6

Numerical Solution of Ordinary Differential Equations

Most problems in the real world are modeled with differential equations because it is easier to see the relationship in terms of a derivative. An obvious example is Newton's Law—$f = M * a$—where the acceleration a is the rate of change of the velocity. Velocity is also a derivative, the rate of change the position, s, of an object of mass, M, when it is acted on by force, f. So we should think of Newton's Law as

$$d^2s/dt^2 = a = f/M,$$

a *second-order ordinary differential equation.* It is *ordinary* because it does not involve partial differentials and *second order* because the order of the derivative is two. The solution to this equation is a function, $s(t)$. This is a particularly easy problem to solve analytically when the acceleration is constant:

$$s(t) = (1/2)\, at^2 + v_0 t + s_0.$$

The solution contains two arbitrary constants, v_0 and s_0, the initial values for the velocity and position. The equation for $s(t)$ allows the computation of a numerical value for s, the position of the object, at any value for time, the independent variable, t.

Many differential equations can be solved analytically and you probably learned how to do this in a previous course. The general analytical solution will include arbitrary constants in a number equal to the order of the equation. If the same number of conditions on the solution are given, these constants can be evaluated.

When all of the conditions on the problem are specified at the same value for the independent variable, the problem is termed an *initial-value problem.* If these are at two different values for the independent variable, usually at the boundaries of some region of interest, it is called a *boundary-value problem.*

This chapter describes techniques for solving ordinary differential equations by numerical methods. To solve the problem numerically, the required number of conditions must be known and these values are used in the numerical solution. We will begin the chapter with a Taylor series method that is not only a good method in itself but serves as the basis for

several other methods. We start with first-order initial-value problems and later cover higher-order problems and boundary-value problems.

With an initial-value problem, the numerical solution begins at the initial point and marches from there to increasing values for the independent variable. With a boundary problem, one must march toward the other boundary and match with the condition(s) there. This is not as easy to accomplish. Certain boundary-value problems have a solution only for *characteristic values* for a parameter; these are known as *characteristic-value problems.*

When we attempt to solve a differential equation, we must be sure that there really is a solution and that the solution we get is unique. This requires that $f(x, y)$ in $dy/dx = f(x, y)$ meet the *Lipschitz condition:*

Let $f(x, y)$ be defined and continuous on a region R that contains the point (x_0, y_0). We assume that the region is a closed and bounded rectangle. Then $f(x, y)$ is said to satisfy the Lipschitz condition if:

There is an $L > 0$ so that for all x, y_1, y_2 in R, we have

$$|f(x, y_1) - f(x, y_2)| < L|y_1 - y_2|.$$

For most problems and all examples of this chapter, the condition is met.

There is a similar set of conditions for the solution to a boundary-value problem to exist and be unique. A linear problem of the form

$$\frac{d^2u}{dx^2} = pu' + qu + r, \qquad \text{for } x \text{ on } [a, b],$$

with

$$u(a) = uL, \qquad u(b) = uR,$$

where p, q, and r are functions of x only, has a unique solution if two conditions are met:

$$p, q, \text{ and } r \text{ must be continuous on } [a, b],$$

and

$$q > 0 \text{ on } [a, b].$$

If the problem is nonlinear, more severe conditions apply that involve the partial derivatives of the right-hand side with respect to u and u'.

Contents of This Chapter

6.2 The Euler Method and Its Modifications

Describes a method that is easy to use but is not very precise unless the step size, the intervals for the projection of the solution, is very small. Modifications permit the use of a larger step size or give greater accuracy at the same size of steps. These methods are based on low-order Taylor series.

6.3 Runge–Kutta Methods

Presents methods that are based on more terms of a Taylor series than the Euler methods and are thereby much more accurate. A very widely used method, the Runge–Kutta–Fehlberg method (RKF) allows an estimation of the error as computations are made so the step size can be changed appropriately.

6.4 Multistep Methods

Covers methods that are more efficient than the previous methods, which are called *single-step methods.* They require a number of starting values in addition to the initial value. A Runge–Kutta method is frequently used to get these starting values. A valuable adjunct to a multistep method is to first compute a *predicted* value and then do a second computation to get a *corrected value.* Doing this monitors the accuracy of the computations.

6.5 Higher-Order Equations and Systems

Describes how the methods previously covered can solve an equation of order higher than the first. This is done by converting the equation to a system of first-order problems. Hence, even a system of higher-order problems can be handled.

6.6 Stiff Equations

Discusses a type of problem that poses difficulties in avoiding *instability,* the growth of initial error as the solution proceeds.

6.7 Boundary-Value Problems

Extends the methods previously described to solve a differential equation whose conditions are specified at not just the initial point. This section also describes how the solution can be approximated if the derivatives are replaced by difference quotients, as explained in Chapter 5.

6.8 Characteristic-Value Problems

Shows how that class of boundary-value problems that have a solution only for certain values of a parameter can be solved. These certain values are the *eigenvalues* of the system; eigenvalues and their associated *eigenvectors* are essential matrix-related quantities that have applications in many fields. Two different ways to obtain eigenvalues are described.

6.1 The Taylor-Series Method

As you have seen before, a Taylor series is a way to express (most) functions as a power series. When expanded about the point $x = a$, the coefficients of the powers of $(x - a)$ include the values of the successive derivatives of the function at $x = a$. This means that if we know enough about a function at some point $x = a$, that is, its value and the value of all of its derivatives, we can (usually) write a series that has the same value as the function at all values of x. We will use x_0 to represent $x = a$.

In the present application, we are given the function that is the first derivative of $y(x)$: $y' = f(x, y)$, and an initial value, $y(x_0)$. With this information we can write the Taylor series for $y(x)$ about $x = x_0$. We just differentiate $y'(x) = f(x, y)$ as many times as we desire and evaluate these derivatives at $x = x_0$. The problem is that, when $y'(x)$ involves not just x but the unknown y as well, the higher derivatives may not be easy to come by.

Even so, these higher derivatives can be written in terms of x and the lower derivatives of y. We only want their values at $x = x_0$. Here is an example:

$$\frac{dy}{dx} = -2x - y, \qquad y(0) = -1. \tag{6.1}$$

(This particularly simple example is chosen to illustrate the method so that you can readily check the computational work. The analytical solution,

$$y(x) = -3e^{-x} - 2x + 2$$

is obtained immediately by application of standard methods and will be compared with our numerical results to show the error at any step.)

We develop the relation between y and x by finding the coefficients of the Taylor series in which we expand y about the point $x = x_0$:

$$y(x) = y(x_0) + y'(x_0)(x - x_0) + \frac{y''(x_0)}{2!}(x - x_0)^2 + \frac{y'''(x_0)}{3!}(x - x_0)^3 + \cdots.$$

If we let $x - x_0 = h$, we can write the series as

$$y(x) = y(x_0) + y'(x_0)h + \frac{y''(x_0)}{2}h^2 + \frac{y'''(x_0)}{6}h^3 + \cdots. \tag{6.2}$$

Because $y(x_0)$ is our *initial condition,* the first term is known from the initial condition $y(0) = -1$. (Because the expansion is about the point $x = 0$, our Taylor series is actually the Maclaurin series in this example.)

We get the coefficient of the second term by substituting $x = 0, y = -1$ in the equation for the first derivative, Eq. (6.1):

$$y'(x_0) = y'(0) = -2(0) - (-1) = 1.$$

We get the second- and higher-order derivatives by successively differentiating the equation for the first derivative. Each of these derivatives is evaluated corresponding to $x = 0$ to get the various coefficients:

Table 6.1

x	$y(x)$	Anal	Error
0.00000	−1.00000	−1.00000	0.00000
0.10000	−0.91451	−0.91451	0.00000
0.20000	−0.85620	−0.85619	0.00001
0.30000	−0.82251	−0.82245	0.00006
0.40000	−0.81120	−0.81096	0.00024
0.50000	−0.82031	−0.81959	0.00072
0.60000	−0.84820	−0.84643	0.00177

$$y''(x) = -2 - y', \qquad y''(0) = -2 - 1 = -3,$$
$$y'''(x) = -y'', \qquad y'''(0) = 3,$$
$$y^{(4)}(x) = -y''', \qquad y^{(4)}(0) = -3.$$

We then write our series solution for y, letting $x = h$ be the value at which we wish to determine y:

$$y(h) = -1 + 1.0h - 1.5h^2 + 0.5h^3 - 0.125h^4 + \text{error}.$$

Table 6.1 shows how the computed solutions compare to the analytical between $x = 0$ and $x = 0.6$. At the start, the Taylor-series solution agrees well, but beyond $x = 0.3$ they differ increasingly. More terms in the series would extend the range of good agreement.

The error of this computation is given by the next term in the series, evaluated at a point between 0 and x:

$$\text{Error} = (x^5/120)y^{(5)}(\xi), \qquad 0 < \xi < x.$$

We have used the so-called next-term rule before. How good is this estimate of the error at $x = 0.6$? The next term is (3/120) * $(0.6)^5$ = 0.00194, comparing well to the actual error of 0.00177.

We stated earlier that the analytical solution of the example differential equation can be obtained by "the application of standard methods." MATLAB can do this:

```
EDU>> dsolve('Dy = -2*x - y', 'y(0) = -1', 'x')
     ans =
       -2*x + 2 - 3*exp(-x)
```

which is the same as the above with terms in a different order.

Maple can get the Taylor-series solution:

```
> deq := diff(y(x),x) = -2*x - y(x):
> dsolve ({deq, y(0) = -1), y(x), series);
```

$$y(x) = -1 + x - \frac{3}{2}x^2 + \frac{1}{2}x^3 - \frac{1}{8}x^4 + \frac{1}{40}x^5 + O(x^6)$$

which is the series of order 6 and the error order.

When the function that defines $y'(x)$ is not as simple as this, getting the successive derivatives is not as easy. Consider

$$y'(x) = \frac{x}{(y - x^2)}.$$

You will find that the successive derivatives get very messy.

Even though computers are not readily programmed to produce these higher derivatives, computer algebra systems like Maple and *Mathematica* do have the capabilities that we need.

There is another approach—*automatic differentiation*. This is different from the symbolic differentiation that computer algebra systems use. It produces machine code that finds values of the derivatives when *dy/dx* is defined through a *code list*.

We will not give a thorough explanation, only an example, but L. R. Rall (1981) and Corliss and Chang (1982) are good sources for more information. Here is our example:

$$\text{Solve } y' = f(x, y) = \frac{x}{(y - x^2)} \text{ using automatic differentiation with } y(0) = 1.$$

To do this, we first create a code list, which is just a name for a sequence of statements that define *dy/dx*, with only a single operation on each line:

```
T1 = x*x
T2 = y - T1
dy/dx = x/T2    [which is f(x, y)].
```

We will use a simplified notation for the terms of the Taylor series:

$$(y)_k = \left(\frac{1}{k!}\right)\left[\frac{d^k y}{dx^k}\right], \qquad k = 0, \ldots, n.$$

And we will use $(x)_0 = x_0$. We then have $(y)_0 = y(x_0)$.

The software for automatic differentiation includes the standard rules for differentiation in recursive form, such as the derivatives of $(u + v)_k$, $(u - v)_k$, $(u * v)_k$, and $(u/v)_k$, plus the elementary functions, including sin, cos, ln, exp, and so on.

In our example, we have $(x)_0 = 0$, $(x)_1 = 1$ (because $dx/dx = 1$), so that $(x)_k = 0$ for all higher derivatives of x. From the initial condition, $(y)_0 = 1$ and from the expression for $y'(x)$, $(y)_1 = 0$. It is not hard to determine that $(y)_2 = 0.5$. The automatic differentiation software develops a recursion formula for the additional coefficients of the Taylor series. This formula is something like this:

$$(y)_k = \alpha_k \sum_{i=1}^{k-1} i(y)_i(y)_{k-1},$$

where the multiplier, α_k, is a complicated function of k.

Similar recursion formulas will be derived by the software for any differential equation that can be compiled into a code list, and these can have any initial condition.

For our example, all the odd-order terms are zero; the even-order terms are:

Order	0	2	4	6	8
Coefficient	1	$\frac{1}{2}$	$\frac{1}{8}$	$\frac{1}{48}$	$\frac{-1}{384}$

Using this in the Taylor series produces $y(0.1) = 1.0050125$, $y(0.2) = 1.0202013$.

The authors are especially grateful to Professor Ramon E. Moore of Ohio State University for calling our attention to this method for solving ordinary differential equations.

While getting the higher derivatives of $y' = x/(y - x^2)$ is awkward by hand, Maple has no trouble. If we want these up to the 22nd power of x, we must first reset the `Order` from its default value, then use the series option of `dsolve`.

```
> Order: = 22:
> deq: = diff(y(x), x) = x/(y(x) - x^2):
> dsolve({deq, y(0) = 1}, y(x), series);
```

$$y(x) = 1 + \frac{1}{2}x^2 + \frac{1}{8}x^4 + \frac{1}{48}x^6 - \frac{1}{384}x^8 - \frac{13}{3840}x^{10} - \frac{47}{46080}x^{12}$$
$$+ \frac{73}{645120}x^{14} + \frac{2447}{10321920}x^{16} + \frac{16811}{185794560}x^{18} - \frac{15551}{3715891200}x^{20} + O(x^{22})$$

The Taylor series is easily applied to a higher-order equation. For example, if we are given

$$y'' = 3 + x - y^2, \qquad y(0) = 1, \qquad y'(0) = -2,$$

we can find the derivative terms in the Taylor series as follows:

$y(0)$, and $y'(0)$ are given by the initial conditions.

$y''(0)$ comes from substitution into the differential equation from $y(0)$ and $y'(0)$.

$y'''(0)$ and higher derivatives are found by differentiating the equation for the previous order of derivative and substituting previously computed values.

6.2 The Euler Method and Its Modifications

The first truly numerical method that we discuss is the Euler method. We can solve the differential equation

$$dy/dx = f(x, y), \qquad y(x_0) = y_0,$$

by using just one term of the Taylor-series method:

$$y(x) = y(x_0) + y'(x_0)\,(x - x_0) + \text{error},$$
$$\text{error} = (h^2/2)y''(\xi) = O(h^2).$$

This is known as the Euler method. In effect, we project along the tangent line from the starting point, $y(x_0)$. If the increment to x, $(x - x_0) = h$, is small enough, the error will be small. Once we have y at $x_0 + h$, we can repeat to get more y-values:

$$y_{n+1} = y_n + hy'_n + O(h^2).* \tag{6.3}$$

The method is easy to program for we know the formula for $y'(x)$ and a starting value, $y_0 = y\,(x_0)$.

---

* This error is just the local error. Over many steps, the global error becomes $O(h)$.

Table 6.2

x_n	y_n	y'_n	hy'_n
0.0	−1.00000	1.00000	0.10000
0.1	−0.90000	0.70000	0.07000
0.2	−0.83000	0.43000	0.04300
0.3	−0.78700	0.18700	0.01870
0.4	−0.76830	−0.03170	

(Analytical answer is −0.81096, error is −0.04266.)

To see this in action, we apply it to the sample equation:

$$\frac{dy}{dx} = -2x - y, \qquad y(0) = -1,$$

where the computation can be done rather simply. It is convenient to arrange the work as in Table 6.2. Here we take $h = 0.1$.

Each of the y_n values is computed using Eq. (6.3), adding hy'_n and y_n of the previous line. Comparing the last result to the analytical answer $y(0.40) = -0.81096$, we see that there is only one-decimal-place accuracy, even though we have advanced the solution only four steps! To gain four-decimal-place accuracy, we must reduce the error by more than 400-fold. Because the global error is about proportional to h, we will need to reduce the step size about 426-fold, to <0.00024.

Improving the Simple Euler Method

The trouble with this most simple method is its lack of accuracy, requiring an extremely small step size. Figure 6.1 suggests how we might improve this method with just a little additional effort.

In the simple Euler method, we use the slope at the beginning of the interval, y'_n, to determine the increment to the function. This technique would be correct only if the function were linear. What we need instead is the correct average slope within the interval. This can be approximated by the mean of the slopes at both ends of the interval.

Suppose we use the arithmetic average of the slopes at the beginning and end of the interval to compute y_{n+1}:

$$y_{n+1} = y_n + h\,\frac{y'_n + y'_{n+1}}{2}. \tag{6.4}$$

This should give us an improved estimate for y at x_{n+1}. However, we are unable to employ Eq. (6.4) directly, because the derivative is a function of both x and y and we cannot evaluate y'_{n+1} with the true value of y_{n+1} unknown. The modified Euler method works around this problem by estimating or "predicting" a value of y_{n+1} by the simple Euler relation, Eq. (6.3). It then uses this value to compute y'_{n+1}, giving an improved estimate

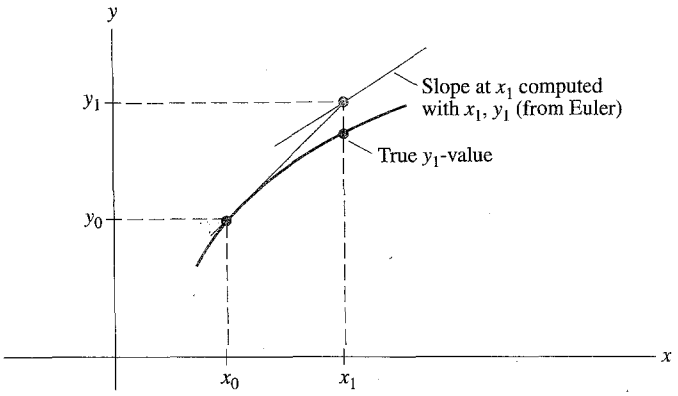

Figure 6.1

(a "corrected" value) for y_{n+1}. Because the "predicted" value for y_{n+1} is not usually very accurate, the value for y'_{n+1} that we compute from it is also inaccurate. One might be tempted to recorrect, using the first "corrected" value to recompute y'_{n+1} to get a better value for y'_{n+1} and repeat this until there is no significant change. However, this is less efficient than using a more powerful method, as we describe in the next section.

Table 6.3 shows the results of this modified Euler method on this same problem, $dy/dx = -2x - y$, $y(0) = -1$.

We can find the error of the modified Euler method by comparing it with the Taylor series:

$$y_{n+1} = y_n + y'_n h + \frac{1}{2} y''_n h^2 + \frac{y'''(\xi)}{6} h^3, \qquad x_n < \xi < x_n + h.$$

Replace the second derivative by the forward-difference approximation for y'', $(y'_{n+1} - y'_n)/h$, which has error of $O(h)$, and write the error term as $O(h^3)$:

$$y_{n+1} = y_n + h\, y'_n + \frac{1}{2}\left[\frac{y'_{n+1} - y'_n}{h} + O(h)\right] h^2 + O(h^3),$$

Table 6.3

x_n	y_n	hy'_n	$y_{n+1,p}$	$hy'_{n+1,p}$	hy'_{av}	$y_{n+1,c}$
0.0	−1.0000	0.1000	−0.9000	0.0700	0.0850	−0.9150
0.1	−0.9150	0.0715	−0.8435	0.0444	0.0579	−0.8571
0.2	−0.8571	0.0457	−0.8114	0.0211	0.0334	−0.8237
0.3	−0.8237	0.0224	−0.8013	0.0001	0.0112	−0.8124
0.4	−0.8124	0.0012	−0.8112	−0.0189	−0.0088	−0.8212
0.5	−0.8212					

[$y(0.5) = -0.81959$, the analytical value]

$$y_{n+1} = y_n + h\left(y'_n + \frac{1}{2}y'_{n+1} - \frac{1}{2}y'_n\right) + O(h^3),$$

$$y_{n+1} = y_n + h\left(\frac{y'_n + y'_{n+1}}{2}\right) + O(h^3).$$

This shows that the error of one step of the modified Euler method is $O(h^3)$. This is the local error. There is an accumulation of errors from step to step, so that the error over the whole range of application, the so-called global error, is $O(h^2)$. This seems intuitively reasonable, because the number of steps into which the interval is subdivided is proportional to $1/h$; hence the order of error is reduced to $O(h^2)$ on continuing the technique.

Another Way to Improve the Euler Method

The technique that we have called the modified Euler method tries to find a value for the average slope of y between x_n and $x_n + h$ by averaging the slopes at x_n and at x_{n+1}. There are other ways to do this. The *midpoint method* uses the slope at the midpoint of the interval as the average slope. It uses the simple Euler method to estimate y at $x + h/2$ and evaluates y' at the midpoint with this. For some derivative functions this is better than modified Euler and for others it is less accurate; for the example used to construct Tables 6.2 and 6.3, this midpoint method gives precisely the same results.

Propagation of Errors

The errors that we have mentioned for these Euler methods are the truncation errors, those due to truncating the Taylor series on which they are based. There are other errors; round off in particular will enter. It is important to understand that errors made early in the process will also affect the later computations—the early error will be propagated. The analysis of propagated error is not easy. We do it here only for the simple Euler method—this will indicate how such analysis can be accomplished.

We consider the first-order equation $dy/dx = f(x, y)$, $y(x_0) = y_0$. Let

$$Y_n = \text{calculated value at } x_n,$$

$$y_n = \text{true value at } x_n,$$

$$e_n = y_n - Y_n = \text{error in } Y_n;\ y_n = Y_n + e_n.$$

By the Euler algorithm,

$$Y_{n+1} = Y_n + hf(x_n, Y_n).$$

By Taylor series,

$$y_{n+1} = y_n + hf(x_n, y_n) + \frac{h^2}{2}y''(\xi_n), \qquad x_n < \xi_n < x_n + h,$$

$$e_{n+1} = y_{n+1} - Y_{n+1} = y_n - Y_n + h[f(x_n, y_n) - f(x_n, Y_n)] + \frac{h^2}{2}y''(\xi_n) \qquad (6.5)$$

$$= e_n + h\frac{f(x_n, y_n) - f(x_n, Y_n)}{y_n - Y_n}(y_n - Y_n) + \frac{h^2}{2}y''(\xi_n)$$

$$= e_n + hf_y(x_n, \eta_n)e_n + \frac{h^2}{2}y''(\xi_n), \qquad \eta_n \text{ between } y_n, Y_n.$$

In Eq. (6.5), we have used the mean-value theorem, imposing continuity and existence conditions on $f(x, y)$ and f_y. We suppose, in addition, that the magnitude of f_y is bounded by the positive constant K in the region of x, y-space in which we are interested.* Hence,

$$e_{n+1} \leq (1 + hK)e_n + \frac{1}{2}h^2y''(\xi_n). \tag{6.6}$$

Here, $y(x_0) = y_0$ is our initial condition, which we assume free of error. Because $Y_0 = y_0$, $e_0 = 0$:

$$e_1 \leq (1 + hK)e_0 + \frac{1}{2}h^2y''(\xi_0) = \frac{1}{2}h^2y''(\xi_0),$$

$$e_2 \leq (1 + hK)\left[\frac{1}{2}h^2y''(\xi_0)\right] + \frac{1}{2}h^2y''(\xi_1) = \frac{1}{2}h^2[(1 + hK)y''(\xi_0) + y''(\xi_1)].$$

Similarly,

$$e_3 \leq \frac{1}{2}h^2[(1 + hK)^2y''(\xi_0) + (1 + hK)y''(\xi_1) + y''(\xi_2)],$$

$$e_n \leq \frac{1}{2}h^2[(1 + hK)^{n-1}y''(\xi_0) + (1 + hK)^{n-2}y''(\xi_1) + \cdots + y''(\xi_{n-1})].$$

If $f_y \leq K$ is positive, the truncation error at every step is propagated to every later step after being amplified by the factor $(1 + hf_y)$ each time. Note that as $h \to 0$, the error at any point is just the sum of all the previous errors. If the f_y are negative and of magnitude such that $|hf_y| < 2$, the errors are propagated with diminishing effect.

