INPUT** endpoints \( a, b \); boundary conditions \( \alpha, \beta \); 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 \( h = (b - a)/N; \)

\[
k = 1;
\]

\[
TK = (\beta - \alpha)/(b - a). \quad \text{(Note: TK could also be input.)}
\]

**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.

(The Runge-Kutta method for systems is used in Steps 5 and 6.)

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

**Step 6** Set \( k_{1,1} = hw_{2,i-1}; \)

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

**Step 7** If \(|w_{1,N} - \beta| \leq \text{TOL}\) 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** (Procedure is complete.)

STOP.

**Step 10** Set \(TK = TK - \left( \frac{w_{1,N} - \beta}{u_1} \right)\); (Newton's method is used to compute \(TK\).)

\(k = k + 1.\)

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

(Procedure completed unsuccessfully.)

STOP.

In Step 7, the best approximation to \(\beta\) we can expect for \(w_{1,N}(t_k)\) is \(O(h^n)\), if the approximation method selected for Step 6 gives \(O(h^n)\) rate of convergence.
The value \( t_0 = TK \) selected in Step 1 is the slope of the straight line through \((a, \alpha)\) and \((b, \beta)\). If the problem satisfies the hypotheses of Theorem 11.1, any choice of \( t_0 \) will give convergence; but given a good choice of \( t_0 \), the convergence will improve and the procedure will work for many problems that do not satisfy these hypotheses.

**Example 1** Consider the boundary-value problem

\[
y'' = \frac{1}{8} (32 + 2x^3 - yy'), \quad 1 \leq x \leq 3, \quad y(1) = 17, \quad y(3) = \frac{43}{3},
\]

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

Applying the Shooting method given in Algorithm 11.2 to this problem requires approximating solutions to the initial-value problems

\[
y'' = \frac{1}{8} (32 + 2x^3 - yy'), \quad 1 \leq x \leq 3, \quad y(1) = 17, \quad y'(1) = t_k,
\]

and

\[
z'' = \frac{\partial f}{\partial y} z + \frac{\partial f}{\partial y'} z' = -\frac{1}{8} (y'z + yz'), \quad 1 \leq x \leq 3, \quad z(1) = 0, \quad z'(1) = 1,
\]

at each step in the iteration.

| \( x_i \) | \( w_{L,i} \) | \( y(x_i) \) | \( |w_{L,i} - y(x_i)| \) |
|---|---|---|---|
| 1.0 | 17.000000 | 17.000000 | |
| 1.1 | 15.755495 | 15.755455 | 4.06 \( \times \) 10^{-5} |
| 1.2 | 14.773389 | 14.773333 | 5.60 \( \times \) 10^{-5} |
| 1.3 | 13.997752 | 13.997692 | 5.94 \( \times \) 10^{-5} |
| 1.4 | 13.388629 | 13.388571 | 5.71 \( \times \) 10^{-5} |
| 1.5 | 12.916719 | 12.916667 | 5.23 \( \times \) 10^{-5} |
| 1.6 | 12.560046 | 12.560000 | 4.64 \( \times \) 10^{-5} |
| 1.7 | 12.301805 | 12.301765 | 4.02 \( \times \) 10^{-5} |
| 1.8 | 12.128923 | 12.128889 | 3.14 \( \times \) 10^{-5} |
| 1.9 | 12.031081 | 12.031053 | 2.84 \( \times \) 10^{-5} |
| 2.0 | 12.000023 | 12.000000 | 2.32 \( \times \) 10^{-5} |
| 2.1 | 12.029066 | 12.029048 | 1.84 \( \times \) 10^{-5} |
| 2.2 | 12.112741 | 12.112727 | 1.40 \( \times \) 10^{-5} |
| 2.3 | 12.246532 | 12.246522 | 1.01 \( \times \) 10^{-5} |
| 2.4 | 12.426673 | 12.426667 | 6.68 \( \times \) 10^{-6} |
| 2.5 | 12.650004 | 12.650000 | 3.61 \( \times \) 10^{-6} |
| 2.6 | 12.913847 | 12.913846 | 9.17 \( \times \) 10^{-7} |
| 2.7 | 13.215924 | 13.215926 | 1.43 \( \times \) 10^{-6} |
| 2.8 | 13.554282 | 13.554286 | 3.46 \( \times \) 10^{-6} |
| 2.9 | 13.927236 | 13.927241 | 5.21 \( \times \) 10^{-6} |
| 3.0 | 14.333327 | 14.333333 | 6.69 \( \times \) 10^{-6} |
If the stopping technique

$$|w_{1,N}(t_k) - y(3)| \leq 10^{-5}$$

is used, this problem requires four iterations and \( t_4 = -14.000203 \). The results obtained for this value of \( t \) are shown in Table 11.2.

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. For a general discussion of the convergence of the shooting techniques for nonlinear problems, the reader is referred to the excellent book by Keller [K,L]. In that reference, more general boundary conditions are discussed. It is also noted that the shooting technique for nonlinear problems is sensitive to round-off errors, especially if the solution \( y(x) \) and \( z(x, t) \) are rapidly increasing functions on \([a, b]\).

**Exercise Set 11.2**

1. Use the Nonlinear Shooting Algorithm with \( h = 0.5 \) to approximate the solution to the boundary-value problem

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

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

2. Use the Nonlinear Shooting Algorithm with \( h = 0.25 \) to approximate the solution to the boundary-value problem

$$y'' = 2y^3, \quad \text{for} \quad 1 \leq x \leq 2, \quad \text{where} \quad y(1) = \frac{1}{4} \quad \text{and} \quad y(2) = \frac{1}{5}.$$  

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

3. Use the Nonlinear Shooting method with \( TOL = 10^{-4} \) to approximate the solution to the following boundary-value problems. The actual solution is given for comparison to your results.

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

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

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

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

4. Change Algorithm 11.2 to incorporate the Secant method instead of Newton's method. Use \( t_0 = (\beta - \alpha)/(b - a) \) and \( t_1 = t_0 + (\beta - y(b, t_0))/(b - a) \).

5. Repeat Exercise 3(a) and 3(c) using the Secant algorithm derived in Exercise 4, and compare the number of iterations required for the two methods.