We now show that the accumulated error after n steps is $O(h)$; that is, the global error of the simple Euler method is $O(h)$. We assume, in addition, that y'' is bounded, $|y''(x)| < M$, $M > 0$. After taking absolute values, Eq. (6.6) becomes

$$|e_{n+1}| \leq (1 + hK)|e_n| + \frac{1}{2}h^2M.$$

Now we compare to the first-order difference equation:

$$Z_{n+1} = (1 + hK)Z_n + \frac{1}{2}h^2M,$$
$$Z_0 = 0. \tag{6.7}$$

* This is essentially the same as the Lipschitz condition, which will guarantee existence and uniqueness of a solution.

Obviously the values of Z_n are at least equal to the magnitudes of $|e_n|$. The solution to Eq. (6.7) is (check by direct substitution)

$$Z_n = \frac{hM}{2K}(1 + hK)^n - \frac{hM}{2K}.$$

The Maclaurin expansion of e^{hk} is

$$e^{hK} = 1 + hK + \frac{(hK)^2}{2} + \frac{(hK)^3}{6} + \cdots,$$

so that

$$1 + hK < e^{hK} \quad (K > 0),$$

$$Z_n < \frac{hM}{2k}(e^{hK})^n - \frac{hM}{2K} = \frac{hM}{2K}(e^{nhK} - 1)$$
$$= \frac{hM}{2K}(e^{(x_n - x_0)K} - 1) = O(h).$$

It follows that the global error e_n is $O(h)$. (This result can be derived without difference equations.)

6.3 Runge–Kutta Methods

The simple Euler method comes from using just one term from the Taylor series for $y(x)$ expanded about $x = x_0$. The modified Euler method can be derived from using two terms:

$$y(x_0 + h) = y(x_0) + y'(x_0) * h + y''(x_0) * h^2/2.$$

If we replace the second derivative with a backward-difference approximation,

$$y(x_0 + h) = y(x_0) + y'(x_0) * h + [(y'(x_0 + h) - y'(x_0))/h] * h^2/2$$
$$= y(x_0) + \frac{y'(x_0) + y'(x_0 + h)}{2} h,$$

we get the formula for the modified method. What if we use more terms of the Taylor series? Two German mathematicians, Runge and Kutta, developed algorithms from using more than two terms of the series. We will consider only fourth- and fifth-order formulas. The modified Euler method is a second-order Runge–Kutta method.

Second-order Runge–Kutta methods are obtained by using a weighted average of two increments to $y(x_0)$, k_1 and k_2. For the equation $dy/dx = f(x, y)$:

$$\begin{aligned} y_{n+1} &= y_n + ak_1 + bk_2, \\ k_1 &= hf(x_n, y_n), \\ k_2 &= hf(x_n + \alpha h, y_n + \beta k_1). \end{aligned} \tag{6.8}$$

We can think of the values k_1 and k_2 as estimates of the change in y when x advances by h, because they are the product of the change in x and a value for the slope of the curve, dy/dx. The Runge–Kutta methods always use the simple Euler estimate as the first estimate of Δy; the other estimate is taken with x and y stepped up by the fractions α and β of h and of the earlier estimate of Δy, k_1. Our problem is to devise a scheme of choosing the four parameters, a, b, α, β. We do so by making Eq. (6.8) agree as well as possible with the Taylor-series expansion, in which the y-derivatives are written in terms of f, from $dy/dx = f(x, y)$,

$$y_{n+1} = y_n + hf(x_n, y_n) + \frac{h^2}{2} f'(x_n, y_n) + \cdots.$$

An equivalent form, because $df/dx = f_x + f_y\, dy/dx = f_x + f_y f$, is

$$y_{n+1} = y_n + hf_n + h^2\left(\frac{1}{2}f_x + \frac{1}{2}f_y f\right)_n. \tag{6.9}$$

[All the derivatives in Eq. (6.9) are calculated at the point (x_n, y_n).] We now rewrite Eq. (6.9) by substituting the definitions of k_1 and k_2:

$$y_{n+1} = y_n + ahf(x_n, y_n) + bhf[x_n + \alpha h, y_n + \beta hf(x_n, y_n)]. \tag{6.10}$$

To make the last term of Eq. (6.10) comparable to Eq. (6.9), we expand $f(x, y)$ in a Taylor series in terms of x_n, y_n, remembering that f is a function of two variables,* retaining only first derivative terms:

$$f[x_n + \alpha h, y_n + \beta hf(x_n, y_n)] \approx (f + f_x \alpha h + f_y \beta hf)_n. \tag{6.11}$$

On the right side of both Eqs. (6.9) and (6.11) f and its partial derivatives are all to be evaluated at (x_n, y_n).

Substituting from Eq. (6.11) into Eq. (610), we have

$$y_{n+1} = y_n + ahf_n + bh(f + f_x \alpha h + f_y \beta hf)_n,$$

or, rearranging,

$$y_{n+1} = y_n + (a + b)hf_n + h^2(\alpha b f_x + \beta b f_y f)_n. \tag{6.12}$$

Equation (6.12) will be identical to Eq. (6.9) if

$$a + b = 1, \qquad \alpha b = \frac{1}{2}, \qquad \beta b = \frac{1}{2}.$$

* Appendix A will remind readers of this expansion.

Note that only three equations need to be satisfied by the four unknowns. We can choose one value arbitrarily (with minor restrictions); hence, we have a set of second-order methods.

One choice can be $a = 0$, $b = 1$; $\alpha = 1/2$, $\beta = 1/2$. This gives the midpoint method. Another choice can be $a = 1/2$, $b = 1/2$; $\alpha = 1$, $\beta = 1$, which give the modified Euler. Still another possibility is $a = 1/3$, $b = 2/3$, $\alpha = 3/4$, $\beta = 3/4$; this is said to give a minimum bound to the error. All of these are special cases of second-order Runge–Kutta methods.

Fourth-order Runge–Kutta methods are most widely used and are derived in similar fashion. Greater complexity results from having to compare terms through h^4, and this gives a set of 11 equations in 13 unknowns. The set of 11 equations can be solved with 2 unknowns being chosen arbitrarily. The most commonly used set of values leads to the procedure:

$$
\begin{aligned}
y_{n+1} &= y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4),\\
k_1 &= hf(x_n, y_n),\\
k_2 &= hf\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_1\right),\\
k_3 &= hf\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_2\right),\\
k_4 &= hf(x_n + h, y_n + k_3).
\end{aligned}
\tag{6.13}
$$

Using Eqs. (6.13) to apply the Runge–Kutta fourth order to the problem, $dy/dx = -2x - y$, $y(0) = -1$ with $h = 0.1$, we obtain the results shown in Table 6.4. The results here are very impressive compared to those given in Table 6.1, where we computed the values using the terms of the Taylor series up to the h^4 term. Table 6.4 agrees to five decimals with the analytical result—illustrating a further gain in accuracy with less effort than with the Taylor-series method of Section 6.1—and it certainly is better than the Euler or modified Euler methods.

Table 6.4

x	y	k_1	k_2	k_3	k_4	k_{avg}
0.0	−1.00000	0.1000	0.0850	0.0858	0.0714	0.0855
0.1	−0.91451	0.0715	0.0579	0.0586	0.0456	0.0584
0.2	−0.85619	0.0456	0.0333	0.0340	0.0222	0.0338
0.3	−0.82246	0.0222	0.0111	0.0117	0.0011	0.0115
0.4	−0.81096	0.0011	−0.0090	−0.0085	−0.0181	−0.0086
0.5	−0.81959	−0.0180	−0.0271	−0.0267	−0.0354	−0.0268
0.6	−0.84644					

(The analytical value of $y(0.6)$ is −0.846434.)

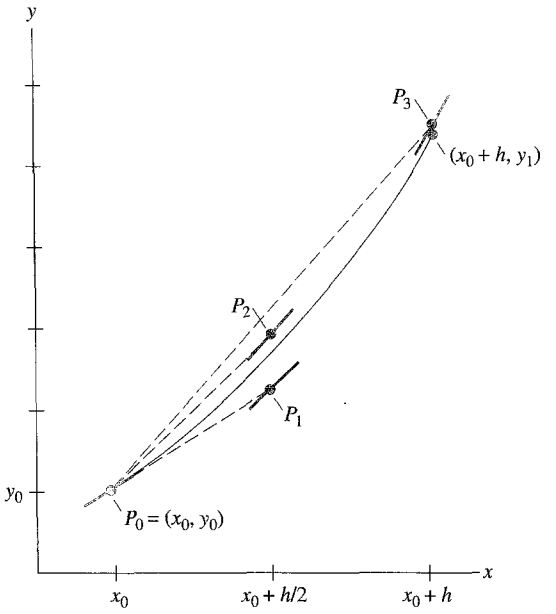

Figure 6.2

Figure 6.2 illustrates the four slope values that are combined in the four k's of the Runge–Kutta method.

The local error term for the fourth-order Runge–Kutta method is $O(h^5)$; the global error would be $O(h^4)$. It is computationally more efficient than the modified Euler method because, although four evaluations of the function are required per step rather than two, the steps can be manyfold larger for the same accuracy. The Runge–Kutta techniques have been very popular, especially the fourth-order method just presented. Because going from second to fourth order was so beneficial, we may wonder whether we should use a still higher order of formula. Higher-order (fifth, sixth, and so on) Runge–Kutta formulas have been developed and can be used to advantage in determining a suitable size for h, as we will see. Still, Runge–Kutta methods of order greater than 4 have the disadvantage that the number of function evaluations that are required is greater than the order of the method, while Runge–Kutta methods of order-4 or less require the same number of evaluations as the order.

How Do We Know If the Step-Size Is Right?

One way to determine whether the Runge–Kutta values are sufficiently accurate is to recompute the value at the end of each interval with the step size halved. If only a slight change in the value of y_{n+1} occurs, the results are accepted; if not, the step must be halved again until the results are satisfactory. This procedure is very expensive, however. For

instance, to implement Eq. (6.13) this way, we would need an additional seven function evaluations to determine the accuracy of our y_{n+1}. The best case then would require 4 + 6 = 10 function evaluations to go from (x_n, y_n) to (x_{n+1}, y_{n+1}).

A different approach uses two Runge–Kutta methods of different orders. For instance, we could use one fourth-order and one fifth-order method to move from (x_n, y_n) to (x_{n+1}, y_{n+1}). We would then compare our results at y_{n+1}. The Runge–Kutta–Fehlberg method, now one of the most popular of these methods, does just this. Only six functional evaluations (versus ten) are required, and we also have an estimate of the error (the difference of the two y's at $x = x_{n+1}$):

An Algorithm for the Runge–Kutta–Fehlberg Method

Given $y' = f(x, y)$ and $y(x_n) = y_n$, to compute $y(x_{n+1}) = y_{n+1}$ where $x_{n+1} = x_x + h$, evaluate:

$$k_1 = h \cdot f(x_n, y_n),$$

$$k_2 = h \cdot f\left(x_n + \frac{h}{4}, y_n + \frac{k_1}{4}\right),$$

$$k_3 = h \cdot f\left(x_n + \frac{3h}{8}, y_n + \frac{3k_1}{32} + \frac{9k_2}{32}\right),$$

$$k_4 = h \cdot f\left(x_n + \frac{12h}{13}, y_n + \frac{1932k_1}{2197} - \frac{7200k_2}{2197} + \frac{7296k_3}{2197}\right),$$

$$k_5 = h \cdot f\left(x_n + h, y_n + \frac{439k_1}{216} - 8k_2 + \frac{3680k_3}{513} - \frac{845k_4}{4104}\right),$$

$$k_6 = h \cdot f\left(x_n + \frac{h}{2}, y_n - \frac{8k_1}{27} + 2k_2 - \frac{3544k_3}{2565} + \frac{1859k_4}{4104} - \frac{11k_5}{40}\right);$$

$$\hat{y}_{n+1} = y_n + \left(\frac{25k_1}{216} + \frac{1408k_3}{2565} + \frac{2197k_4}{4104} - \frac{k_5}{5}\right), \text{ with global error } O(h^4),$$

$$y_{n+1} = y_n + \left(\frac{16k_1}{135} + \frac{6656k_3}{12825} + \frac{28561k_4}{56430} - \frac{9k_5}{50} + \frac{2k_6}{55}\right),$$

with global error $O(h^5)$;

$$\text{Error, } E = \frac{k_1}{360} - \frac{128k_3}{4275} - \frac{2197k_4}{75240} + \frac{k_5}{50} + \frac{2k_6}{55}.$$

The basis for the Runge–Kutta–Fehlberg scheme is to compute two Runge–Kutta estimates for the new value of y_{n+1} but of different orders of errors. Thus, instead of comparing estimates of y_{n+1} for h and $h/2$, we compare the estimates $\hat{y}_{n+1}$ and y_{n+1} using fourth- and fifth-order (global) Runge–Kutta formulas. Moreover, both equations make use of the same k's, so only six function evaluations are needed versus the previous 11. In

addition, one can increase or decrease h depending on the value of the estimated error. As our estimate for the new y_{n+1}, we use the fifth-order (global) estimate.

As an example, we once more solve $dy/dx = -2x - y$, $y(0) = -1$ with $h = 0.1$, using the Runge–Kutta–Fehlberg method:

$$k_1 = 0.1,$$
$$k_2 = 0.0925000,$$
$$k_3 = 0.0889609,$$
$$k_4 = 0.0735157,$$
$$k_5 = 0.0713736,$$
$$k_6 = 0.0853872,$$

$\hat{y}_1 = -0.914512212$, $y_1 = -0.914512251$, Error, $E = -0.000000040$.

The exact value is $y(0.1) = -0.914512254$. Thus, on the first step, y_1 agrees with the exact answer to eight decimal places with only two additional function evaluations. Moreover, we have the value E to adjust our step size for the next iteration. Of course, we would use the more accurate y_{n+1} for the next step. This algorithm is well documented and implemented in the FORTRAN program, RKF45, of Forsythe, Malcolm, and Moler (1977). MATLAB has two numerical procedures `ode45` and `ode23`. Maple has `rkf45` in its arsenal to get the numerical solution to differential equations.

A summary and comparison of the numerical methods we have studied for solving $y' = f(x, y)$ is presented in Table 6.5.

To see empirically that the global errors of Table 6.5 hold, again consider the example $dy/dx = -2x - y$, $y(0) = -1$. Table 6.6 shows how the errors of $y(0.4)$ decrease as h is halved. The table shows the ratios of errors of successive calculations.

In Table 6.6, we obtain the second row in this way: For a step size of $h = 0.2$, we compute the errors in the values for y at $x = 0.4$ using the three methods indicated at the top of columns two through four. We write down the values of the differences between the computed value and the analytical value. The last three columns represent the ratio between the previous error (larger step size h) and the current. For instance, the 3.3 in the second row is the ratio of 2.11E-01/9.10E-2 for the errors from Euler's method for $h = 0.4$ and $h = 0.2$. We do the same for the modified Euler method and the Runge–Kutta fourth-order method in columns six and seven. We see that as h gets smaller, the last three columns approach the

Table 6.5

Method	Estimate of slope	Global error	Local error	Evaluations per step
Euler	Initial value	$O(h)$	$O(h^2)$	1
Modified Euler	Average, initial and final	$O(h^2)$	$O(h^3)$	2
Midpoint	Midpoint of interval	$O(h^2)$	$O(h^3)$	2
Runge–Kutta (fourth-order)	Weighted average, four values	$O(h^4)$	$O(h^5)$	4
Runge–Kutta–Fehlberg	Weighted average, six values	$O(h^5)$	$O(h^6)$	6

Table 6.6

	Error in value computed at $x = 0.4$			Ratios of successive errors		
h	Euler	Modified Euler	Runge–Kutta 4th	Euler	Modified Euler	Runge–Kutta 4th
0.4000	2.11E-01	2.90E-02	2.40E-04			
0.2000	9.10E-02	6.42E-03	1.27E-05	2.3	4.5	18.9
0.1000	4.27E-02	1.44E-03	7.29E-07	2.1	4.5	17.4
0.0500	2.07E-02	3.48E-04	4.37E-08	2.1	4.1	16.7
0.0250	1.02E-02	8.54E-05	2.76E-09	2.0	4.1	15.8
0.0125	5.06E-03	2.11E-05	1.65E-10	2.0	4.0	16.7

ratios of 2.0, 4.0, and 16.0. This is what we expect, because these three methods are, respectively, $O(h)$, $O(h^2)$, and $O(h^4)$ and because at each stage the step size is halved.

We end this section by showing the Runge–Kutta–Merson method, another fourth-order method even though five different k's must be computed. It can be seen from the formula that the order is given, not by the number of k's, but by the global error.

$$k_1 = h \cdot f(x_n, y_n),$$

$$k_2 = h \cdot f\left(x_n + \frac{h}{3}, y_n + \frac{k_1}{3}\right),$$

$$k_3 = h \cdot f\left(x_n + \frac{h}{3}, y_n + \frac{k_1}{6} + \frac{k_2}{6}\right),$$

$$k_4 = h \cdot f\left(x_n + \frac{h}{2}, y_n + \frac{k_1}{8} + \frac{3k_3}{8}\right),$$

$$k_5 = h \cdot f\left(x_n + h, y_n + \frac{k_1}{2} - \frac{3k_3}{2} + 2k_4\right);$$

$$y_{n+1} = y_n + \frac{(k_1 + 4k_4 + k_5)}{6} + O(h^5);$$

$$\text{Error, } E = \frac{1}{30}(2k_1 - 9k_3 + 8k_4 - k_5).$$

As we have already indicated, there are methods that use Runge–Kutta formulas of orders 5, 6, and higher. In fact, the IMSL routine DVERK uses formulas or orders 5 and 6 that were developed by J. H. Verner. In this case, the method uses eight function evaluations. Maple has an option in its procedure for solving differential equations that is called `dverk78`.

Although the Runge–Kutta method has been very popular in the past, it has its limitations in solving certain types of differential equations. However, for a large class of problems the methods presented in this section produce some very stunning results. Also

the technique introduced by Fehlberg in comparing two different orders rather than halving step sizes increases the efficiency of the Runge–Kutta methods.

The methods so far discussed are called single-step methods. They use only the information at (x_n, y_n) to get to (x_{n+1}, y_{n+1}). In the next sections, we examine methods that utilize past information from previous points to get (x_{n+1}, y_{n+1}).

Here is the MATLAB solution to our sample problem through its `ode45` command, which uses the RKF method with the step size automatically adjusted. We first create an M-file that defines the derivative function:

```
function dydx = deq1(x,y)
dydx = -2*x - y;
```

Now we use the 'ode45' command to get the solution between $x = 0$ and $x = 0.6$ using the RKF method:

```
EDU>> [x,y] = ode45(@deq1, [0,.6], -1)
```

and MATLAB displays a list of the x-values used in the computations followed by the corresponding y-values. Though not apparent here, the procedure uses automatic step-size adjustment. We show only a portion of the whole output; the default of 40 intervals is used. We show the y-values side by side with the x-values. (The solution is much more accurate than four digits.)

```
x =              y =
     0             -1.0000
  0.0150           -0.9853
  0.0300           -0.9713
  0.0450           -0.9580
  0.0600           -0.9453
     .                .
     .                .
     .                .
     .                .
  0.5100           -0.8215
  0.5250           -0.8247
  0.5400           -0.8282
  0.5550           -0.8322
  0.5700           -0.8366
  0.5850           -0.8413
  0.6000           -0.8464
```

6.4 Multistep Methods

Runge–Kutta-type methods (which include Euler and modified Euler as special cases) are called single-step methods because they use only the information from the last step computed. In this, they have the ability to perform the next step with a different step size and are ideal for beginning the solution where only the initial conditions are available. After

the solution has begun, however, there is additional information available about the function (and its derivative) if we are wise enough to retain it in the memory of the computer. A multistep method is one that takes advantage of this fact.

The principle behind a multistep method is to utilize the past values of y and/or y' to construct a polynomial that approximates the derivative function, and extrapolate this into the next interval. Most methods use equispaced past values to make the construction of the polynomial easy. The Adams method is typical.* The number of past points that are used sets the degree of the polynomial and is therefore responsible for the truncation error. The order of the method is equal to the power of h in the global error term of the formula, which is also equal to one more than the degree of the polynomial.

To derive the relations for the Adams method, we write the differential equation $dy/dx = f(x, y)$ in the form

$$dy = f(x, y)\, dx,$$

and we integrate between x_n and x_{n+1}:

$$\int_{x_n}^{x_{n+1}} dy = y_{n+1} - y_n = \int_{x_n}^{x_{n+1}} f(x, y)\, dx.$$

To integrate the term on the right, we approximate $f(x, y)$ as a polynomial in x, deriving this by making it fit at several past points. If we use three past points, the approximating polynomial will be a quadratic. If we use four points, it will be a cubic. The more points we use, the better the accuracy (until round off interferes, of course).

You saw in Chapter 3 how interpolating polynomials can be developed. *Mathematica* can do this for us with its `Interpolating Polynomial` function. With this, we can get a quadratic approximation:

$$f(x, y) = \frac{1}{2} h^2(f_n - 2f_{n-1} + f_{n-2})x^2 + \frac{1}{2} h(3f_n - 4f_{n-1} + f_{n-2})x + f_n.$$

Now we again use *Mathematica* to integrate between the limits of $x = x_n$ and $x = x_{n+1}$. The result is a formula for the increment in y:

$$y_{n+1} - y_n = \frac{h}{12}(23f_n - 16f_{n-1} + 5f_{n-2}),$$

and we have the formula to advance y:

$$y_{n+1} - y_n = \frac{h}{12}[23f_n - 16f_{n-1} + 5f_{n-2}] + O(h^4). \qquad \textbf{(6.14)}$$

* This is often called the Adams–Bashford method.

Observe that Eq. (6.14) resembles the single-step formulas of the previous sections in that the increment to y is a weighted sum of the derivatives times the step size, but differs in that past values are used rather than estimates in the forward direction.

EXAMPLE 6.1 We illustrate the use of Eq. (6.14) to calculate $y(0.6)$ for $dy/dx = -2x - y$, $y(0) = -1$. We compute good values for $y(0.2)$ and $y(0.4)$ using a single-step method. In this case we obtain these values using the Runge–Kutta–Fehlberg method with $h = 0.2$. These values are given in Table 6.7.

Then, from Eq. (6.14), we have

$$y(0.6) = -0.81096 + \frac{0.2}{12}[23(0.01096) - 16(0.45619) + 5(1.0)]$$
$$= -0.84508.$$

Comparing our result with the exact solution (-0.84643), we find that the computed value has an error of 0.00135. We can reduce the size of the error by doing the calculations with a smaller step size of 0.1. We use the fifth-order values of the Runge–Kutta–Fehlberg method once again to obtain the values in Table 6.8.

Using Eq. (6.14) again with the values for $f(x, y)$ at $x = 0.3$, $x = 0.4$, $x = 0.5$ from Table 6.8, we recompute $y(0.6)$:

$$y(0.6) = -0.81959 + \frac{0.1}{12}[23(-0.18041) - 16(0.01096) + 5(0.22245)]$$
$$= -0.84636,$$

which has an error of 0.00007.

Adams Fourth-Order Formula

Equation (6.14) is a third-order formula that uses y-values at three past points, x_n, x_{n-1}, and x_{n-2}, to estimate y_{n+1}. Using four past points is equivalent to integrating a cubic interpolating polynomial through four past points. We can use the method of undetermined coefficients to obtain this.

Table 6.7

x	y	y, analytical	$f(x, y)$
0.0	−1.0000000	−1.0000000	1.0000000
0.2	−0.8561921	−0.8561923	0.4561921
0.4	−0.8109599	−0.8109601	0.0109599

Table 6.8

x	y	y, analytical	$f(x,y)$
0.0	−1.00000	−1.00000	1.00000
0.1	−0.91451	−0.91451	0.71451
0.2	−0.85619	−0.85619	0.45619
0.3	−0.82245	−0.82245	0.22245
0.4	−0.81096	−0.81096	0.01096
0.5	−0.81959	−0.81959	−0.18041

We desire a formula of the form

$$\int_{x_n}^{x_{n+1}} f(x)\,dx = c_0 f_{n-3} + c_1 f_{n-2} + c_2 f_{n-1} + c_3 f_n.$$

With four constants, we can make the formula exact when $f(x)$ is any polynomial of degree-3 or less. Accordingly, we replace $f(x)$ successively by x^3, x^2, x, and 1 to evaluate the coefficients.

It is apparent that the formula must be independent of the actual x-values. To simplify the equations, let us shift the origin to the point $x = x_n$; our integral is then taken over the interval from 0 to h, where $h = x_{n+1} - x_n$:

$$\int_0^h f(x)\,dx = c_0 f(-3h) + c_1 f(-2h) + c_2 f(-h) + c_3 f(0).$$

Carrying out the computations by replacing $f(x)$ with the particular polynomials, we have

$$\frac{h^4}{4} = c_0(-3h)^3 + c_1(-2h)^3 + c_2(-h)^3 + c_3(0),$$

$$\frac{h^3}{3} = c_0(-3h)^2 + c_1(-2h)^2 + c_2(-h)^2 + c_3(0),$$

$$\frac{h^2}{2} = c_0(-3h) + c_1(-2h) + c_2(-h) + c_3(0),$$

$$h = c_0(1) + c_1(1) + c_2(1) + c_3(1).$$

The linear system we are to solve is

$$\begin{bmatrix} -27 & -8 & -1 & 0 \\ 9 & 4 & 1 & 0 \\ -3 & -2 & -1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_4 \end{bmatrix} = \begin{bmatrix} 1/4 \\ 1/3 \\ 1/2 \\ 1 \end{bmatrix}$$

whose solution is

$$c_0 = -9/24,\ c_1 = 37/24,\ c_2 = -59/24,\ c_3 = 55/24.$$

The fourth-order Adams formula is then

Table 6.9

Number of points used	Estimate of $y(0.6)$	Error ($h = 0.1$)
3	−0.8463626	0.000072
4	−0.8464420	0.000007

$$y_{n+1} = y_n + \frac{h}{24}[55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3}] + O(h^5). \tag{6.15}$$

If we repeat Example 6.1 with this fourth-order formula, taking values at $x = 0.2, 0.3, 0.4$, and 0.5, we compute:

$$\begin{aligned} y(0.6) &= -0.81959 + \frac{0.1}{24}[55(-0.18041) - 59(0.01096) \\ &\quad + 37(0.22245) - 9(0.45619)] \\ &= -0.84644. \end{aligned}$$

The error of this computation has been reduced to 0.00001. We summarize the results of these two formulas in Table 6.9.

The Error Term We get the error term for the fourth-order Adams formula by integrating the error of the cubic interpolating polynomial. This turns out to be

$$\text{Error} = \frac{251}{720} h^5 y^{(5)}(\xi),$$

which is $O(h^5)$ as we have used before.

The Adams–Moulton Method

An improvement over the Adams method is the Adams–Moulton method. It uses the Adams method as a *predictor formula,* then applies a *corrector formula,* based on constructing another cubic interpolating formula through four points—the one obtained with the predictor formula and three previously computed points. (You may want to use undetermined coefficients to confirm this.)

Predictor:

$$y_{n+1} = y_n + \frac{h}{24}(55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3}) + \frac{251}{720} h^5 y^{(5)}(\xi_1). \tag{6.16}$$

Corrector:

$$y_{n+1} = y_n + \frac{h}{24}(9f_{n+1} + 19f_n - 5f_{n-1} + f_{n-2}) - \frac{19}{720}h^5y^{(5)}(\xi_2). \quad (6.17)$$

We illustrate the Adams–Moulton method using our earlier example, $dy/dx = -2x - y$, $y(0) = -1$. Using Eqs. (6.16) and (6.17) we construct Table 6.10. Here is how the entries in the table were obtained. By the predictor formula of (6.16), we get

$$\begin{aligned} y(0.4) &= -0.8224547 + \frac{0.1}{24}[55(0.2224547) - 59(0.4561923) \\ &\quad + 37(0.7145123) - 9(1.0)] \\ &= -0.8109687. \end{aligned}$$

Then $f(0.4, -0.8109687)$ is computed, to get 0.0109688, and we use the corrector formula of Eq. (6.17) to get

$$\begin{aligned} y(0.4) &= -0.8224547 + \frac{0.1}{24}[9(0.0109688) + 19(0.2224547) \\ &\quad -5(0.4561923) + 0.7145123] \\ &= -0.8109652. \end{aligned}$$

The computations are continued in the same manner to get $y(0.5)$. The corrected value almost agrees to five decimals with the predicted value. Comparing error terms of Eqs. (6.16) and (6.17) and assuming that the two fifth-derivative values are equal, we see that the true value should lie between the predicted and corrected values, with the error in the corrected value being about

$$\frac{19}{251 + 19} \quad \text{or} \quad \frac{1}{14.2}$$

times the difference between the predicted and corrected values. A frequently used criterion for accuracy of the Adams–Moulton method with four starting values is that the corrected value is not in error by more than 1 in the last place if the difference between

Table 6.10

x	y	$f(x, y)$	
0.0	−1.0000000	1.0000000	
0.1	−0.9145122	0.7145123	
0.2	−0.8561923	0.4561923	
0.3	−0.8224547	0.2224547	
0.4	(−0.8109687) predicted		
	(−0.8109652) corrected		(−0.8109601 analytical)
0.5	(−0.8195978) predicted		
	(−0.8195905) corrected		(−0.8195920 analytical)

predicted and corrected values is less than 14 in the last decimal place. If this degree of accuracy is not met, we know that h is too large.

Changing the Step Size

When the predicted and corrected values agree to as many decimals as the desired accuracy, we can save computational effort by increasing the step size. We can conveniently double the step size, after we have seven equispaced values, by omitting every second one. When the difference between predicted and corrected values reaches or exceeds the accuracy criterion, we should decrease step size. If we interpolate two additional y-values with a fourth-degree polynomial, where the error will be $O(h^5)$, consistent with the rest of our work, we can readily halve the step size. Convenient formulas for this are

$$y_{n-1/2} = \frac{1}{128}[35y_n + 140y_{n-1} - 70y_{n-2} + 28y_{n-3} - 5y_{n-4}],$$

$$y_{n-3/2} = \frac{1}{128}[-5y_n + 60y_{n-1} + 90y_{n-2} - 20y_{n-3} + 3y_{n-4}].$$

Use of these values with y_n, y_{n-1} gives four values of the function at intervals of $\Delta x = h/2$.

The efficiency of Adams–Moulton is about twice that of the Runge–Kutta–Fehlberg and Runge–Kutta methods. Only two function evaluations are needed per step for the former method, whereas six or four are required with the single-step alternatives. All have similar error terms. Change of step size with the multistep methods is considerably more awkward, however.

Stability Considerations

In getting the solution to a differential equation, one must always worry whether the method is *stable*. In a stable method, early errors (due to the imprecision of the method or to an initial value that is slightly incorrect) are damped out as the computations proceed; they do not grow without bound. The opposite is true for an *unstable* method.

In the discussion of the Euler method in Section 6.2, we showed the conditions for stability. This was not a simple task. It is easier to see if a method is stable or unstable by testing it with certain kinds of derivative functions, $y'(x) = f(x, y)$.

Consider this equation:

$$dy/dx = f(x, y) = -2y + 2, \qquad y(0) = -1,$$

whose analytical solution is $y(x) = 1 - 2e^{-2x}$. The curve for $y(x)$ is smooth, starting at $y = -1$, proceeding rapidly upward with a slope of 4, crossing the x-axis at about $x = 0.35$, and approaching the asymptote of $y = 1$ as x increases. By $x = 3$, the y-value is within 0.5% of its limiting values.

Suppose that we use a very simple multistep formula:

$$y_{n+1} = y_{n-1} + 2hf(x_n, y_n), \tag{6.18}$$

which has a truncation error of $(1/6)h^3y'''(\xi)$, smaller than for the simple Euler method, which is $(1/2)h^2y''(\xi)$, particularly with small values for h.

If we apply Eq. (6.18) to $y' = -2y + 2$, $y(0) = -1$, with an h-value of 0.1 we get the results in Table 6.11. (We need starting values at $x = 0$ and $x = 0.1$; these were from the given $y(0) = -1$ and the analytical value at $x = 0.1$.)

Table 6.11 Results from Eq. (6.18)

x	y	Analytical	Error	Rel error
0.20	−0.34502	−0.34064	0.00438	−0.01284
0.30	−0.09946	−0.09762	0.00183	−0.01877
0.40	0.09477	0.10134	0.00658	0.06488
0.50	0.26264	0.26424	0.00160	0.00607
0.60	0.38971	0.39761	0.00790	0.01987
0.70	0.50675	0.50681	0.00005	0.00010
0.80	0.58701	0.59621	0.00920	0.01543
0.90	0.67195	0.66940	−0.00255	−0.00380
1.00	0.71823	0.72933	0.01110	0.01522
1.10	0.78466	0.77839	−0.00626	−0.00805
1.20	0.80437	0.81856	0.01420	0.01734
1.30	0.86291	0.85145	−0.01146	−0.01346
1.40	0.85920	0.87838	0.01918	0.02183
1.50	0.91923	0.90043	−0.01880	−0.02088
1.60	0.89151	0.91848	0.02696	0.02936
1.70	0.96262	0.93325	−0.02937	−0.03147
1.80	0.90646	0.94535	0.03889	0.04114
1.90	1.00004	0.95526	−0.04478	−0.04688
2.00	0.90645	0.96337	0.05692	0.05909
2.10	1.03746	0.97001	−0.06745	−0.06954
2.20	0.89146	0.97545	0.08398	0.08610
2.30	1.08087	0.97990	−0.10098	−0.10305
2.40	0.85911	0.98354	0.12443	0.12651
2.50	1.13723	0.98652	−0.15070	−0.15276
2.60	0.80422	0.98897	0.18474	0.18681
2.70	1.21554	0.99097	−0.22457	−0.22662
2.80	0.71801	0.99260	0.27460	0.27664
2.90	1.32834	0.99394	−0.33439	−0.33643
3.00	0.58667	0.99504	0.40837	0.41041
3.10	1.49367	0.99594	−0.49773	−0.49976
3.20	0.38920	0.99668	0.60747	0.60950
3.30	1.73799	0.99728	−0.74071	−0.74273
3.40	0.09401	0.99777	0.90376	0.90578
3.50	2.10038	0.99818	−1.10221	−1.10422
3.60	−0.34614	0.99851	1.34465	1.34666
3.70	2.63884	0.99878	−1.64006	−1.64207
3.80	−1.00168	0.99900	2.00068	2.00269
3.90	3.43951	0.99918	−2.44033	−2.44234
4.00	−1.97749	0.99933	2.97682	2.97882

Table 6.12 Results from Simple Euler Method

x	y	Analytical	Error	Rel error
0.00	−1.00000	−1.00000	0.00000	0.00000
0.10	−0.60000	−0.63746	−0.03746	0.05877
0.20	−0.28000	−0.34064	−0.06064	0.17802
0.30	−0.02400	−0.09762	−0.07362	0.75416
0.40	0.18080	0.10134	−0.07946	−0.78406
0.50	0.34464	0.26424	−0.08040	−0.30426
0.60	0.47571	0.39761	−0.07810	−0.19642
0.70	0.58057	0.50681	−0.07376	−0.14555
0.80	0.66446	0.59621	−0.06825	−0.11447
0.90	0.73156	0.66940	−0.06216	−0.09286
1.00	0.78525	0.72933	−0.05592	−0.07668
1.10	0.82820	0.77839	−0.04981	−0.06399
1.20	0.86256	0.81856	−0.04400	−0.05375
1.30	0.89005	0.85145	−0.03860	−0.04533
1.40	0.91204	0.87838	−0.03366	−0.03832
1.50	0.92963	0.90043	−0.02921	−0.03244
1.60	0.94371	0.91848	−0.02523	−0.02747
1.70	0.95496	0.93325	−0.02171	−0.02326
1.80	0.96397	0.94535	−0.01862	−0.01969
1.90	0.97118	0.95526	−0.01592	−0.01666
2.00	0.97694	0.96337	−0.01357	−0.01409
2.10	0.98155	0.97001	−0.01154	−0.01190
2.20	0.98524	0.97545	−0.00980	−0.01004
2.30	0.98819	0.97990	−0.00830	−0.00847
2.40	0.99056	0.98354	−0.00701	−0.00713
2.50	0.99244	0.98652	−0.00592	−0.00600
2.60	0.99396	0.98897	−0.00499	−0.00504
2.70	0.99516	0.99097	−0.00420	−0.00424
2.80	0.99613	0.99260	−0.00353	−0.00355
2.90	0.99691	0.99394	−0.00296	−0.00298
3.00	0.99752	0.99504	−0.00248	−0.00249
3.10	0.99802	0.99594	−0.00208	−0.00209
3.20	0.99842	0.99668	−0.00174	−0.00174
3.30	0.99873	0.99728	−0.00145	−0.00146
3.40	0.99899	0.99777	−0.00121	−0.00122
3.50	0.99919	0.99818	−0.00101	−0.00101
3.60	0.99935	0.99851	−0.00084	−0.00085
3.70	0.99948	0.99878	−0.00070	−0.00070
3.80	0.99958	0.99900	−0.00059	−0.00059
3.90	0.99967	0.99918	−0.00049	−0.00049
4.00	0.99973	0.99933	−0.00041	−0.00041

Observe in Table 6.11 that we get good results up to about $x = 0.8$, but from $x = 2$ the computed values are increasingly poor, and as x approaches 4 they are completely useless; they oscillate widely about the asymptotic value for y.

Compare these with the results from a simple Euler computation, also with $h = 0.1$, that are given in Table 6.12. These are much less accurate at small values of x (the magnitudes

of the errors from the simple Euler computation between $x = 0.2$ and $x = 0.5$ are on the average nearly 20 times as large).

On the other hand, the Euler results closely resemble the analytical values at larger values for x and do not show the same oscillations.

The method of Eq. (6.18) is unstable while the Euler method is stable.

There is another unstable method but its instability is less apparent. *Milne's method* is a multistep predictor–corrector that uses these equations:

Predictor:

$$y_{n+1} - y_{n-3} = \frac{4h}{3}(2f_n - f_{n-1} + 2f_{n-2}) + \frac{28}{90}h^5 y^{(5)}(\xi_1), \qquad x_{n-3} < \xi_1 < x_{n+1}.$$

Corrector:

$$y_{n+1,c} - y_{n-1} = \frac{h}{3}(f_{n+1} + 4f_n + f_{n-1}) - \frac{h^5}{90}y^{(5)}(\xi_2), \qquad x_{n-1} < \xi_2 < x_{n+1}. \quad \textbf{(6.19)}$$

Observe that the error term after correcting has a multiplier that is less than half that of Adams–Moulton so we should expect very accurate results. However, if we solve the same equation,

$$dy/dx = f(x, y) = -2y + 2, \qquad y(0) = -1,$$

with the formulas of Eq. (6.19), we again observe oscillatory behavior as exhibited in Table 6.13, but the oscillations are slight and do not appear until about $x = 2$ and even at $x = 8$ they are not large but they are increasing in magnitude.

Of course, this demonstration of instability for Milne's method is not entirely satisfactory. We can do this more theoretically. Consider the differential equation

$$dy/dx = Ay,$$

where A is a constant. The general solution is $y = ce^{Ax}$. Suppose now that $y(x_0) = y_0$ is the initial condition; it then follows that the value of c must be $c = y_0 e^{-Ax_0}$. Hence, letting y_n be the value of the function when $x = x_n$, the analytical solution is

$$y_n = y_0 e^{A(x_n - x_0)}.$$

If we solve the differential equation by the method of Milne, we have, from the corrector formula,

$$y_{n+1} = y_{n-1} + \frac{h}{3}(y'_{n+1} + 4y'_n + y'_{n-1}).$$

Letting $y'_n = Ay_n$, from the original differential equation, and rearranging, we get

$$y_{n+1} = y_{n-1} + \frac{h}{3}(Ay_{n+1} + 4Ay_n + Ay_{n-1}),$$

Table 6.13 Results with Milne's method

x	y	Analytical	Error	Rel error
0.40	0.101355	0.101342	−0.000013	−0.000127
0.50	0.264249	0.264241	−0.000008	−0.000029
0.60	0.397630	0.397612	−0.000019	−0.000047
0.70	0.506816	0.506806	−0.000010	−0.000020
0.80	0.596227	0.596207	−0.000020	−0.000033
⋱	⋱	⋱	⋱	⋱
1.80	0.945365	0.945353	−0.000012	−0.000013
1.90	0.955257	0.955258	0.000002	0.000002
2.00	0.963380	0.963369	−0.000011	−0.000012
2.10	0.970006	0.970009	0.000003	0.000003
2.20	0.975456	0.975445	−0.000010	−0.000011
⋱	⋱	⋱	⋱	⋱
3.50	0.998167	0.998176	0.000010	0.000010
3.60	0.998518	0.998507	−0.000011	−0.000011
3.70	0.998767	0.998778	0.000010	0.000010
3.80	0.999010	0.998999	−0.000011	−0.000011
3.90	0.999169	0.999181	0.000011	0.000011
4.00	0.999341	0.999329	−0.000012	−0.000012
⋱	⋱	⋱	⋱	⋱
7.70	0.999968	1.000000	0.000031	0.000031
7.80	1.000032	1.000000	−0.000032	−0.000032
7.90	0.999967	1.000000	0.000033	0.000033
8.00	1.000033	1.000000	−0.000034	−0.000034
8.10	0.999965	1.000000	0.000035	0.000035

$$\left(1 - \frac{hA}{3}\right) y_{n+1} - \frac{4hA}{3} y_n - \left(1 + \frac{hA}{3}\right) y_{n-1} = 0.$$

This a second-order difference equation that has the solution:

$$y_n = C_1 Z_1^n + C_2 Z_2^n,$$

where Z_1, Z_2 are the roots of the quadratic

$$\left(1 - \frac{hA}{3}\right) Z^2 - \frac{4hA}{3} Z - \left(1 + \frac{hA}{3}\right) = 0,$$

which you may check by direct substitution. We can simplify this by letting $hA/3 = r$; the roots of the quadratic are then

$$Z_1 = \frac{2r + \sqrt{3r^2 + 1}}{1 - r},$$

$$Z_2 = \frac{2r - \sqrt{3r^2 + 1}}{1 - r}.$$

What happens if the step size h becomes small? As $h \to 0$, $r \to 0$, and $r^2 \to 0$ even faster. We then can neglect the $3r^2$ terms in comparison to 1 under the radical and get, after

dividing the fractions,

$$Z_1 \approx \frac{2r+1}{1-r} = 1 + 3r + O(r^2) = 1 + Ah + O(h^2),$$

$$Z_2 \approx \frac{2r-1}{1-r} = -1 + r + O(r^2) = -\left(1 - \frac{Ah}{3}\right) + O(h^2).$$

We now compare this to the Maclaurin series for the exponential function,

$$e^{hA} = 1 + hA + O(h^2),$$

$$e^{-hA/3} = 1 - \frac{hA}{3} + O(h^2).$$

We see that, for $h \to 0$,

$$Z_1 = e^{hA}, \qquad Z_2 = -e^{-hA/3}.$$

Hence, the Milne solution is represented by

$$y_n = C_1(e^{hA})^n + C_2(e^{-hA/3})^n = C_1 e^{A(x_n - x_0)} + C_2 e^{-A(x_n - x_0)/3}.$$

In this, we have used $x_n - x_0 = nh$. The solution consists of two parts. The first term obviously agrees with the analytical solution. The second term, called a *parasitic term*, will die out as x_n increases if A is a positive constant, but if A is negative, it will grow exponentially with x_n. Note that we get this peculiar behavior independent of h; smaller step size is of no benefit in eliminating the error.

Hamming's Method

The analysis of Milne's method shows that the instability comes from the corrector equation. Hamming describes a way to avoid this instability while still using the Milne predictor with its simplicity. Hammings equations are

Predictor:

$$y_{i+1,p} = y_{i-3} + \frac{4h}{3}(2f_i - f_{i-1} + 2f_{i-2}),$$

which is first modified as

$$y_{i+1,m} = y_{i+1,p} - \frac{112}{121}(y_{i,p} - y_{i,c}),$$

and the modified value is used in the corrector:

$$y_{i+1,c} = \frac{1}{8}[9y_i - y_{i-2} + 3h(f_{i+1,m} + 2f_i - f_{i-1})],$$

The error of this method is not as small as with Milne, but it is a little better than Adams–Moulton.

6.5 Higher-Order Equations and Systems

In the opening portion of this chapter, we pointed out that Newton's law of motion, $f = m * a$, is a differential equation with a being the acceleration, the rate of change of velocity with time. Velocity is itself the derivative of distance with time, dx/dt. So, $f = ma$ is really

$$f = m * d^2x/dt^2,$$

a second-order differential equation.

We can solve this equation numerically by changing it into a pair of first-order equations. We rearrange the equation to put the derivative on the left

$$d^2x/dt^2 = f/m,$$

and then, by letting $dx/dt = y$, a new variable, we have

$$dx/dt = y,$$
$$dy/dt = d^2x/dt^2 = f/m.$$

To solve the original second-order equation for x as a function of time, we need two initial conditions, the starting position, x_0, and the starting velocity, x_0'. So, the equation for dx/dt begins with $x = x_0$, and that for dy/dt begins with $y = y_0 = x_0'$.

Here is another example, a variation on the familiar spring-mass problem. Figure 6.3 shows our system. Mass 1 is a block that rolls along a horizontal surface and whose motion is controlled by the linear spring whose spring constant is k_1. The second mass, m_2, is a wheel of radius r_2 that rolls on the top of mass 1 and is attached to another spring whose spring constant is k_2. The equations of motion for this system are:

$$(m_1 + 0.5m_2)\frac{d^2x_1}{dt^2} - 0.5m_2\frac{d^2x_2}{dt^2} + k_1x_1 = 0,$$

$$-0.5m_2\frac{d^2x_2}{dt^2} + 1.5m_2\frac{d^2x_1}{dt^2} + k_2x_2 = 0.$$

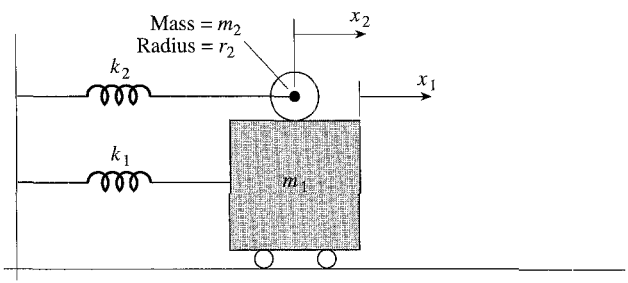

Figure 6.3

These equations make up a system of two second-order equations. To solve this problem numerically, we reduce to a system of four first-order equations by substituting dy/dt for d^2x_1/dt^2 and dz/dt for d^2x_2/dt^2. You should write out these for equations for practice. What are the four initial conditions?

Systems of First-Order Equations

It is clear that all we need to do to solve higher-order equations, even a system of higher-order initial-value problems, is to reduce them to a system of first-order equations. We illustrate how a system of first-order problems can be solved with a pair of equations whose solution at $t = 0.1$ is $x = 0.913936$, $y = -0.909217$.

$$\begin{aligned} \frac{dx}{dt} &= xy + t, \qquad x(0) = 1, \\ \frac{dy}{dt} &= ty + x, \qquad y(0) = -1. \end{aligned} \tag{6.20}$$

Taylor-Series Method

We need the various derivatives $x', x'', x''', \ldots, y', y'', y''', \ldots$, all evaluated at $t = 0$:

$$\begin{aligned} x' &= xy + t, & x'(0) &= (1)(-1) + 0 = -1 \\ y' &= ty + x, & y'(0) &= (0)(-1) + 1 = 1, \\ x'' &= xy' + x'y + 1, & x''(0) &= (1)(1) + (-1)(-1) + 1 = 3, \\ y'' &= y + ty' + x', & y''(0) &= -1 + (0)(1) - 1 = -2, \\ x''' &= x'y' + xy'' + x''y + x'y', & x'''(0) &= -7, \\ y''' &= y' + y' + ty'' + x'', & y'''(0) &= 5, \\ &\text{and so on;} & &\text{and so on;} \end{aligned}$$

$$\begin{aligned} x(t) &= 1 - t + \frac{3}{2}t^2 - \frac{7}{6}t^3 + \frac{27}{24}t^4 - \frac{124}{120}t^5 + \cdots, \\ y(t) &= -1 + t - t^2 + \frac{5}{6}t^3 - \frac{13}{24}t^4 + \frac{47}{120}t^5 + \cdots. \end{aligned} \tag{6.21}$$

At $t = 0.1$, $x = 0.9139$ and $y = -0.9092$.

Equations (6.21) are the solution to the set (6.20). Note that we need to alternate between the functions in getting the derivatives; for example, we cannot get $x''(0)$ until $y'(0)$ is known; we cannot get $y'''(0)$ until $x''(0)$ is known. After we have obtained the coefficients of the Taylor-series expansions in Eq. (6.21), we can evaluate x and y at any value of t, but the error will depend on how many terms we employ.

Euler Predictor–Corrector Method (Modified Euler)

We apply the predictor to each equation; then the corrector can be used. Again, note that we work alternately with the two functions.

Take $h = 0.1$. Let p and c subscripts indicate predicted and corrected values, respectively:

$$x_p(0.1) = 1 + 0.1[(1)(-1) + 0] = 0.9,$$
$$y_p(0.1) = -1 + 0.1[(0)(-1) + 1] = -0.9,$$
$$x_c(0.1) = 1 + 0.1\left(\frac{-1 + [(0.9)(-0.9) + 0.1]}{2}\right) = 0.9145,$$
$$y_c(0.1) = -1 + 0.1\left(\frac{1 + [(0.1)(-0.9) + 0.9145]}{2}\right) = -0.9088.$$

In computing $x_c(0.1)$, we used the x_p and y_p. In computing y_c (0.1) after $x_c(0.1)$ is known, we have a choice between x_p and x_c. There is an intuitive feel that one should use x_c, with the idea that one should always use the best available values. This does not always expedite convergence, probably due to compensating errors. Here we have used the best values to date. If we use the corrected values to recompute the value of the derivatives at $h = 0.1$, we can obtain better values. Doing so gives

$$x(0.1) = 0.9135,$$
$$y(0.1) = -0.9089,$$

but this is not as efficient as using a more powerful method. We can now advance the solution another step if desired, by using the computed values at $t = 0.1$ as the starting values. From this point, we can advance one more step, and so on for any value of t. The errors will be the combination of local truncation error at each step plus the propagated error resulting from the use of inexact starting values.

Runge–Kutta–Fehlberg Method

Again there is an alternation between the x and y calculations. In applying this method, one always uses the previous k-value in incrementing the function values and the value of h to increment the independent variable. As in the previous calculations, we alternate between computations for x and for y; for example, we do $k_{1,x}$, then $k_{1,y}$, before doing $k_{2,x}$, and so on.

Keeping in mind that the equations are

$$\frac{dx}{dt} = f(t, x, y) = xy + t, \quad x(0) = 1,$$
$$\frac{dy}{dt} = g(t, x, y) = ty + x, \quad y(0) = -1,$$

the k-values for x and y are

for x:

$$\begin{aligned}
k_{1,x} &= hf(0, 1, -1) \\
&= 0.1[(1)(-1) + 0] \\
&= -0.1; \\
k_{2,x} &= hf(0.025, 0.975, -0.975) \\
&= 0.1[(0.975)(-0.975) + 0.025] \\
&= -0.092562; \\
k_{3,x} &= hf(0.038, 0.965, -0.964) \\
&= 0.1[(0.965)(-0.964) + 0.038] \\
&= -0.089226; \\
k_{4,x} &= hf(0.092, 0.919, -0.915) \\
&= 0.1[(0.919)(-0.915) + 0.092] \\
&= -0.074892; \\
k_{5,x} &= hf(0.1, 0.913, -0.908) \\
&= 0.1[(0.913)(-0.908) + 0.1] \\
&= -0.072904; \\
k_{6,x} &= hf(0.05, 0.954, -0.953) \\
&= 0.1[(0.954)(-0.953) + 0.05] \\
&= -0.085868.
\end{aligned}$$

for y:

$$\begin{aligned}
k_{1,y} &= hg(0, 1, -1) \\
&= 0.1[(0)(-1) + 1] \\
&= 0.1; \\
k_{2,y} &= hg(0.025, 0.975, -0.975) \\
&= 0.1[(0.025)(-0.975) + 0.975] \\
&= 0.095062; \\
k_{3,y} &= hg(0.038, 0.965, -0.964) \\
&= 0.1[(0.038)(-0.964) + 0.965] \\
&= 0.092845; \\
k_{4,y} &= hg(0.092, 0.919, -0.915) \\
&= 0.1[(0.092)(-0.915) + 0.919] \\
&= 0.083461; \\
k_{5,y} &= hg(0.1, 0.913, -0.908) \\
&= 0.1[(0.1)(-0.908) + 0.913] \\
&= 0.082178; \\
k_{6,y} &= hg(0.05, 0.954, -0.953) \\
&= 0.1[(0.05)(-0.953) + 0.954] \\
&= 0.090628.
\end{aligned}$$

Then, using the fifth-order formula, we get

$$\begin{aligned}
x(0.1) &= 1 + (-0.01185 - 0.046307 - 0.037905 + 0.013123 - 0.003122) \\
&= 0.913936; \\
y(0.1) &= -1 + (0.01185 + 0.048185 + 0.042242 - 0.014792 + 0.003296) \\
&= -0.909217.
\end{aligned}$$

Extending the Taylor-series solution even further shows that the Runge–Kutta–Fehlberg values are correct to more than five decimals, whereas the modified Euler values are correct to only three, so $h = 0.1$ may be too large for that method.

Advancing the solution by the Runge–Kutta–Fehlberg method will again involve using the computed values of x and y as the initial values for another step. The errors here will be much less than those for the Euler predictor–corrector method.

Table 6.14

	t	x	x'	t	y	y'
Starting values	0.000	1.0	−1.0	0.00	−1.0	1.0
	0.025	0.9759	−0.9271	0.025	−0.9756	0.9515
	0.050	0.9536	−0.8582	0.050	−0.9524	0.9060
	0.075	0.9330	−0.7929	0.075	−0.9303	0.8632
Predicted	0.10	(0.9139)	(−0.7310)	0.10	(−0.9092)	(0.8230)
Corrected		0.9139			−0.9092	

Adams-Moulton Method

After getting four starting values, we proceed with the algorithm of Eqs. (6.16) and (6.17), again alternately computing x and then y (see Table 6.14.)

In the computations we first get predicted values of x and y:

$$x(0.1) = 0.9330 + \frac{0.025}{24}[55(-0.7929) - 59(-0.8582) + 37(-0.9271) - 9(-1.0)]$$
$$= 0.913937;$$
$$y(0.1) = -0.9303 + \frac{0.025}{24}[55(0.8632) - 59(0.9060) + 37(0.9515) - 9(1.0)]$$
$$= -0.909217.$$

After getting x' and y' at $t = 0.1$, using $x(0.1)$ and $y(0.1)$, we then correct:

$$x(0.1) = 0.9330 + \frac{0.025}{24}[9(-0.7310) + 19(-0.7929) - 5(-0.8582) + (-0.9271)]$$
$$= 0.913936;$$
$$y(0.1) = -0.9303 + \frac{0.025}{24}[9(0.8230) + 19(0.8632) - 5(0.9060) + (0.9515)]$$
$$= -0.909217.$$

The close agreement of predicted and corrected values indicates six-decimal-place accuracy.

In this method, as we advance the solution to larger values of t, the comparison between predictor and corrector values tells us whether the step size needs to be changed.

Our computer algebra systems have no trouble in solving a system of first-order equations. Here is how Maple can solve the same problem that we have used to illustrate the methods:

```
> deqs := {D(x)(t) = x(t)*y(t) + t, D(y)(t) = t*y(t) + x(t)}:
> inits := {x(0) = 1, y(0) = -1}:
> soln := dsolve(degs union inits, {x(t),y(t)}, numeric,
```

```
output = array([0, 0.1, 0.2, 0.3, 0.4]));
                    [t, x(t) y(t)]
          0          1.                  -1.
         .1    .91393569117289    -.90921691879919
soln: =  .2    .85218609746503    -.83408937511807
         .3    .81063353106742    -.77109331990007
         .4    .78634968913429    -.71735810231063
```

Here, we asked for the solution at x-values between 0 and 0.4 in steps of 0.1 and the results are given in tabular form. MATLAB and *Mathematica* can do so similarly.

6.6 Stiff Equations

Some initial value problems pose significant difficulties for their numerical solution. Acton points out several kinds of such difficulties—one of his examples is Bessel's equation:

$$y'' + y'/x + y = 0, \qquad y(0) = 1, \qquad y'(0) = 0.$$

There is a singularity at the origin, but this is surmounted by the initial value for y ($y = 0$), so that one can replace the equation at $x = 0$ and get a starting value with

$$2y'' + y = 0.$$

There are other difficult situations: The equation may change its form at certain critical points, or it may have a sharp narrow peak that will be missed if too large an interval is used.

One particular difficult case is one that we now discuss—*stiff differential equations*. The word *stiff* comes from an analogy to a spring system where the natural frequency of vibration is very great if the spring constant is large.

When the solution to a differential equation (say, of second order) has a general solution that involves the sum or difference of terms of the form ae^{ct} and be^{dt} where both c and d are negative but c is much smaller than d, the numerical solution can be very unstable even with a very small step size.

An example is the following:

$$\begin{aligned} x' &= 1195x - 1995y, \qquad x(0) = 2, \\ y' &= 1197x - 1997y, \qquad y(0) = -2. \end{aligned} \tag{6.22}$$

The analytical solution of Eq. (6.22) is

$$x(t) = 10e^{-2t} - 8e^{-800t}, \qquad -y(t) = 6e^{-2t} - 8e^{-800t}.$$

Observe that the exponents are all negative and of very different magnitude, qualifying this as a stiff equation. Suppose we solve Eq. (6.22) by the simple Euler method with $h = 0.1$, applying just one step. The iterations are

$$\begin{aligned} x_{i+1} &= x_i + hf(x_i, y_i) = x_i + 0.1(1195x_i - 1995y_i), \\ y_{i+1} &= y_i + hg(x_i, y_i) = y_i + 0.1(1197x_i - 1997y_i). \end{aligned}$$

This gives $x(0.1) = 640$, $y(0.1) = 636$, while the analytical values are $x(0.1) = 8.187$ and $y(0.1) = 4.912$. Such a result is typical (although here exaggerated) for stiff equations.

One solution to this problem is to use an implicit method rather than an explicit one. All the methods so far discussed have been explicit, meaning that new values, x_{i+1} and y_{i+1}, are computed in terms of previous values, x_i and y_i. An implicit method computes the increment only with the new (unknown) values. Suppose that

$$x' = f(x, y) \qquad \text{and} \qquad y' = g(x, y).$$

The implicit form of the Euler method is

$$\begin{aligned} x_{i+1} &= x_i + hf(x_{i+1}, y_{i+1}), \\ y_{i+1} &= y_i + hg(x_{i+1}, y_{i+1}). \end{aligned} \tag{6.23}$$

If the derivative functions $f(x, y)$ and $g(x, y)$ are nonlinear, this is difficult to solve. However, in Eq. (6.22) they are linear. Solving Eq. (6.22) by use of Eq. (6.23) we have

$$\begin{aligned} x_{i+1} &= x_i + 0.1(1195x_{i+1} - 1995y_{i+1}), \\ y_{i+1} &= y_i + 0.1(1197x_{i+1} - 1997y_{i+1}). \end{aligned}$$

The system is linear, so we can write

$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} = \begin{bmatrix} (1 - 1195(0.1)) & 1995(0.1) \\ -1197(0.1) & (1 + 1997(0.1)) \end{bmatrix}^{-1} \begin{bmatrix} x_i \\ y_i \end{bmatrix}$$

which has the solution $x(0.1) = 8.23$, $y(0.1) = 4.90$, reasonably close to the analytical values.

In summary, our results for the solution of Eq. (6.22) are

	$x(0.1)$	$y(0.1)$
Analytical	8.19	4.91
Euler		
Explicit	640	636
Implicit	8.23	4.90

If the step size is very small, we can get good results from the simpler Euler after the first step. With $h = 0.0001$, the table of results becomes

	$x(0.0001)$	$y(0.0001)$
Analytical	2.61	−1.39
Euler		
Explicit	2.64	−1.36
Implicit	2.60	−1.41

but this would require 1000 steps to reach $t = 0.1$, and round-off errors would be large.

If we anticipate some material from Section 6.8, we can give a better description of stiffness as well as indicate the derivation of the general solution to Eq. (6.22). We rewrite Eq. (6.22) in matrix form:

$$\begin{bmatrix} x \\ y \end{bmatrix}' = A\begin{bmatrix} x \\ y \end{bmatrix}, \qquad \text{where } A = \begin{bmatrix} 1195 & -1995 \\ 1197 & -1997 \end{bmatrix}.$$

The general solution, in matrix form, is

$$\begin{bmatrix} x \\ y \end{bmatrix} = ae^{-2t}v_1 + ce^{-800t}v_2,$$

where

$$v_1 = \begin{bmatrix} 5 \\ 3 \end{bmatrix} \qquad \text{and} \qquad v_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

You can easily verify that $Av_1 = -2v_1$ and $Av_2 = -800v_2$. This means that v_1 is an eigenvector of A and that -2 is the corresponding eigenvalue. Similarly, v_2 is an eigenvector of A with the corresponding eigenvalue of -800. (In Section 6.8, you will learn additional methods to find the eigenvectors and eigenvalues of a matrix.)

A stiff equation can be defined in terms of the eigenvalues of the matrix A that represents the right-hand sides of the system of differential equations. When the eigenvalues of A have real parts that are negative and differ widely in magnitude as in this example, the system is stiff. In the case of a nonlinear system

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}' = \begin{bmatrix} f_1(x_1, x_2, \ldots, x_n) \\ f_2(x_1, x_2, \ldots, x_n) \\ \vdots \\ f_n(x_1, x_2, \ldots, x_n) \end{bmatrix},$$

one must consider the Jacobian matrix whose terms are $\partial f_i/\partial x_j$. See Gear (1971) for more information.

6.7 Boundary-Value Problems

As we have seen, a second-order differential equation (or a pair of first-order problems) must have two conditions for its numerical solution. Up until now, we have considered that both of these conditions are given at the start—these are initial-value problems. That is not always the case; the given conditions may be at different points, usually at the endpoints of the region of interest. For equations of order higher than two, more than two conditions are required and these also may be at different x-values. We consider now how such problems can be solved.

Here is an example that describes the temperature distribution within a rod of uniform cross section that conducts heat from one end to the other. Look at Figure 6.4. By concentrating our attention on an element of the rod of length dx located at a distance x from the left end, we can derive the equation that determines the temperature, u, at any point along

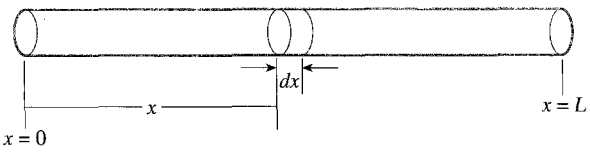

Figure 6.4

the rod. The rod is perfectly insulated around its outer circumference so that heat flows only laterally along the rod. It is well known that heat flows at a rate (measured in calories per second) proportional to the cross-sectional area (A), to a property of the material [k, its thermal conductivity, measured in cal/(sec * cm^2 * (°C/cm))], and to the temperature gradient, du/dx (measured in °C/cm), at point x. We use $u(x)$ for the temperature at point x, with x measured from the left end of the rod. Thus, the rate of flow of heat into the element (at $x = x$) is

$$-kA\left(\frac{du}{dx}\right).$$

The minus sign is required because du/dx expresses how rapidly temperatures increase with x, while the heat always flows from high temperature to low.

The rate at which heat leaves the element is given by a similar equation, but now the temperature gradient must be at the point $x + dx$:

$$-kA\left[\frac{du}{dx} + \frac{d}{dx}\left(\frac{du}{dx}\right)dx\right],$$

in which the gradient term is the gradient at x plus the change in the gradient between x and $x + dx$.

Unless heat is being added to the element (or withdrawn by some means), the rate that heat flows from the element must equal the rate that heat enters, or else the temperature of the element will vary with time. In this chapter, we consider only the case of *steady-state* or *equilibrium* temperatures, so we can equate the rates of heat entering and leaving the element:

$$-kA\left(\frac{du}{dx}\right) = -kA\left[\frac{du}{dx} + \frac{d}{dx}\left(\frac{du}{dx}\right)dx\right].$$

When some common terms on each side of the equation are canceled, we get the very simple relation

$$kA\frac{d}{dx}\left(\frac{du}{dx}\right)dx = kA\frac{d^2u}{dx^2} = 0,$$

where we have written the second derivative in its usual form. For this particularly simple example, the equation for u as a function of x is the solution to

$$\frac{d^2u}{dx^2} = 0,$$

and this is obviously just

$$u = ax + b,$$

a linear relation. This means that the temperatures vary linearly from *TL* to *TR* as x goes from 0 to L.

The rod could also lose heat from the outer surface of the element. If this is Q (cal/(sec * cm^2)), the rate of heat flow in must equal the rate leaving the element by conduction along the rod plus the rate at which heat is lost from the surface. This means that:

$$-kA\left(\frac{du}{dx}\right) = -kA\left[\frac{du}{dx} + \frac{d}{dx}\left(\frac{du}{dx}\right)dx\right] + Qp\,dx,$$

where p is the perimeter at point x. (Q might also depend on the difference in temperature within the element and the temperature of the surroundings, but we will ignore that for now.)

If this equation is expanded and common terms are canceled, we get a somewhat more complicated equation whose solution is not obvious:

$$\frac{d^2u}{dx^2} = \frac{Qp}{(kA)}. \tag{6.24}$$

In Eq. (6.24), Q can be a function of x.

The situation may not be quite as simple as this. The cross section could vary along the rod, or k could be a function of x (some kind of composite of materials, possibly). Suppose first that only the cross section varies with x. We will have, then, for the rate of heat leaving the element

$$-k[A + A'\,dx]\left[\frac{du}{dx} + u''\,dx\right],$$

where we have used a prime notation for derivatives with respect to x. Equating the rates in and out as before and canceling common terms results in

$$kAu''\,dx + kA'u'\,dx + kA'u''\,dx^2 = Qp\,dx.$$

We can simplify this further by dropping the term with dx^2 because it goes to zero faster than the terms in dx. After also dividing out dx, this results in a second-order differential equation similar in form to some we have discussed in Section 6.5:

$$kAu'' + kA'u' = Qp. \tag{6.25}$$

The equation can be generalized even more if k also varies along the rod. We leave to the reader as an exercise to show that this results in

$$kAu'' + (kA' + k'A)u' = Qp. \tag{6.26}$$

If the rate of heat loss from the outer surface is proportional to the difference in temperatures between that within the element and the surroundings (u_s), (and this is a common situation), we must substitute for Q:

$$Q = q(u - u_s),$$

giving

$$kAu'' + (kA' + k'A)u' - q * pu = -q * pu_s. \tag{6.27}$$

This chapter will discuss two ways to solve equations like Eqs. (6.24) to (6.27).

Heat flow has been used in this section as the physical situation that is modeled, but equations of the same form apply to diffusion, certain types of fluid flow, torsion in objects subject to twisting, distribution of voltage, in fact, to any problem where the potential is proportional to the gradient.

The Shooting Method

We can rewrite Eq. (6.27) as

$$A\frac{d^2u}{dx^2} + B\frac{du}{dx} + Cu = D, \tag{6.28}$$

where the coefficients, A, B, C, and D are functions of x. (Actually, they could also be functions of both x and u, but that makes the problem more difficult to solve. In a temperature-distribution problem, such nonlinearity can be caused if the thermal conductivity, k, is considered to vary with the temperature, u. That is actually true for almost all materials but, as the variation is usually small, it is often neglected and an average value is used.)

To solve Eq. (6.28), we must know two conditions on u or its derivative. If both u and u' are specified at some starting value for x, the problem is an *initial-value problem.* In this section, we consider Eq. (6.28) to have two values of u to be given but these are at two different values for x—this makes it a *boundary-value problem.* In this section, we discuss how the same procedures that apply to an initial-value problem can be adapted.

The strategy is simple: Suppose we know u at $x = a$ (the beginning of a region of interest) and u at $x = b$ (the end of the region). We wish we knew u' at $x = a$; that would make it an initial-value problem. So, why not assume a value for this? Some general knowledge of the situation may indicate a reasonable guess. Or we could blindly select some value. The test of the accuracy of the guess is to see if we get the specified $u(b)$ by solving the problem over the interval $x = a$ to $x = b$. If the initial slope that we assumed is too large, we will often find that the computed value for $u(b)$ is too large. So, we try again with a smaller initial slope. If the new value for $u(b)$ is too small, we have bracketed the correct initial slope. This method is called the *shooting method* because of its resemblance to the problem faced by an artillery officer who is trying to hit a distant target. The right elevation of the gun can be found if two shots are made of which one is short of the target and the other is beyond. That means that an intermediate elevation will come closer.

EXAMPLE 6.2 Solve

$$u'' - \left(1 - \frac{x}{5}\right)u = x, \qquad u(1) = 2, \qquad u(3) = -1.$$

(This is an instance of Eq. (6.28) with $A = 1$, $B = 0$, $C = -(1 - x/5)$, and $D = x$.) Assume that $u'(1) = -1.5$ (which might be a reasonable guess, because u declines

Table 6.15

	Assume $u'(1) = -1.5$		Assume $u'(1) = -3.0$		Assume $u'(1) = -3.4950$	
x	u	u'	u	u'	u	u'
1.00	2.0000	−1.5000	2.0000	−3.0000	2.0000	−3.4950
1.20	1.7614	−0.9886	1.4598	−2.5118	1.3503	−3.0145
1.40	1.6043	−0.4814	0.9921	−2.0719	0.7900	−2.5967
1.60	1.5597	0.0389	0.6192	−1.6598	0.3099	−2.2204
1.80	1.6218	0.5876	0.3275	−1.2580	−0.0997	−1.8671
2.00	1.7976	1.1783	0.1163	−0.8512	−0.4385	−1.5209
2.20	2.0967	1.8227	−0.0118	−0.4259	−0.7076	−1.1679
2.40	2.5309	2.5310	−0.0520	0.0299	−0.9043	−0.7955
2.60	3.1139	3.3116	0.0029	0.5266	−1.0237	−0.3925
2.80	3.8608	4.1706	0.1620	1.0732	−1.0586	0.0511
3.00	4.7876	5.1119	0.4360	1.6773	−1.0000	0.5439

between $x = 1$ and $x = 3$; this number is the average slope over the interval). If we use a program that implements the Runge–Kutta–Fehlberg method, we get the values shown in the first part of Table 6.15.

Because the value for $u(3)$ is 4.7876 rather than the desired -1, we try again with a different initial slope, say $u'(1) = -3.0$, and get the middle part of Table 6.15. The resulting value for $u(3)$ is still too high: 0.4360 rather than -1. We could guess at a third trial for $u'(1)$, but let us interpolate linearly between the first two trials.* Doing so suggests a value for $u'(1)$ of -3.4950. Lo and behold, we get the correct answer for $u(3)$! These results are shown in the third part of Table 6.15.

It was not just by chance that we got the correct solution by interpolating from the first two trials. The problem is *linear* and for linear equations this will always be true. Except for truncation and round-off errors, the exact solution to a linear boundary-value problem by the shooting method is a linear combination of two trial solutions:

Suppose that $x_1(t)$ and $x_2(t)$ are two trial solutions of a boundary-value problem

$$x'' + Fx' + Gx = H, \qquad x(t_0) = A, \qquad x(t_f) = B$$

(where F, G, and H are functions of t only) and both trial solutions begin at the correct value of $x(t_0)$.

We then state that

$$y(t) = \frac{c_1 x_1 + c_2 x_2}{c_1 + c_2}$$

* If G = guess, and R = result: DR = desired result: G3 = G2 + (DR − R2)(G1 − G2)/(R1 − R2)

is also a solution. We show that this is true, because, since x_1 and x_2 are solutions, it follows that

$$x_1'' + Fx_1' + Gx_1 = H, \qquad \text{and} \qquad x_2'' + Fx_2' + Gx_2 = H.$$

If we substitute y into the original equations, with

$$y' = \frac{c_1x_1' + c_2x_2'}{c_1 + c_2}, \qquad \text{and} \qquad y'' = \frac{c_1x_1'' + c_2x_2''}{c_1 + c_2},$$

we get

$$\frac{c_1x_1'' + c_2x_2''}{c_1 + c_2} + \frac{c_1x_1' + c_2x_2'}{c_1 + c_2}F + \frac{c_1x_1 + c_2x_2}{c_1 + c_2}G = \frac{c_1x_1'' + c_1Fx_1' + c_1Gx_1 + c_2x_2'' + c_2Fx_2' + c_2Gx_2}{c_1 + c_2}$$

$$= \frac{c_1H}{c_1 + c_2} + \frac{c_2H}{c_1 + c_2} = H,$$

which shows that y is also a solution that begins at the correct value for $x(t_0)$. The implication of this is that, if c_1 and c_2 are chosen so that $y(t_f) = x(t_f) = B$, $y(t)$ is the correct solution to the boundary-value problem.

It must also be true that $y'(t_0)$ is the correct initial slope and that one can interpolate between every pair of computed values to get correct values for $y(x)$ at intermediate points.

This next example shows that we cannot get the correct solution so readily when the problem is *nonlinear.*

EXAMPLE 6.3 Solve

$$u'' - \left(1 - \frac{x}{5}\right)uu' = x, \qquad u(1) = 2, \qquad u(3) = -1.$$

This resembles Example 6.2 but observe that the coefficient of u' involves u, the dependent variable. This problem is nonlinear and we shall see that it is not as easy to solve. If we again use the Runge–Kutta–Fehlberg method, we get the results summarized in Table 6.16. Here the third trial, which used the interpolated value from the first two trials,

Table 6.16

Assumed value for $u'(1)$	Calculated value for $u(3)$
−1.5	−0.0282
−3.0	−2.0705
−2.2137*	−1.2719
−1.9460*	−0.8932
−2.0215*	−1.0080
−2.0162*	−1.0002
−2.0161*	−1.0000

* Interpolated from two previous values

Table 6.17

x	u	u'
1.0000	2.0000	−2.0161
1.2000	1.5552	−2.4130
1.4000	1.0459	−2.6438
1.6000	0.5318	−2.6352
1.8000	0.0082	−2.3832
2.0000	−0.4272	−1.9472
2.2000	−0.7640	−1.4110
2.4000	−0.9896	−0.8441
2.6000	−1.1022	−0.2848
2.8000	−1.1047	0.2569
3.0000	−1.0000	0.7909

does not give the correct solution. A nonlinear problem requires a kind of search operation. We could interpolate with a quadratic from the results of three trials, an adaptation of Muller's method. Table 6.17 gives the computed values for $u(x)$ between $x = 1$ and $x = 3$ with the final (good) estimate of the initial slope.

The shooting method is often quite laborious, especially with problems of fourth or higher order. With these, the necessity of assuming two or more conditions at the starting point (and matching with the same number of conditions at the end) is slow and tedious.

There are times when it is better to compute "backwards" from $x = b$ to $x = a$. For example, if $u(b)$ and $u'(a)$ are the known boundary values, the technique just described works best if we compute from $x = b$ to $x = a$. Another time that computing backwards would be preferred is in a fourth-order problem where three conditions are given at $x = b$ and only one at $x = a$.

Maple's `dsolve` command works with boundary-value problems. Here is how it can solve Example 6.3.

```
>de2 := diff(u(x),x$2)-(1-x/5)*u(x)*diff(u(x)x) = x:
>F := dsolve({de2, u(1) = 2, u(3) = -1}, u(x), numeric);
     F := proc(bvp_x . . . end proc
>F(1); F(2); F(3);
       x = 1., u(x) = 2.,∂/∂x u(x) = -2.01607429521390014
  x = 2., u(x) = -.427176163177449108, ∂/∂x u(x) =
    -1.94723020165843686
  x = 3., u(x) = -1.00000000000000022, ∂/∂x u(x) =
    .790910254537530277
>F(1.4); F(2.6);
  x = 1.4, u(x) = 1.04594603838311962, ∂/∂x u(x) =
     -2.64376847138324100
  x = 2.6, u(x) = -1.10221333664797760, ∂/∂x u(x) =
     -.284818239545453100
```

In this, we first defined the second-order equation, then used the dsolve command to get the solution, F, (a "procedure" that is not spelled out). When we asked for values of the solution at $x = 1, 2, 3, 1.4$, and 2.6, Maple displayed results that match to Table 6.17 but with many more digits of precision.

Solution Through a Set of Equations

There is another way to solve boundary-value problems like Example 6.2. We have seen in Chapter 5 that derivatives can be approximated by finite-difference quotients. If we replace the derivatives in a differential equation by such expressions, we convert it into a difference equation whose solution is an approximation to the solution of the differential equation. This method is sometimes preferred over the shooting method, but it really can be used only with linear equations. (If the differential equation is nonlinear, this technique leads to a set of nonlinear equations that are more difficult to solve. Solving such a set of nonlinear equations is best done by iteration, starting with some initial approximation to the solution vector.)

EXAMPLE 6.4 Solve the boundary-value problem of Example 6.2 but use a set of equations obtained by replacing the derivative with a central difference approximation. Divide the region into four equal subintervals and solve the equations, then divide into ten subintervals. Compare both of these solutions to the results of Example 6.2.

When the interval from $x = 1$ to $x = 3$ is subdivided into four subintervals, there are interior points (these are usually called *nodes*) at $x = 1.5$, 2.0, and 2.5. Label the nodes as x_1, x_2, and x_3. The endpoints are x_0 and x_4. We write the difference equation at the three interior nodes. The equation, $u'' - (1 - x/5)u = x$, $u(1) = 2$, $u(3) = -1$, becomes

$$\text{At } x_1: \quad \frac{(u_0 - 2u_1 + u_2)}{h^2} - \left(1 - \frac{x_1}{5}\right)u_1 = x_1,$$

$$\text{At } x_2: \quad \frac{(u_1 - 2u_2 + u_3)}{h^2} - \left(1 - \frac{x_2}{5}\right)u_2 = x_2,$$

$$\text{At } x_3: \quad \frac{(u_2 - 2u_3 + u_4)}{h^2} - \left(1 - \frac{x_3}{5}\right)u_3 = x_3.$$

These equations are all of the form:

$$\text{At } x_i: \quad \frac{(u_{i-1} - 2u_i + u_{i+1})}{h^2} - \left(1 - \frac{x_i}{5}\right)u_i = x_i,$$

which can be rearranged into:

$$\text{At } x_i: \quad u_{i-1} - \left[2 + h^2\left(1 - \frac{x_i}{5}\right)\right]u_i + u_{i+1} = h^2 x_i.$$

Substitute $h = 0.5$, substitute the x-values at the nodes, and substitute the u-values at the endpoints and arrange in matrix form, which gives

$$\begin{bmatrix} -2.175 & 1 & 0 \\ 1 & -2.150 & 1 \\ 0 & 1 & -2.125 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} -1.625 \\ 0.5 \\ 1.625 \end{bmatrix}.$$

Observe that the system is tridiagonal and that this will always be true even when there are many more nodes, because any derivative of u involves only points to the left, to the right, and the central point.

When this system is solved, we get

$$x_1 = 0.552, \qquad x_2 = -0.424 \qquad \text{and} \qquad x_3 = -0.964.$$

If we solve the problem again but with ten subintervals ($h = 0.2$), we must solve a system of nine equations, because there are nine interior nodes where the value of u is unknown. The answers, together with the results from the shooting method for comparison, are

x	Values from the finite-difference method	Values from the shooting method
1.2	1.351	1.350
1.4	0.792	0.790
1.6	0.311	0.309
1.8	−0.097	−0.100
2.0	−0.436	−0.438
2.2	−0.705	−0.708
2.4	−0.903	−0.904
2.6	−1.022	−1.024
2.8	−1.058	−1.059

There is quite close agreement. It is difficult to say from this which method is more accurate because both are subject to error. We can compare the methods and determine how making the number of subintervals greater increases the accuracy by examining the results for a problem with a known analytical answer.

EXAMPLE 6.5 Compare the accuracy of the finite-difference method with the shooting method on this second-order boundary-value problem:

$$u'' = u, \qquad u(1) = 1.17520, \qquad u(3) = 10.01787,$$

whose analytical solution is $u = \sinh(x)$.

When the problem is solved by finite-difference approximations to the derivatives, the typical equation is

$$u_{i-1} - (2 + h^2)u_i + u_{i+1} = 0.$$

Solving with $h = 1$, $h = 0.5$, and $h = 0.25$, we get the values in Table 6.18. If we solve this with the shooting method (employing Runge–Kutta–Fehlberg), we get Table 6.19.

Table 6.18 Solutions with the finite-difference method

	u-values with		
x	**2 subintervals**	**4 subintervals**	**8 subintervals**
1.25			1.60432
1.50		2.14670	2.13372
1.75			2.79647
2.00	3.73102	3.65488	3.63400
2.25			4.69866
2.50		7.07678	7.05698
2.75			7.79387
error at $x = 2.00$	0.10416	0.02802	0.00714

In both tables, the errors at $x = 2.0$ are shown. This is nearly the maximum error of any of the results.

When the results from the two methods are compared, it is clear that (1) the shooting method is much more accurate at the same number of subintervals, its errors being from 80 to over 500 times smaller; and (2) the errors for the finite-difference method decrease about four times when the number of subintervals is doubled, which is as expected.

The reader should make a similar comparison for other equations.

Derivative Boundary Conditions

The conditions at the boundary often involve the derivative of the dependent variable in addition to its value. A hot object loses heat to its surroundings proportional to the

Table 6.19 Solutions with the shooting method

	u-values with		
x	**2 subintervals**	**4 subintervals**	**8 subintervals**
1.25			1.60192
1.50		2.12931	2.12928
1.75			2.79042
2.00	3.62814	3.62692	3.62686
2.25			4.69117
2.50		7.05025	7.05020
2.75			7.78935
error at $x = 2.00$	0.00128	0.00006	0.00000

Figure 6.5

difference between the temperature at the surface of the object and the temperature of the surroundings. The proportionality constant is called the *heat-transfer coefficient* and is frequently represented by the symbol h. (This can cause confusion because we use h for the size of a subinterval. To avoid this confusion, we shall use a capital letter, H, for the heat-transfer coefficient.) The units of H are cal/sec/cm^2/°C (of temperature difference). In this section we consider a rod that loses heat to the surroundings from one or both ends. Of course, heat could be gained from the surroundings if the surroundings are hotter than the rod.

Names have been given to the various types of boundary conditions. If the value for u is specified at a boundary, it is called a *Dirichlet condition.* This is the type of problem that we have solved before. If the condition is the value of the derivative of u, it is a *Neumann condition.* When a boundary condition involves both u and its derivative, it is called a *mixed condition.*

We now develop the relations when heat is lost from the ends of a rod that conducts heat along the rod but is insulated around its perimeter so that no heat is lost from its lateral surface. First consider the right end of the rod and assume that heat is being lost to the surroundings (implying that the surface is hotter than the surroundings). Figure 6.5 will help to visualize this. At the right end of the rod ($x = x_R$), the temperature is u_R; the temperature of the surroundings is u_{SR}. Heat then is being lost from the rod to the surroundings at a rate [measured in (cal/sec)] of

$$HA(u_R - u_{SR}),$$

where A is the area of the end of the rod. This heat must be supplied by heat flowing from inside the rod to the surface, which is at the rate of

$$-kA\frac{du}{dx},$$

where the minus sign is required because heat flows from high to low temperature. Equating these two rates and solving for du/dx (the gradient) gives (the A's cancel):

$$\frac{du}{dx} = -\left(\frac{H}{k}\right)(u_R - u_{SR}), \qquad \text{at the right end.}$$

Now consider the left end of the rod, at $x = 0$, where $u = u_L$. Assume that the temperature of the surroundings here are at some other temperature, u_{SL}. Here, heat is flowing from right to left, so we have

$$\text{Heat leaving the rod: } -HA(u_L - u_{SL}).$$

For the rate at which heat flows from inside the rod we still have

$$-kA\frac{du}{dx},$$

and, after equating and solving for the gradient:

$$\frac{du}{dx} = \left(\frac{H}{k}\right)(u_L - u_{SL}), \qquad \text{at the left end.}$$

The fact that the signs in the equations for the gradients are not the same can be a source of confusion. Of course, if both ends lose heat to the surroundings, the equilibrium or steady-state temperatures of the rod will just be a linear relation between the two (possibly different) surrounding temperatures. In practical situations of heat distribution in a rod, only one end of the rod loses (or gains) heat to (from) the surroundings, the other end being held at some constant temperature.

A minor problem is presented in the cases under consideration. We need to give consideration to how to approximate the gradient at the end of the rod. One could use a forward difference approximation (at the right end, a backward difference at the left), but that seems inappropriate when central differences are used to approximate the derivatives within the rod. This conflict can be resolved if we imagine that the rod is fictitiously extended by one subinterval at the end of the rod that is losing heat. Doing so permits us to approximate the derivative with a central difference. The "temperature" at this fictitious point is eliminated by using the equation for the gradient. The next example will clarify this.

EXAMPLE 6.6 An insulated rod is 20 cm long and is of uniform cross section. It has its right end held at 100° while its left end loses heat to the surroundings, which are at 20°. The rod has a thermal conductivity, k, of 0.52 cal/(sec * cm * °C), and the heat-transfer coefficient, H, is 0.073 cal/(sec/cm^2/°C). Solve for the steady-state temperatures using the finite-difference method with eight subintervals.

For this example, because the boundary condition at the left end involves both the u-value at the left end and the derivative there, this example has a mixed condition at the left end, whereas it has a Dirichlet condition at the right end.

The equation that applies is Eq. (6.24) with $Q = 0$, because no heat is added at points along the rod:

$$\frac{d^2u}{dx^2} = 0.$$

The typical equation is

$$u_{i-1} - 2u_i + u_{i+1} = 0,$$

and this applies at each node. At the left end we imagine a fictitious point at x_{-1}, and this allows us to write the equation for that node. At the left endpoint, at $x = x_0$, we write an equation for the gradient:

$$\frac{du}{dx} = \left(\frac{H}{k}\right)(u_L - u_{SL}),$$

or,

$$\frac{(u_1 - u_{-1})}{2h} = \frac{(u_1 - u_{-1})}{(2 * 2.5)}$$
$$= \left(\frac{0.073}{0.52}\right) * (u_0 - 20),$$

which we use to eliminate u_{-1}:

$$u_{-1} = u_1 - (2 * 2.5) * \left[\left(\frac{0.073}{0.52}\right)(u_0 - 20)\right]$$
$$= u_1 - 0.70192u_0 + 14.0385.$$

We will use this last for the equation written at x_0, to give, at that point:

$$u_{-1} - 2u_0 + u_1 = (u_1 - 0.70192u_0 + 14.0385) - 2u_0 + u_1 = 0,$$

or,

$$-2.70192u_0 + 2u_1 = -14.0385,$$

which is the first equation of the set. Here is the augmented matrix for the problem:

$$\begin{bmatrix} -2.70192 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & -14.0385 \\ 1 & -2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -2 & -100 \end{bmatrix}$$

for which the solution is

i:	0	1	2	3	4	5	6	7	(8)
u_i:	41.0103	48.3840	55.7577	63.1314	70.5051	77.8789	85.2526	92.6263	(100)

Observe that the gradient all along the rod is a constant (2.94948°C/cm).

Here is another example that illustrates an important point about derivative boundary conditions.

EXAMPLE 6.7 Solve $u'' = u$, $u'(1) = 1.17520$, $u'(3) = 10.01787$, with the finite-difference method.

This example is identical to that of Example 6.5, except that the boundary conditions are the derivatives of u rather than the values of u. (It has Neumann conditions at both

ends.) For this problem, the known solution is $u = \cosh(x) + C$, and the boundary values are values of sinh(1) and sinh(3).

Because the values of u are not given at either end of the interval, we must add fictitious points at both ends; call these u_{LF} and u_{RF}. With four subintervals, ($h = 2/4 = 0.5$), we can write five equations (at each of the three interior nodes plus the two endpoints where u is unknown). We label the nodes from x_0 (at the left end) to x_4 (at the right end). Each equation is of the form:

$$u_{i-1} - 2u_i + u_{i+1} = h^2 u_i, \quad i = 0, 1, 2, 3, 4, \quad h^2 = 0.25,$$

where u_{-1} and u_5 are the fictitious points u_{LF} and u_{RF}.

Doing so gives this augmented matrix:

$$\begin{bmatrix} -2.25 & 1 & 0 & 0 & 0 & -u_{LF} \\ 1 & -2.25 & 1 & 0 & 0 & 0 \\ 0 & 1 & -2.25 & 1 & 0 & 0 \\ 0 & 0 & 1 & -2.25 & 1 & 0 \\ 0 & 0 & 0 & 1 & -2.25 & -u_{RF} \end{bmatrix}$$

There are two more unknowns in this than equations: the unknown fictitious points. However, these can be eliminated by using the derivative conditions at the ends. As before, we use central difference approximation to the derivative:

$$u'(1) = 1.17520 = \frac{(u_1 - u_{LF})}{2h},$$

$$u'(3) = 10.01787 = \frac{(u_{RF} - u_3)}{2h}, \qquad (h = 0.5),$$

which we solve for the fictitious points in terms of nodal points:

$$u_{LF} = u_1 - 1.17520, \qquad u_{RF} = 10.01787 + u_3.$$

Substituting these relations for the fictitious points changes the first and last equations to

$$-2.25u_0 + 2u_1 = 1.17520,$$
$$2u_3 - 2.25u_4 = -10.01787.$$

When the five equations are solved, we get these answers:

x	Answers	$\cosh(x)$	Error
1.0	1.55219	1.54308	−0.00911
1.5	2.33382	2.35241	0.01859
2.0	3.69870	3.76220	0.06350
2.5	5.98870	6.13229	0.14359
3.0	9.77568	10.06770	0.29202

We observe that the accuracy is much poorer than it was in Example 6.5. Take note of the fact that the numerical solution is not identical to the analytical solution; the arbitrary constant is missing (or, we may say, is equal to zero).

Using the Shooting Method

We can solve boundary-value problems where the derivative is involved at one or both end conditions by "shooting." In fact, as this method computes both the dependent variable and its derivative, this is quite natural. Here is how Example 6.7 can be solved by the shooting method.

EXAMPLE 6.8 Solve $u'' = u$, $u'(1) = 1.17520$, $u'(3) = 10.01787$ by the shooting method.

We can begin at either end, but it seems more natural to begin from $x = 1$. To begin the solution, we must guess at a value for $u(1)$—not for the derivative as we have been doing. From this point, we compute values for u and u' by, say, RKF. If the value of $u'(3)$ is not 10.01787, we try again with a guess for $u(1)$. This will probably not give the correct value for $u'(3)$, but, because the problem is linear, we can interpolate to find the proper value to use for $u(1)$. Here are the answers when four subintervals are used:

x	$u(x)$	$u'(x)$	$\cosh(x)$
1.0	1.54319	1.17520	1.54308
1.5	2.35250	2.12932	2.35241
2.0	3.76228	3.62692	3.76220
2.5	7.13236	7.05027	6.13229
3.0	10.06767	10.01790	10.06770

The results are surprisingly accurate even though the subdivision was coarse; the largest error in the $u(x)$ values is 0.00011 at $x = 1$ and the errors are less as x increases. For this example, the shooting method is much more accurate than using finite-difference approximations to the derivative.

Here is an example that has a mixed end condition.

EXAMPLE 6.9 Solve Example 6.6 by the shooting method. We restate the problem:

An insulated rod is 20 cm long and is of uniform cross section. It has its right end held at 100° while its left end loses heat to the surroundings, which are at 20°. The rod has a thermal conductivity, k, of 0.52 cal/(sec * cm * °C), and the heat-transfer coefficient, H, is 0.073 cal/(sec * cm^2 * °C). Use the shooting method with eight subintervals.

The procedure here is similar to that used in Example 6.8 but it is necessary to begin at the right end and solve "backwards." (That is no problem; we just use a negative value for Δx.) Beginning at $x = 0$ would be very difficult because we would have to guess at both $u(0)$ and $u'(0)$.

Finding the correct value for u' at $x = 20$ is not as easy as in the previous example because we must fit to a combination of $u(0)$ and $u'(0)$. Here are the results after finding the correct value for $u'(20)$ by a trial and error technique.

i:	0	1	2	3	4	5	6	7	(8)
u_i:	41.005	48.379	55.754	63.128	70.502	77.877	85.251	92.626	(100)

(The gradient here is 2.94975 throughout.) These values match those of Example 6.6 very closely.

We note that Maple can solve a boundary-value problem with an end condition that involves the derivative.

6.8 Characteristic-Value Problems

Problems in the fields of elasticity and vibration (including applications of the wave equation of modern physics) fall into a special class of boundary-value problems known as *characteristic-value problems*. Some problems of statistics also fall into this class. We discuss only the most elementary forms of characteristic-value problems.

Consider the homogeneous* second-order equation with homogeneous boundary conditions:

$$\frac{d^2u}{dx^2} + k^2u = 0, \qquad u(0) = 0, \qquad u(1) = 0, \tag{6.29}$$

where k^2 is a parameter. (Using k^2 guarantees that the parameter is a positive number.) We first solve this equation nonnumerically to show that there is a solution only for certain particular or "characteristic" values of the parameter. These characteristic values are more often called the *eigenvalues* from the German word. The general solution is

$$u = a \sin(kx) + b \cos(kx),$$

which can easily be verified by substituting into the differential equation. The solution contains the two arbitrary constants a and b because the equation is of second order. The constants a and b are to be determined to make the general solution agree with the boundary conditions.

At $x = 0$, $u = 0 = a \sin(0) + b \cos(0) = b$. Then b must be zero. At $x = 1$, $u = 0 = a \sin(k)$; we may have either $a = 0$ or $\sin(k) = 0$ to satisfy the end condition. However, if $a = 0$, y is everywhere zero—this is called the *trivial solution,* and is usually of no interest. To get a useful solution, we must choose $\sin(k) = 0$, which is true only for certain "characteristic" values:

$$k = \pm n\pi, \qquad n = 1, 2, 3, \ldots .$$

* Homogeneous here means that all terms in the equation are functions of u or its derivatives.

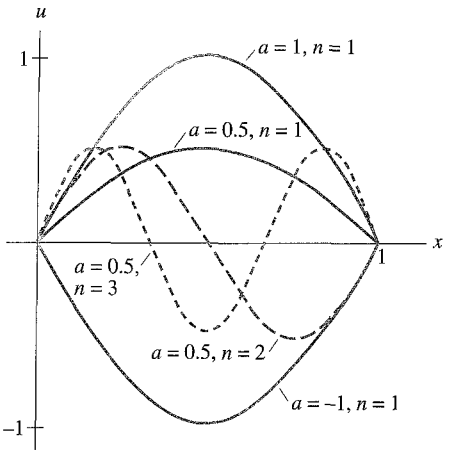

Figure 6.6

These are the eigenvalues for the equation, and the solution to the problem is

$$u = a \sin(n\pi x), \qquad n = 1, 2, 3, \ldots . \tag{6.30}$$

The constant a can have any value, so these solutions are determined only to within a multiplicative constant. Figure 6.6 sketches several of the solutions to Eq. (6.30).

These eigenvalues are the most important information for a characteristic-value problem. In a vibration problem, these give the natural frequencies of the system, which are important because, if the system is subjected to external loads applied at or very near to these frequencies, resonance causes an amplification of the motion and failure is likely.

Corresponding to each eigenvalue is an eigenfunction, $u(x)$, which determines the possible shapes of the elastic curve when the system is at equilibrium. Figure 6.6 shows such eigenfunctions. Often the smallest eigenvalue is of particular interest; at other times, it is the one of largest magnitude.

We can solve Eq. (6.29) numerically, and that is what we concentrate on in this section. We will replace the derivatives in the differential equation with finite-difference approximations, so that we replace the differential equation with difference equations written at all nodes where the value of u is unknown (which are all the nodes of a one-dimensional system except for the endpoints).

EXAMPLE 6.10 Solve Eq. (6.29) with five subintervals. We restate the problem:

$$\frac{d^2u}{dx^2} + k^2u = 0, \qquad u(0) = 0, \qquad u(1) = 0.$$

The typical equation is

$$\frac{(u_{i-1} - 2u_i + u_{i+1})}{h^2} + k^2u_i = 0.$$

With five subintervals, $h = 0.2$, and there are four equations because there are four interior nodes. In matrix form these are

$$\begin{bmatrix} 2 - 0.04k^2 & -1 & 0 & 0 \\ -1 & 2 - 0.04k^2 & -1 & 0 \\ 0 & -1 & 2 - 0.04k^2 & -1 \\ 0 & 0 & -1 & 2 - 0.04k^2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{6.31}$$

where we have multiplied by -1 for convenience. Observe that this can be written as the matrix equation $(A - \lambda I)u = 0$, where I is the identity matrix and the A matrix is

$$\begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix},$$

and $\lambda = 0.04k^2$.

The approximate solution to the characteristic-value problem, Eq. (6.29) is found by solving the system of Eq. (6.31). However, this system is an example of a homogeneous system (the right-hand sides are all equal to zero), and it has a nontrivial solution only if the determinant of the coefficient matrix is zero. Hence, we set

$$\det(A - \lambda I) = 0.$$

Expanding the determinant will give an eighth-degree polynomial in k. (This is *not* the preferred way!) Doing so and getting the zeros of that polynomial gives these values for k:

$$k = \pm 3.09, \qquad k = \pm 5.88, \qquad k = \pm 8.09, \qquad k = \pm 9.51.$$

The analytical values for k are

$$k = \pm 3.14\ (\pm\pi), \qquad k = \pm 7.28\ (\pm 2\pi),$$
$$k = \pm 9.42\ (\pm 3\pi), \qquad k = \pm 12.57\ (\pm 4\pi),$$

and we see that the estimates for k are not very good and get progressively worse. We would need a much smaller subdivision of the interval to get good values. There are other problems with this technique: Expanding the determinant of a matrix of large size is computationally expensive, and solving for the roots of a polynomial of high degree is subject to large round-off errors. The system is very ill-conditioned.*

We normally find the eigenvalues for a characteristic-value problem from $(A - \lambda I)u = 0$ in other ways that are not subject to the same difficulties. We describe these now. For clarity we use small matrices.

The Power Method

The *power method* is an iterative technique. The basis for this is presented below. We illustrate the method through an example.

* One authority says never to use the characteristic polynomial for a matrix larger than 5×5.

EXAMPLE 6.11 Find the eigenvalues (and the eigenvectors) of matrix A:

$$A = \begin{bmatrix} 3 & -1 & 0 \\ -2 & 4 & -3 \\ 0 & -1 & 1 \end{bmatrix}$$

(The eigenvalues of A are 5.47735, 2.44807, and 0.074577, which are found, perhaps, by expanding the determinant of $A - \lambda I$. The eigenvectors are found by solving the equations $Au = \lambda u$ for each value of λ. After normalizing, these vectors are

$$u_1 = [-0.40365, 1, -0.22335],$$
$$u_2 = [1, 0.55193, -0.38115],$$
$$u_3 = [0.31633, 0.92542, 1],$$

where the normalization has been to set the largest component equal to unity.)*

We will find that both the eigenvalues and the eigenvectors are produced by the power method. We begin this by choosing a three-component vector more or less arbitrarily. (There are some choices that don't work but usually the column vector $u = [1, 1, 1]$ is a good starting vector.) We always use a vector with as many components as rows or columns of A.

We repeat these steps:

1. Multiply $A * u$.
2. Normalize the resulting vector by dividing each component by the largest in magnitude.
3. Repeat steps 1 and 2 until the change in the normalizing factor is negligible. At that time, the normalization factor is an eigenvalue and the final vector is an eigenvector.

Step 1, with $u = [1, 1, 1]$:

$$A * u \qquad \text{gives} \qquad [2, -1, 0].$$

Step 2:

Normalizing gives $2 * [1, -.5, 0]$, and u now is $[1, -.5, 0]$.

Repeating, we get

$A * u = [3.5, -4, .5]$,
normalized: $-4 * [-.875, 1, -.125]$;
$A * u = [-3.625, 6.125, -1.125]$,
normalized: $6.125 * [-.5918, 1, -.1837]$;
$A * u = [-2.7755, 5.7347, -1.1837]$,
normalized: $5.7347 * [-.4840, 1, -.2064]$;

After 14 iterations, we get

---

* It is more common to set some norm equal to 1.

$$A * u = [-2.21113, 5.47743, -1.22333],$$
$$\text{normalized: } 5.47743 * [-.40368, 1, -.22334].$$

The fourteenth iteration shows a negligible change in the normalizing factor: We have approximated the largest eigenvalue and the corresponding eigenvector. (Twenty iterations will give even better values.) Although not very rapid, the method is extremely simple and easy to program. Any of the computer algebra systems can do this for us.

The Inverse Power Method

The previous example showed how the power method gets the eigenvalue of largest magnitude. What if we want the one of smallest magnitude? All we need to do to get this is to work with the inverse of A. For the matrix A of Example 6.11, its inverse is

$$\begin{bmatrix} 1 & 1 & 3 \\ 2 & 3 & 9 \\ 2 & 3 & 10 \end{bmatrix}.$$

Applying the power method to this matrix gives a value for the normalizing factor of 13.4090 and a vector of [.3163, .9254, 1]. For the original matrix A, the eigenvalue is the reciprocal, 0.07457. The eigenvector that corresponds is the same; no change is needed.

Shifting with the Power Method

As we have seen, the power method may not converge very fast. We can accelerate the convergence as well as get eigenvalues of magnitude intermediate between the largest and smallest by *shifting*. Suppose we wish to determine the eigenvalue that is nearly equal to some number s. If s is subtracted from each of the diagonal elements of A, the resulting matrix has eigenvalues the same as for A but with s subtracted from them. This means that there is an eigenvalue for the shifted matrix that is nearly zero. We now use the inverse power method on this shifted matrix, and the reciprocal of this very small eigenvalue is usually very much larger in magnitude than any other. As shown below, this causes the convergence to be rapid. Observe that if we have some knowledge of what the eigenvalues of A are, we can use this shifted power method to get the value of any of them.

How can we estimate the eigenvalues of a matrix? *Gerschgorin's theorem* can help here. This theorem is especially useful if the matrix has strong diagonal dominance. The first of Gerschgorin's theorems says that the eigenvalues lie in circles whose centers are at a_{ii} with a radius equal to the sum of the magnitudes of the other elements in row i. (Eigenvalues can have complex values, so the circles are in the complex plane.)

Gerschgorin's Theorem We will not give a proof of this theorem,* but only show that it applies in several examples.

* Proofs can be found in Ralston (1965) and in Burdern and Faires (2001).

If matrix A is diagonal, its eigenvalues are the diagonal elements:

$$\begin{matrix} 10 & 0 & 0 \\ 0 & 7 & 0 \\ 0 & 0 & 4 \end{matrix} \rightarrow \quad 4,\ 7,\ 10, \text{ which are in } 4 \pm 0,\ 7 \pm 0,\ 10 \pm 0.$$

If matrix A has small off-diagonal elements:

$$\begin{matrix} 10 & 0.1 & 0.1 \\ 0.1 & 7 & 0.1 \\ 0.1 & 0.1 & 4 \end{matrix} \rightarrow 3.9951,\ 6.9998,\ 10.0051, \text{ in } 4 \pm 0.2,\ 7 \pm 0.2,\ 10 \pm 0.2,$$

and there is a small change.

When the off-diagonals are larger:

$$\begin{matrix} 10 & 1 & 1 \\ 1 & 7 & 1 \\ 1 & 1 & 4 \end{matrix} \rightarrow 3.6224,\ 6.8329,\ 10.5446, \text{ in } 4 \pm 2,\ 7 \pm 2,\ 10 \pm 2,$$

there is a greater change.

If they are still larger:

$$\begin{matrix} 10 & 2 & 2 \\ 2 & 7 & 2 \\ 2 & 2 & 4 \end{matrix} \rightarrow 2.8606,\ 6.2151,\ 11.9243, \text{ in } 4 \pm 4,\ 7 \pm 4,\ 10 \pm 4,$$

there is a still greater change, but the theorem holds.

Even in this case, the theorem holds:

$$\begin{matrix} 10 & 4 & 4 \\ 4 & 7 & 4 \\ 4 & 4 & 4 \end{matrix} \rightarrow 1.0398,\ 4.4704,\ 15.4898, \text{ in } 4 \pm 8,\ 7 \pm 8,\ 10 \pm 8.$$

Whenever the matrix is diagonally dominant or nearly so, shifting by the value of a diagonal element will speed up convergence in the power method.

EXAMPLE 6.12 Given matrix A:

$$\begin{bmatrix} 4 & -1 & 1 \\ 1 & 1 & 1 \\ -2 & 0 & -6 \end{bmatrix}$$

find all of its eigenvalues using the shifted power method.

Gerschgorin's theorem says that there are eigenvalues within -6 ± 2, 1 ± 2, and 4 ± 2. We shift first by -6 and get an eigenvalue equal to -5.76851 (vector = $[-.11574, -.13065, 1]$) using the inverse power method in four iterations; the tolerance on change in the normalization factor was 0.0001. (Getting this largest-magnitude eigenvalue through

the regular power method required 23 iterations.) If we repeat but shift by one, the inverse power method gives 1.29915 as an eigenvalue (vector = [.41207, 1, −.11291]) in six iterations. (Using just the inverse power method to get this smallest of the eigenvalues required eight iterations.)

For this 3×3 matrix, we do not have to get the other eigenvalue; the sum of the eigenvalues equals the trace of the matrix. So, if we subtract $(-5.76851 + 1.29915)$ from -1 (the trace) we get the third eigenvalue, 3.46936. (It is always true that the sum of the eigenvalues equals the trace.) The eigenvalues satisfy Gerschgorin's theorem: -5.76851 is in -6 ± 2, 1.29915 is in 1 ± 2, 3.46936 is in 4 ± 2.

Getting the third eigenvalue from the trace does not give us its eigenvector; we can use the shifted inverse power method on the original matrix to find it.

Shifting by 4 in this example runs into a problem; a division by zero is attempted. We overcome this problem by distorting the shift amount slightly. Shifting by 3.9 and employing the inverse power method gives the eigenvalue: 3.46936, and the vector [1, .31936, −.21121] in six iterations. (If a division by zero occurs, it is advisable to distort the shift amount slightly.)

The Basis for the Power Method

The utility of the power method is that it finds the eigenvalue of largest magnitude and its corresponding eigenvector in a simple and straightforward manner. It has the disadvantage that convergence is slow if there is a second eigenvalue of nearly the same magnitude. The following discussion proves this and also shows why some starting vectors are unsuitable.

The method works because the eigenvectors are a set of *basis vectors*. A set of basis vectors is said to *span the space,* meaning that any n-component vector can be written as a unique linear combination of them. Let $v^{(0)}$ be any vector and $x_1, x_2, \ldots, x_n$, be eigenvectors. Then, for a starting vector, $v^{(0)}$,

$$v^{(0)} = c_1x_1 + c_2x_2 + \cdots + c_nx_n.$$

If we multiply $v^{(0)}$ by matrix A, because the x_i are eigenvectors with corresponding eigenvalues λ_i and remembering that $Ax_i = \lambda_i x$, we have,

$$\begin{aligned} v^{(1)} = Av^{(0)} &= c_1Ax_1 + c_2Ax_2 + \cdots + c_nA_nx_n \\ &= c_1\lambda_1x_1 + c_2\lambda_2x_2 + \cdots + c_n\lambda_nx_n, \end{aligned} \tag{6.32}$$

Upon repeated multiplication by A, after m such multiplies, we get,

$$v^{(m)} = A^mv^{(0)} = c_1\lambda_1^mx_1 + c_2\lambda_2^mx_2 + \cdots + c_n\lambda_n^mx_n.$$

Now, if one of the eigenvalues, call it λ_1, is larger than all the rest, it follows that all the coefficients in the last equation become negligibly small in comparison to λ_1^m as m gets large, so

$$A^mv^{(0)} \rightarrow c_1\lambda_1^mx_1,$$

which is some multiple of eigenvector x_1 with the normalization factor λ_1, provided only that $c_1 \neq 0$. This is the principle behind the power method. Observe that if another of the eigenvalues is exactly of the same magnitude as λ_1, there never will be convergence to a single value. Actually, in this case, the normalization values alternate between two numbers and the eigenvalues are the square root of the product of these values. If another eigenvalue is not equal to λ_1, but is near to it, convergence will be slow. Also, if the starting vector, $v^{(0)}$, is such that the coefficient c_1 in Eq. (6.32) equals zero, the method will not work. (This last will be true if the starting vector is "perpendicular" to the eigenvector that corresponds to λ_1—that is, the dot-product equals zero.) On the other hand, if the starting vector is almost "parallel" to the eigenvector of λ_1, all the other coefficients in Eq. (6.32) will be very small in comparison to c_1 and convergence will be very rapid.

The preceding discussion also shows why shifting and then using the inverse power method can often speed up convergence to the eigenvalue that is near the shift quantity. Here we create, in the shifted matrix, an eigenvalue that is nearly zero, so that using the inverse method makes the reciprocal of this small number much larger than any other eigenvalue.

The power method with its variations is fine for small matrices. However, if a matrix has two eigenvalues of equal magnitude, the method fails in that the successive normalization factors alternate between two numbers. The duplicated eigenvalue in this case is the square root of the product of the alternating normalization factors. If we want all the eigenvalues for a larger matrix, there is a better way.

The QR Method, Part 1—Similarity Transformations

If matrix A is diagonal or upper- or lower-triangular, its eigenvalues are just the elements on the diagonal. This can be proved by expanding the determinant of $(A - \lambda I)$. This suggests that, if we can transform A to upper-triangular, we have its eigenvalues! We have done such a transformation before: The Gaussian elimination method does it. Unfortunately, this transformation changes the eigenvalues!!

There are other transformations that do not change the eigenvalues. These are called *similarity transformations*. For any nonsingular matrix, M, the product $M * A * M^{-1} = B$, transforms A into B, and B has the same eigenvalues as A. The trick is to find matrix M such that A is transformed into a similar upper-triangular matrix from which we can read off the eigenvalues of A from the diagonal of B. The QR technique does this. We first change one of the subdiagonal elements of A to zero; we then continue to do this for all the elements below the diagonal until A has become upper-triangular. The process is slow; many iterations are required, but the procedure does work.

Suppose that A is 4×4. Here is a matrix, Q, also 4×4, that will create a zero in position a_{42}:

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c & 0 & s \\ 0 & 0 & 1 & 0 \\ 0 & -s & 0 & c \end{bmatrix},$$

where

$$d = \sqrt{(a_{22}^2 + a_{42}^2)},$$

$$c = \frac{a_{22}}{d},$$

$$s = \frac{a_{42}}{d}.$$

EXAMPLE 6.13 Given this matrix A, create a zero in position (4, 2) by multiplying by the proper Q matrix.

$$A = \begin{bmatrix} 7 & 8 & 6 & 6 \\ 1 & 6 & -1 & -2 \\ 1 & -2 & 5 & -2 \\ 3 & 4 & 3 & 4 \end{bmatrix}$$

We compute:

$$d = \sqrt{(6^2 + 4^2)} = 7.21110,$$

$$c = \frac{6}{d} = 0.83205,$$

$$s = \frac{4}{d} = 0.55470.$$

The Q matrix is

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & .83205 & 0 & .55470 \\ 0 & 0 & 1 & 0 \\ 0 & -.55470 & 0 & .83205 \end{bmatrix}.$$

When we multiply Q by A, we get for $Q * A$:

$$\begin{bmatrix} 7 & 8 & 6 & 6 \\ 2.49615 & 7.21110 & .83205 & .55470 \\ 1 & -2 & 5 & -2 \\ 1.94145 & 0 & 3.05085 & 4.43760 \end{bmatrix}$$

where the element in position (4, 2) is zero, as we wanted. However, we do not yet have a similarity transformation. (The trace has been changed, meaning that the eigenvalues are not the same as those of A.) To get the similarity transformation that is needed, we must now postmultiply by the inverse of Q. Getting the inverse (which is Q^{-1}) is easy in this case because for any Q as defined here, its inverse is just its transpose! (When this is true for a matrix, it is called a *rotation matrix*.) If we now multiply $Q * A * Q^{-1}$, we get

$$\begin{bmatrix} 7 & 9.98460 & 6 & 0.55470 \\ 2.49615 & 6.30769 & 0.83205 & -3.53846 \\ 1 & -2.77350 & 5 & -0.55470 \\ 1.94145 & 2.46154 & 3.05085 & 3.69231 \end{bmatrix},$$

for which the trace is the same as that of the original A and whose eigenvalues are the same. However, it seems that we have not really done what we desired; the element in position (4, 2) is zero no longer! There has been some improvement, though. Observe that the sum of the magnitudes of the off-diagonal elements in row 4 is smaller than in matrix A. This means that 3.69231 is closer to one of the eigenvalues (which will turn out to be 1) than the original value, 4. Also, the element in position (2, 2) (6.30769) is closer to another eigenvalue (which is equal to 7) than the original number, 6.

This suggests that we should continue doing such similarity transformations to reduce all below-diagonal elements to zero. It takes many iterations, but, after doing 111 of these, we get

$$\begin{bmatrix} 10 & 1.5811 & -11.0680 & -3.0000 \\ 0 & 7 & -1.0000 & 0.0000 \\ 0 & 0 & 4 & -3.1623 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

where the numbers have been rounded to four decimals. (All the below-diagonal elements have a value of 0.00001 or less.) We have found the eigenvalues of A; these are 10, 7, 4, and 1.

The QR Method, Part 2 – Making the Matrix Upper Hessenberg

The trouble with doing such similarity transformations repeatedly is poor efficiency. We can improve the method by first doing a *Householder transformation,* which is a similarity transformation that creates zeros in matrix A for all elements below the "subdiagonal." (This means all elements below the diagonal except for those immediately below the diagonal. We might call such a matrix "almost triangular.") The name for such a matrix is *upper Hessenberg*. The Householder transformation changes matrix A into upper Hessenberg. Once an $n \times n$ matrix has been converted to upper Hessenberg, there are only $n - 1$ elements to reduce, compared to $(n)(n - 1)/2$.

There is another technique that further speeds up the reduction of matrix A to upper-triangular. We can employ shifting (similar to that done in the power method). The easiest way to shift is to do it with the element in the last row and last column.

Here are the steps that we will use:

1. Convert to upper Hessenberg.
2. Shift by a_{nn}, then do similarity transformations for all columns from 1 to $n - 1$.
3. Repeat step 2 until all elements to the left of a_{nn} are essentially zero. An eigenvalue then appears in position a_{nn}.

4. Ignore the last row and column, and repeat steps 2 and 3 until all elements below the diagonal of the original matrix are essentially zero. The eigenvalues then appear on the diagonal.

How do we convert matrix A to upper Hessenberg without changing the eigenvalues? This is best explained through an example.

EXAMPLE 6.14 Convert the same matrix A (as in Example 6.13) to upper Hessenberg.

We recall that A is

$$\begin{bmatrix} 7 & 8 & 6 & 6 \\ 1 & 6 & -1 & -2 \\ 1 & -2 & 5 & -2 \\ 3 & 4 & 3 & 4 \end{bmatrix}.$$

We can create zeros in the first column and rows 3 and 4 by $B * A * B^{-1}$, where

$$B = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -b_3 & 1 & 0 \\ 0 & -b_4 & 0 & 1 \end{bmatrix}, \qquad B^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & b_3 & 1 & 0 \\ 0 & b_4 & 0 & 1 \end{bmatrix},$$

$$b_3 = a_{31}/a_{21} = 1/1 = 1,$$

$$b_4 = a_{41}/a_{21} = 3/1 = 3.$$

Observe that the B matrix is the identity matrix with the two zeros below the diagonal in column 2 replaced with $-b_3$ and $-b_4$, where these values are the elements of column 1 of matrix A that are to be made zero divided by the subdiagonal element in column 1. The inverse of this B matrix is B with the signs changed for the new elements in its column 2.

If we now perform the multiplications $B_1 * A * B_1^{-1}$, we get

$$\begin{bmatrix} 7 & 32 & 6 & 8 \\ 1 & -1 & -1 & -2 \\ 0 & -2 & 6 & 0 \\ 0 & 22 & 6 & 10 \end{bmatrix},$$

which has zeros below the subdiagonal of column 1 and the same eigenvalues as the original matrix A.

We continue this in column 2, where now

$$B_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -b_4 & 1 \end{bmatrix}, \qquad \text{with} \qquad b_4 = a_{42}/a_{32} = 22/-2 = -11.$$

Here, B_2^{-1} is the same as B_2 except that the sign of b_4 is changed. Now premultiplying the last matrix by B_2 and postmultiplying by B_2^{-1} gives the lower Hessenberg matrix:

$$B_2B_1AB_1^{-1}B_2^{-1} = \begin{bmatrix} 7 & 32 & -60 & 6 \\ 1 & -1 & 21 & -2 \\ 0 & -2 & 6 & 0 \\ 0 & 0 & -38 & 10 \end{bmatrix},$$

which is what was desired.

There is a potential problem with this reduction to the Hessenberg matrix. If the divisor used to create the B matrices is zero or very small, either a division by zero occurs or the round-off error is great. We can avoid these problems by interchanging both rows and columns to put the element of largest magnitude in the subdiagonal position. It is essential to do the interchanges for both rows and columns so that the diagonal elements remain the same.

The QR Method, Part 3—The Steps Combined

If we (1) convert matrix A to upper Hessenberg, and, (2) perform QR operations on this, the final matrix that results is

$$\begin{bmatrix} 10 & 9.8315 & 4.9054 & -3.2668 \\ 0 & 1 & 1.8256 & 2.7199 \\ 0 & 0 & 4 & -1.6958 \\ 0 & 0 & 0 & 7 \end{bmatrix},$$

in which the same eigenvalues appear on the diagonal as when QR operations were done on the original A matrix. However, only seven QR iterations were required after reduction to Hessenberg, compared to 111 if that step is omitted. The other elements are different because row and column interchanges were done in creating the last result.

MATLAB can find the eigenvalues and eigenvectors of a square matrix. Here is an example:

Find the eigenvalues of

$$\begin{bmatrix} 10 & 0 & 0 \\ 1 & -3 & -7 \\ 0 & 2 & 6 \end{bmatrix}.$$

Solution:

We define A in MATLAB:

```
A = [10  0  0; 1  -3  -7; 0  2  6]
A =
```

```
 10   0   0
  1  -3  -7
  0   2   6
```

and then do

```
e = eig(A)
e =
     4
    -1
    10
```

If we want both the eigenvalues and eigenvectors:

```
[V, D] = eig(A)
V =
         0         0   0.9977
   -0.7071    0.9615   0.0605
    0.7071   -0.2747   0.0302
D =
     4   0   0
     0  -1   0
     0   0  10
```

where the eigenvectors appear as the columns of V (they are scaled so each has a norm of one) and the eigenvalues are on the diagonal of matrix D. Observe that MATLAB gets all the eigenvectors at once.

Suppose we want to get the eigenvalues of A after its element in row 1, column 2 is changed to one. If that is what we want, we just enter:

```
A(1, 2) = 1;
eig(A)
ans =
    10.0606
    -1.1250
     4.0644
```

MATLAB uses a QR algorithm to get the eigenvalues after converting to Hessenberg form as described. We can also use the characteristic polynomial:

After defining the original matrix (A) in MATLAB, we do

```
EDU>> pp = poly(A)
pp =
    1.0000  -13.0000   25.0000   46.0000
```

which are the coefficients of the cubic

$$x^3 - 13x^2 + 25x + 46.$$

We get the roots by

```
EDU>> roots(pp)
ans =
   10.0606
    4.0644
   -1.1250
```

which is the same as before, as expected.

Exercises

Section 6.1

1. Use the Taylor series method to get solutions to

$$dy/dx = x + y - xy, \qquad y(0) = 1$$

at $x = 0.1$ and $x = 0.5$. Use terms through x^5.

▶ **2.** The solution to Exercise 1 at $x = 0.5$ is 1.59420. How many terms of a Taylor series must be used to match this?

3. Repeat Exercises 1 and 2 but for

$$y''(x) = x/y, \qquad y(0) = 1, \qquad y'(0) = 1.$$

The correct value for $y(0.5)$ is 1.51676.

4. A spring system has resistance to motion proportional to the square of the velocity, and its motion is described by

$$\frac{d^2x}{dt^2} + 0.1\left(\frac{dx}{dt}\right)^2 + 0.6x = 0.$$

If the spring is released from a point that is a unit distance above its equilibrium point, $x(0) = 1$, $x'(0) = 0$, use the Taylor-series method to write a series expression for the displacement as a function of time, including terms up to t^6.

Section 6.2

5. Repeat Exercise 1, but use the simple Euler method. How small must h be to match to the values of Exercise 1?

▶ **6.** Repeat Exercise 2, but use the simple Euler method. How small must h be?

7. Repeat Exercise 5, but now with the modified Euler method. Comparing to Exercise 5, how much less effort is required?

8. Find the solution to

$$\frac{dy}{dt} = y^2 + t^2, \quad y(1) = 0, \quad \text{at } t = 2,$$

by the modified Euler method, using $h = 0.1$. Repeat with $h = 0.05$. From the two results, estimate the accuracy of the second computation.

9. Solve $y' = \sin(x) + y$, $y(0) = 2$ by the modified Euler method to get $y(0.1)$ and $y(0.5)$. Use a value of h small enough to be sure that you have five digits correct.

▶ **10.** A sky diver jumps from a plane, and during the time before the parachute opens, the air resistance is proportional to the $\frac{3}{2}$ power of the diver's velocity. If it is known that the maximum rate of fall under these conditions is 80 mph, determine the diver's velocity during the first 2 sec of fall using the modified Euler method with $\Delta t = 0.2$. Neglect horizontal drift and assume an initial velocity of zero.

11. Repeat Exercise 8 but use the midpoint method. Are the results the same? If not, which is more accurate?

12. The midpoint method gives results identical to modified Euler for $dy/dx = -2x - xy$, $y(0) = -1$. But for some definitions of dy/dx, it is better; for other definitions, it is worse. What are the conditions on the derivative function that cause

a. The midpoint method to be better?
b. The midpoint method to be poorer?
c. The two methods to give identical results?
d. Give specific examples for parts (a) and (b).

▶ **13.** For some derivative functions, the simple Euler method will have errors that are always positive but for others, the errors will always be negative.

a. What property of the function will determine which kind of error will be experienced?
b. Provide examples for both types of derivative function.

c. When will the errors be positive at first, but then become negative? Give an example where the errors oscillate between positive and negative as the x-values increase.

14. Is the phenomenon of Exercise 13 true for the modified Euler method? If it is, repeat Exercise 13 for this method.

Section 6.3

15. What are the equations that will be used for a second-order Runge–Kutta method if $a = 1/3$, $b = 2/3$, $\alpha = 3/4$ and $\beta = 3/4$. The statement is made that "this is said to give a minimum bound to the error." Test the truth of this statement by comparing this method with modified Euler on the equations of Exercises 1 and 8. Also compare to the midpoint method.

16. What is the equivalent of Eq. (6.10) for a third-order RK method? What then is the equivalent of Eq. (6.12)? Give three different combinations of parameter values that can be employed.

17. Use one set of the parameter values you found in Exercise 16 to solve Exercise 9.

a. How much larger can h be than the value found in Exercise 9?

b. Repeat with the other sets of parameters. Which set is preferred?

18. Solve Exercise 1 with fourth-order Runge–Kutta method. How large can h be to get the correct value at $x = 1.0$, which is 2.19496?

19. Determine y at $x = 1$ for the following equation, using fourth-order Runge–Kutta method with $h = 0.2$. How accurate are the results?

$$dy/dx = 1/(x + y), \qquad y(0) = 2.$$

▶20. Using the conditions of Exercise 10, determine how long it takes for the jumper to reach 90% of his or her maximum velocity, by integrating the equation using the Runge–Kutta technique with $\Delta t = 0.5$ until the velocity exceeds this value, and then interpolating. Then use numerical integration on the velocity values to determine the distance the diver falls in attaining $0.9v_{\max}$.

21. It is not easy to know the accuracy with which the function has been determined by either the Euler methods or the Runge–Kutta method. A possible way to measure accuracy is to repeat the problem with a smaller step size, and compare results. If the two computations agree to n decimal places, one then assumes the values are correct to that many places. Repeat Exercise 20 with $\Delta t = 0.3$, which should give a global error about one-eighth as large, and by comparing results, determine the accuracy in Exercise 20. (Why do we expect to reduce the error eightfold by this change in Δt?)

22. Solve Exercises 1, 9, and 10 by the Runge–Kutta–Fehlberg method.

23. Using Runge–Kutta–Fehlberg, compare your results to that from fourth-order Runge–Kutta method in Exercise 18.

▶24. Solve $y' = 2x^2 - y$, $y(0) = -1$ by the Runge–Kutta–Fehlberg method to $x = 2.0$. How large can h be and still get the solution accurate to 6 significant digits?

25. Add the results from the Runge–Kutta–Fehlberg method to Table 6.6.

26. In the algorithm for the Runge–Kutta–Fehlberg method, an expression for the error is given. Repeat Exercise 19 with the Runge–Kutta–Fehlberg method and compare the actual error to the value from the expression.

Section 6.4

▶27. Derive the formula for the second-order Adams method. Use the method of undetermined coefficients.

28. Use the formula of Exercise 27 to get values as in Example 6.1.

29. For the differential equation

$$\frac{dy}{dt} = y - t^2, \quad y(0) = 1,$$

starting values are known:

$$y(0.2) = 1.2186, \qquad y(0.4) = 1,4682,$$
$$y(0.6) = 1.7379.$$

Use the Adams method, fitting cubics with the last four (y, t) values and advance the solution to $t = 1.2$. Compare to the analytical solution.

▶30. For the equation

$$\frac{dy}{dt} = t^2 - t, \quad y(1) = 0,$$

the analytical solution is easy to find:

$$y = \frac{t^3}{3} - \frac{t^2}{2} + \frac{1}{6}.$$

If we use three points in the Adams method, what error would we expect in the numerical solution? Confirm your expectation by performing the computations.

31. Repeat Exercise 30, but use four points.

32. Solve Exercise 29 with Adams–Moulton fourth order method.

33. For the equation $y' = y * \sin(\pi x)$, $y(0) = 1$, get starting values by RKF for $x = 0.2$, 0.4, and 0.6 and then advance the solution to $x = 1.4$ by Adams–Moulton fourth order method.

34. Get the equivalent of Eqs. (6.16) and (6.17) for a third-order Adams–Moulton method.

35. Derive the interpolation formulas given in Section 6.4 that permit getting additional values to reduce the step size.

▶**36.** Use Eq. (6.18) on this problem

$$dy/dx = 2x + 2, \quad y(1) = 3.$$

a. Is instability indicated?

b. Compare the results with this method to those from the simple Euler method as in Tables 6.11 and 6.12.

37. Use Milne's method on the equation in Exercise in 36. Is there any indication of instability?

38. Parallel the theoretical demonstration of instability with Milne's method with the equation $dy/dx = Ax^n$, where A and n are constants. What do you conclude?

39. What is the error term for Hamming's method? Show that it is a stable method.

Section 6.5

40. The mathematical model of an electrical circuit is given by the equation

$$0.5\frac{d^2Q}{dt^2} + 6\frac{dQ}{dt} + 50Q = 24\sin 10t,$$

with $Q = 0$ and $i = dQ/dt = 0$ at $t = 0$. Express as a pair of first-order equations.

▶**41.** In the theory of beams, it is shown that the radius of curvature at any point is proportional to the bending moment:

$$EI\frac{y''}{\{1 + (y')^2\}^{3/2}} = M(x),$$

where y is the deflection of the neutral axis. In the usual approach, $(y')^2$ is neglected in comparison to unity, but if the beam has appreciable curvature, this is invalid. For the cantilever beam for which $y(0) = y'(0) = 0$, express the equation as a pair of simultaneous first-order equations.

42. A cantilever beam is 12 ft long and bears a uniform load of W lb/in. so that $M(x) = W * x^2/2$. Exercise 41 suggests that a simplified version of the differential equation can be used if the curvature of the beam is small. For what value of W, the value of the uniform load, does the simplified equation give a value for the deflection at the end of the beam that is in error by 5%?

▶**43.** Solve the pair of simultaneous equations

$$dx/dt = xy - t, \qquad x(0) = 1,$$
$$dy/dt = x + t, \qquad y(0) = 0,$$

by the modified Euler method from $t = 0$ to $t = 1.0$ in steps of 0.2.

44. Repeat Exercise 43, but with the Runge–Kutta–Fehlberg method. How accurate are these results? How much are the errors less than those of Exercise 43?

45. Use the first results of Exercise 44 to begin the Adams–Moulton method and then advance the solution to $x = 1.0$. Are the results as accurate as with the Runge–Kutta–Fehlberg method?

▶**46.** The motion of the compound spring system as sketched in Figure 6.7 is given by the solution of the pair of simultaneous equations

$$m_1\frac{d^2y_1}{dt^2} = -k_1y_1 - k_2(y_1 - y_2),$$
$$m_2\frac{d^2y_2}{dt^2} = k_2(y_1 - y_2),$$

where y_1 and y_2 are the displacements of the two masses from their equilibrium positions. The initial conditions are

$$y_1(0) = A, \quad y_1'(0) = B, \quad y_2(0) = C, \quad y_2'(0) = D.$$

Express as a set of first-order equations.

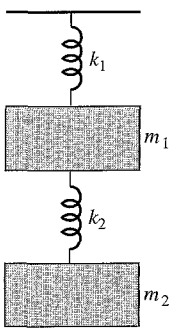

Figure 6.7

47. For the third-order equation

$$y''' + ty' - 2y = t, \quad y(0) = y''(0) = 0, \quad y'(0) = 1,$$

a. Solve for $y(0.2)$, $y(0.4)$, $y(0.6)$ by RKF.
b. Advance the solution to $t = 1.0$ with the Adams–Moulton method.
c. Estimate the accuracy of $y(1.0)$ in part (b).

48. Solve the equation in Exercise 47 by the Taylor-series method. How many terms are needed to be sure that $y(1.0)$ is correct to four significant digits?

49. If some simplifying assumptions are made, the equations of motion of a satellite around a central body are

$$\frac{d^2x}{dt^2} = \frac{-x}{r^3}, \quad \frac{d^2y}{dt^2} = \frac{-y}{r^3},$$

where

$$r = \sqrt{(x^2 + y^2)}, \qquad x(0) = 0.4,$$
$$y(0) = x'(0) = 0, \qquad y'(0) = 2.$$

a. Evaluate $x(t)$ and $y(t)$ from $t = 0$ to $t = 10$ in steps of 0.2. Use any of the single-step methods to do this.
b. Plot the curve for this range of t-values.
c. Estimate the period of the orbit.

Section 6.6

50. Equation 6.22 is for a *stiff* equation. If the coefficients of the equation for x' are changed, for what values is the system no longer stiff?

51. A pair of differential equations has the solution

$$x(t) = e^{-22t} - e^{-t},$$
$$y(t) = e^{-22t} + e^{-t},$$

with initial conditions of $x(0) = 0$, $y(0) = 2$.

a. What are the differential equations?
b. Is that system "stiff"?
c. What are the computed values for $x(0.2)$ and $y(0.2)$ if the equations of part (a) are solved with the simple Euler method, with $h = 0.1$?
d. Repeat part (c), but employing the method of Eq. (6.23). Is this answer closer to the correct value?
e. How small must h be to get the solutions at $t = 0.2$ accurate to four significant digits when using the simple Euler method?
f. Repeat part (e), but now for the method of Eq. (6.23).

▶**52.** When testing a linear system to see if it is "stiff" it is convenient to write it as

$$\begin{bmatrix} x \\ y \end{bmatrix}' = A \begin{bmatrix} x \\ y \end{bmatrix},$$

where the elements of matrix A are the multipliers of x and y in the equations. If the eigenvalues of A are all real and negative and differ widely in magnitude, the system is stiff. (One can get the eigenvalues from the characteristic polynomial as explained in Chapter 2 or with a computer algebra system.)

Suppose that A has these elements:

$$A = \begin{bmatrix} 19 & -20 \\ -20 & 19 \end{bmatrix}.$$

a. What are the eigenvalues of A? Would you call the system stiff?
b. Change the elements of A so that all are positive. What are the eigenvalues of A after this change? Does this make the system "nonstiff"?

53. The definition of a stiff equation as one whose coefficient matrix has negative eigenvalues that "differ widely in magnitude" is rather subjective. Propose an alternate definition of stiffness that is more specific.

Section 6.7

54. Suppose that a rod of length L is made from two dissimilar materials welded together end-to-end. From $x = 0$ to $x = X$, the thermal conductivity is k_1; from $x = X$ to $x = L$, it is k_2. How will the temperatures vary along the rod if $u = 0°$ at $x = 0$ and $u = 100°$ at $x = L$? Assume that Eq. (6.24) applies with $Q = 0$ and that the cross-section is constant.

55. What if k varies with temperature: $k = a + bu + cu^2$? What is the equation that must be solved to determine the temperature distribution along a rod of constant cross section?

56. Solve the boundary value problem

$$d^2x/dt^2 + t\,(dx/dt) - 3x = 3t, \quad x(0) = 1, \quad x(2) = 5$$

by "shooting." (The initial slope is near -1.5.) Use $h = 0.25$ and compare the results from the Runge–Kutta–Fehlberg method and modified Euler methods. Why are the results different? Is it possible to match the Runge–Kutta–Fehlberg method results when the modified Euler method is used? If so, show how this can be accomplished.

57. Repeat Exercise 56, but with smaller values for h. At what h-values with the Runge–Kutta–Fehlberg method are successive computations the same?

58. The boundary-value problem of Exercise 56 is linear. That means that the correct initial slope can be found

by interpolating from two trial values. Show that intermediate values from the computations obtained with these two trial values can themselves be interpolated to get correct intermediate values for $x(t)$.

59. If the equation of Exercise 56 is changed only slightly to

$$d^2x/dt^2 + x\,(dx/dt) - 3x = 3t, \quad x(0) = 1, \quad x(2) = 5,$$

it is no longer linear. Solve it by the shooting method using RKF. Do you find that more than two trials are needed to get the solution? What is the correct value for the initial slope? Use a value of h small enough to be sure that the results are correct to five significant digits.

60. Given this boundary-value problem:

$$\frac{d^2y}{d\theta^2} + \frac{y}{4} = 0, \quad y(0) = 0, \quad y(\pi) = 2,$$

which has the solution $y = 2\sin(\theta/2)$,

a. Solve, using finite difference approximations to the derivative with $h = \pi/4$ and tabulate the errors.
b. Solve again by finite differences but with a value of h small enough to reduce the maximum error to 0.5%. Can you predict from part (a) how small h should be?
c. Solve again by the shooting method. Find how large h can be to have maximum error of 0.5%.

61. Solve Exercise 56 though a set of equations where the derivatives are replaced by difference quotients. How small must h be to essentially match to the results of Exercise 56 when RKF was used?

62. Use finite difference approximations to the derivatives to solve Exercise 59. The equations will be nonlinear so they are not as easily solved. One way to approach the solution is to linearize the equations by replacing x in the second term with an approximate value, then using the results to refine this approximation successively. Solve it this way.

63. Solve this boundary-value problem by finite differences, first using $h = 0.2$, then with $h = 0.1$:

$$y'' + xy' - x^2y = 2x^3, \quad y(0) = 1, \quad y(1) = -1.$$

Assuming that errors are proportional to h^2, extrapolate to get an improved answer. Then, using a very small h-value in the shooting method, see if this agrees with your improved answer.

64. Repeat Exercise 60, except with these derivative boundary conditions:

$$y'(0) = 0, \qquad y'(\pi) = 1.$$

In part (a), compare to $y = -2\cos(\theta/2)$.

65. Solve through finite differences with four subintervals:

$$\frac{d^2y}{dx^2} + y = 0, \quad y'(0) + y(0) = 2,$$

$$y'\left(\frac{\pi}{2}\right) + y\left(\frac{\pi}{2}\right) = -1.$$

66. The most general form of boundary condition normally encountered in second-order boundary-value problems is a linear combination of the function and its derivatives at both ends of the region. Solve through finite difference approximations with four subintervals:

$$x'' - tx' + t^2x = t^3,$$
$$x(0) + x'(0) - x(1) + x'(1) = 3,$$
$$x(0) - x'(0) + x(1) - x'(1) = 2.$$

67. Repeat Exercise 63, but use the Runge–Kutta–Fehlberg method. The errors will not be proportional to h^2

68. Repeat Exercise 66, but use the modified Euler method.

69. Can a boundary-value problem be solved with a Taylor-series expansion of the function? If it can, use the Taylor-series technique for several of the above problems. If it cannot be used, provide an argument in support of this.

▶**70.** In solving a boundary-value problem with finite difference quotients, using smaller values for h improves the accuracy. Can one make h too small?

71. Compare the number of numerical operations used in Example 6.5 to get Tables 6.18 and 6.19.

Section 6.8

72. Consider the characteristic-value problem with k restricted to real values:

$$y'' - k^2y = 0, \quad y(0) = 0, \quad y(1) = 0.$$

a. Show analytically that there is no solution except the trivial solution $y = 0$.
b. Show, by setting up a set of difference equations corresponding to the differential equation with $h = 0.2$, that there are no real values for k for which a solution to the set exists.
c. Show, using the shooting method, that it is impossible to match $y(1) = 0$ for any real value of k [except if $y'(0) = 0$, which gives the trivial solution].

▶**73.** For the equation

$$y'' - 3y' + 2k^2y = 0, \quad y(0) = 0, \quad y(1) = 0,$$

find the principal eigenvalue and compare to $|k| = 2.46166$,

a. using $h = \frac{1}{2}$.

b. using $h = \frac{1}{3}$.

c. using $h = \frac{1}{4}$.

d. Assuming errors are proportional to h^2, extrapolate from parts (a) and (c) to get an improved estimate.

74. Using the principal eigenvalue, $k = 2.46166$, in Exercise 73, find y as a function of x over [0, 1]. This is the corresponding eigenfunction.

75. Parallel the computations of Exercise 73 to estimate the second eigenvalue. Compare to the analytical value of 4.56773.

76. Find the dominant eigenvalue and the corresponding eigenvector by the power method:

a. $\begin{bmatrix} 3 & 1 \\ 2 & 9 \end{bmatrix}$ b. $\begin{bmatrix} 2 & 3 \\ 6 & 5 \end{bmatrix}$ c. $\begin{bmatrix} 2 & 3 \\ 3 & -2 \end{bmatrix}$

d. $\begin{bmatrix} 6 & 2 & 0 \\ 2 & 4 & 1 \\ 0 & 1 & -1 \end{bmatrix}$ e. $\begin{bmatrix} 1 & 2 & 3 \\ 0 & 1 & 3 \\ 2 & 2 & 1 \end{bmatrix}$

[In part (c), the two eigenvalues are equal but of opposite sign.]

77. For the two matrices

$$A = \begin{bmatrix} -5 & 2 & 1 \\ 1 & -9 & -1 \\ 2 & -1 & 7 \end{bmatrix},$$

$$B = \begin{bmatrix} -4 + 2i & -1 & -5i \\ -3 & 7 + i & -i \\ 2 & -1 & 4 - i \end{bmatrix},$$

a. Put bounds on the eigenvalues using Gerschgorin's theorem.

b. Can you tell from part (a) whether either of the matrices is singular?

78. Use the power method or its variations to find all of the eigenvalues and eigenvectors for the matrices of Exercise 77. For matrix B, do you need to use complex arithmetic?

▶79. Get the eigenvalues for matrix A in Exercise 77 from its characteristic polynomial. Then invert the matrix and show that the eigenvalues are reciprocals but the eigenvectors are the same. How do the two characteristic polynomials differ? Can you get the second polynomial directly from the first? Can you do all of this for matrix B?

80. Repeat Exercise 79, but use the power method to get the dominant eigenvalue. Then shift by that amount and get the next one. Finally, get the third from the trace of A.

81. Find three matrices that convert one of the below diagonal elements to zero for matrix A of Exercise 77.

82. Use the matrices of Exercise 81 successively to make one element below the diagonal of A equal to zero, then multiply that product and the inverse of the rotation matrix (which is easy to find because it is just its transpose). We keep the eigenvalues the same because the two multiplications are a similarity transformation.

Repeat this process until all elements below the diagonal are less than 1.0E-4. When this is done, compare the elements now on the diagonal to the eigenvalues of A obtained by iteration. (This will take many steps. You will want to write a short computer program to carry it out.)

▶83. Use similarity transformations to reduce the matrix to upper Hessenberg. (Do no column or row interchanges.)

$$C = \begin{bmatrix} 3 & -1 & 2 & 7 \\ 1 & 2 & 0 & -1 \\ 4 & 2 & 1 & 1 \\ 2 & -1 & -2 & 2 \end{bmatrix}$$

84. Repeat Exercise 83 but with row/column interchanges that maximize the magnitude of the divisors.

85. Repeat Exercise 82 after first converting to upper Hessenberg. How many fewer iterations are needed?

Applied Problems and Projects

APP1. The mass in Figure 6.8 moves horizontally on the frictionless bar. It is connected by a spring to a support located centrally below the bar. The unstretched length of the spring is $L = \sqrt{(10)} = 3.1623$ m (meters); the spring constant is $k = 100$ N/m (newtons per meter); the mass of the block is 3 kg. Let $x(t)$ be the distance from the center of the bar to the location of the block at time t. Clearly the equilibrium position of the block is at $x = 1.0$ m (or $x = -1.0$ m). Let $y_0 = \sqrt{10}$ m (the unstretched length of the spring). This second-order differential equation describes the motion:

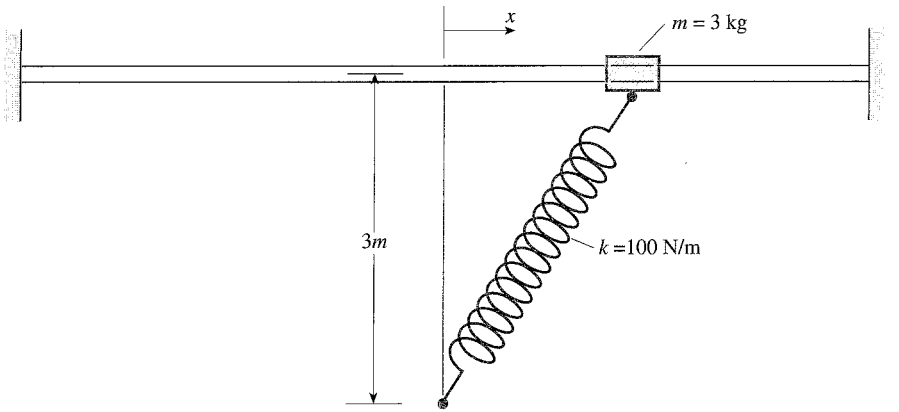

Figure 6.8

$$\frac{d^2x}{dt^2} = -\left(\frac{k}{m}\right)x\left(1 - \frac{y_0}{\sqrt{(x^2 + 9)}}\right).$$

a. Using both single-step and multistep methods, find the position of the block between $t = 0$ and $t = 10$ sec if $x_0 = 1.4$ and the initial velocity is zero.
b. Repeat part (a), but now with the spring stretched more at the start, $x_0 = 2.5$.
c. Use Maple and/or MATLAB to graph the motion for both parts (a) and (b). Compare your graphs to Figure 6.9.

APP2. The equation $y' = 1 + y^2$, $y(0) = 0$ has the solution $y = \tan(x)$. Use modified Euler method to compute values for $x = 0$ to $x = 1.6$ with a value for h small enough to obtain values that differ from the analytical by no more than ± 0.0005. What is the largest h-value to do this? y (x) becomes infinite at $x = \pi/2$. What happens if you try to integrate y' beyond this point? Is there some way you can solve the equation numerically from $x = 0$ to $x = 2$?

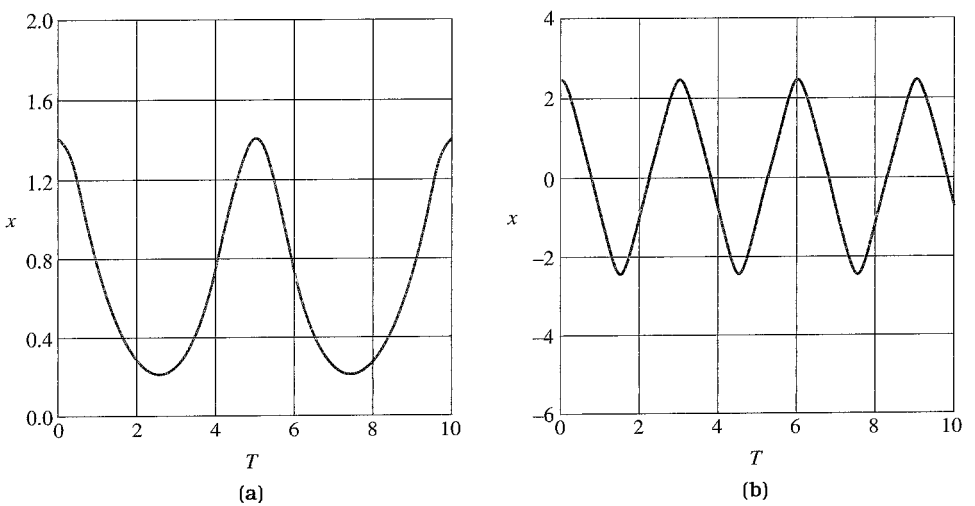

Figure 6.9

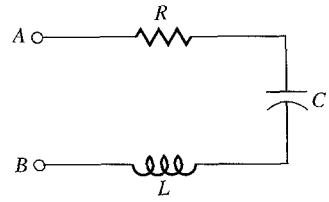

Figure 6.10

APP3. A nonlinear boundary-value problem is more difficult than a linear problem because many trials may be needed to get a good value for the initial slope. From three initial trials it should be possible to use a Muller's-type interpolation. Outline the steps of a program that will do this.

APP4. In an electrical circuit (Figure 6.10) that contains resistance, inductance, and capacitance (and every circuit does), the voltage drop across the resistance is iR (i is current in amperes, R is resistance in ohms), across the inductance it is L, (di/dt) (L is inductance in henries), and across the capacitance it is q/C (q is charge in the capacitor in coulombs, C is capacitance in farads). We then can write, for the voltage, difference between points A and B,

$$V_{AB} = L\frac{di}{dt} + Ri + \frac{q}{C}.$$

Differentiating with respect to t and remembering that $dq/dt = i$, we have a second-order differential equation;

$$L\frac{d^2i}{dt^2} + R\frac{di}{dt} + \frac{1}{C}i = \frac{dV}{dt}.$$

If the voltage V_{AB} (which has previously been 0 V) is suddenly brought to 15 V (let us say, by connecting a battery across the terminals) and maintained steadily at 15 V (so $dV/dt = 0$), current will flow through the circuit. Use an appropriate numerical method to determine how the current varies with time between 0 and 0.1 sec if $C = 1000$ μf, $L = 50$ mH, and $R = 4.7$ ohms; use Δt of 0.002 sec. Also determine how the voltage builds up across the capacitor during this time. You may want to compare the computations with the analytical solution.

APP5. Repeat App 4, but let the voltage source be a 60-Hz sinusoidal input:

$$V_{AB} = 15\sin(120\pi t).$$

How closely does the voltage across the capacitor resemble a sine wave during the last full cycle of voltage variation?

APP6. After the voltages have stabilized in APP4 (15 V across the capacitor), the battery is shorted so that the capacitor discharges through the resistance and inductor. Follow the current and the capacitor voltages for 0.1 sec, again with $\Delta t = 0.002$ sec. The oscillations of decreasing amplitude are called *damped oscillations*. If the calculations are repeated but with the resistance value increased, the oscillations will be damped out more quickly; at $R = 14.14$ ohms the oscillations should disappear; this is called *critical damping*. Perform numerical computations with values of R increasing from 4.7 to 22 ohms to confirm that critical damping occurs at 14.14 ohms.

APP7. Cooling fins are often welded to objects in which heat is generated to conduct the heat away, thus controlling the temperature. If the fin loses heat by radiation to the surroundings the rate of heat loss from the fin is proportional to the difference in fourth powers of the fin temperature and the surroundings, both measured in absolute degrees. The equation reduces to

$$d^2u/dx^2 = k(u^4 - T^4)$$

where u is the fin temperature, T is the surroundings temperature, and x is the distance along the fin. k is a constant. For a fin of given length L, this is not difficult to solve numerically if $u(0)$ and $u(L)$ are known. Solve for $u(x)$, the distribution of temperature along the fin, if $T = 300$, $u(0) = 450$, $u(20) = 350$, $k = 0.23$, utilizing any of the methods for a boundary-value problem. Use a value for h small enough to get temperatures accurate to 0.1 degree.

APP8. In APP7, suppose the fin is of infinite length and we can assume that $\lim (u(x)) = 0$ as $x \to \infty$. Can this problem be solved numerically? If so, get the solution for $u(x)$ between $x = 0$ and $x = 20$.

APP9. A Foucault pendulum is one free to swing in both the x- and y-directions. It is frequently displayed in science museums to exhibit the rotation of the earth, which causes the pendulum to swing in directions that continuously vary. The equations of motion are

$$\ddot{x} - 2\omega \sin \psi \dot{y} + k^2 x = 0,$$
$$\ddot{y} + 2\omega \sin \psi \dot{x} + k^2 y = 0,$$

when damping is absent (or compensated for). In these equations, the dots over the variable represent differentiation with respect to time. Here ω is the angular velocity of the earth's rotation (7.29×10^{-5} sec^{-1}), ψ is the latitude, $k^2 = g/\ell$ where ℓ is the length of the pendulum. How long will it take a 10-m-long pendulum to rotate its plane of swing by 45° at the latitude where you live? How long if located in Quebec, Canada?

APP10. Condon and Odishaw (1967) discuss Duffing's equation for the flux ϕ in a transformer. This nonlinear differential equation is

$$\ddot{\phi} + \omega_0^2 \phi + b\phi^3 = \frac{\omega}{N} E \cos \omega t.$$

In this equation, $E \sin \omega t$ is the sinusoidal source voltage and N is the number of turns in the primary winding, while ω_0 and b are parameters of the transformer design. Make a plot of ϕ versus t (and compare to the source voltage) if $E = 165$, $\omega = 120\pi$, $N = 600$, $\omega_0^2 = 83$, and $b = 0.14$. For approximate calculations, the nonlinear term $b\phi^3$ is sometimes neglected. Evaluate your results to determine whether this makes a significant error in the results.

APP11. Ethylene oxide is an important raw material for the manufacture of organic chemicals. It is produced by reacting ethylene and oxygen together over a silver catalyst. Laboratory studies gave the equation shown.

It is planned to use this process commercially by passing the gaseous mixture through tubes filled with catalyst. The reaction rate varies with pressure, temperature, and concentrations of ethylene and oxygen, according to this equation:

$$r = 1.7 \times 10^6 e^{-9716/T} \left(\frac{P}{14.7}\right) C_E^{0.328} C_O^{0.672},$$

where

r = reaction rate (units of ethylene oxide formed per lb of catalyst per hr),
T = temperature, °K (°C + 273),
P = absolute pressure (lb/in.^{2}),
C_E = concentration of ethylene,
C_O = concentration of oxygen.

Under the planned conditions, the reaction will occur, as the gas flows through the tube, according to the equation

$$\frac{dx}{dL} = 6.42r,$$

where

x = fraction of ethylene converted to ethylene oxide,
L = length of reactor tube (ft).

The reaction is strongly exothermic, so that it is necessary to cool the tubular reactor to prevent overheating. (Excessively high temperatures produce undesirable side reactions.) The reactor will be cooled by surrounding the catalyst tubes with boiling coolant under pressure so that the tube walls are kept at 225°C. This will remove heat proportional to the temperature difference between the gas and the boiling water. Of course, heat is generated by the reaction. The net effect can be expressed by this equation for the temperature change per foot of tube, where B is a design parameter:

$$\frac{dT}{dL} = 24{,}302r - B(T - 225).$$

For preliminary computations, it has been agreed that we can neglect the change in pressure as the gases flow through the tubes; we will use the average pressure of $P = 22$ lb/in.^{2} absolute. We will also neglect the difference between the catalyst temperature (which should be used to find the reaction rate) and the gas temperature. You are to compute the length of tubes required for 65% conversion of ethylene if the inlet temperature is 250°C. Oxygen is consumed in proportion to the ethylene converted; material balances show that the concentrations of ethylene and oxygen vary with x, the fraction of ethylene converted, as follows:

$$C_E = \frac{1 - x}{4 - 0.375x},$$

$$C_O = \frac{1 - 1.125x}{4 - 0.375x}.$$

The design parameter B will be determined by the diameter of tubes that contain the catalyst. (The number of tubes in parallel will be chosen to accommodate the quantities of materials flowing through the reactor.) The tube size will be chosen to control the maximum temperature of the reaction, as set by the minimum allowable value of B. If the tubes are too large in diameter (for which the value of B is small), the temperatures will run wild. If the tubes are *too small* (giving a large value to B), so much heat is lost that the reaction tends to be quenched. In your studies, vary B to find the least value that will keep the maximum temperature below 300°C. Permissible values for the parameter B are from 1.0 to 10.0.

In addition to finding how long the tubes must be, we need to know how the temperature varies with x and with the distance along the tubes. To have some indication of the controllability of the process, you are also asked to determine how much the outlet temperature will change for a 1°C change in the inlet temperature, using the value of B already determined.

APP12. An ecologist has been studying the effects of the environment on the population of field mice. Her research shows that the number of mice born each month is proportional to the number of females in the group and that the fraction of females is normally constant in any group. This implies that the number of births per month is proportional to the total population.

She has located a test plot for further research, which is a restricted area of semiarid land. She has constructed barriers around the plot so mice cannot enter or leave. Under the conditions of the experiment, the food supply is limited, and it is found that the death rate is affected as a result, with mice dying of starvation at a rate proportional to some power of the population. (She also hypothesizes that when the mother is undernourished, the babies have less chance for survival and that starving males tend to attack one another, but these factors are only speculation.)

The net result of this scientific analysis is the following equation, with N being the number of mice at time t (with t expressed in months). The ecologist has come to you for help in solving the equation; her calculus doesn't seem to apply.

$$\frac{dN}{dt} = aN - BN^{1.7}, \qquad \text{with } B \text{ given by Table 6.20.}$$

Table 6.20

t	B	t	B
0	0.0070	5	0.0013
1	0.0036	6	0.0028
2	0.0011	7	0.0043
3	0.0001	8	0.0056
4	0.0004		

As the season progresses, the amount of vegetation varies. The ecologist accounts for this change in the food supply by using a "constant" B that varies with the season.

If 100 mice were initially released into the test plot and if $a = 0.9$, estimate the number of mice as a function of t, for $t = 0$ to $t = 8$.

APP13. A certain chemical company produces a product that is a mixture of two ingredients, A and B. In order to ensure that the product is homogeneous, A and B are fed into a well-mixed tank that holds 100 gal. The desired product must contain two parts of A to one part of B within certain specifications. The normal flows of A and B into the tank are 4 and 2 gal/min. There is no volume change when these are mixed, so the outflow is 6 gal/min and the holding time in the tank is $100/6 = 16.66$ min. Due to an unfortunate accident, the flow of ingredient B is cut off and before this is noticed and corrected, the ratio of A to B in the tank has increased to 10 parts of A to 1 part of B. (There are still 100 gal in the tank.) Set up equations that give the ratio of A to B in the tank as a function of time after the flow of B has been restored to its normal value of 2 gal/min. How long will it take until the output from the tank reaches 2 parts A to 0.99 parts B? How much product is produced (and discarded because it is not up to specification) during this time? How would you suggest that this time to reach specification be reduced?