CHAPTER 8

Numerical Methods

Up to this point we have discussed methods for solving differential equations by using analytical techniques such as integration or series expansions. Usually, the emphasis was on finding an exact expression for the solution. Unfortunately, there are many important problems in engineering and science, especially nonlinear ones, to which these methods either do not apply or are very complicated to use. In this chapter we discuss an alternative approach, the use of numerical approximation methods to obtain an accurate approximation to the solution of an initial value problem. We present these methods in the simplest possible context, namely, a single scalar first-order equation. However, they can readily be extended to systems of first-order equations, and this is outlined briefly in Section 8.5. The procedures described here can be executed easily on a wide variety of computational devices, from smartphones to supercomputers.

8.1 The Euler or Tangent Line Method

To discuss the development and use of numerical approximation procedures, we will concentrate mainly on the first-order initial value problem consisting of the differential equation

$$\frac{dy}{dt} = f(t, y) \tag{1}$$

and the initial condition

$$y(t_0) = y_0. \tag{2}$$

We assume that the functions f and f_y are continuous on some rectangle in the ty-plane containing the point (t_0, y_0). Then, by Theorem 2.4.2, there exists a unique solution $y = \phi(t)$ of the given problem in some interval about t_0. If equation (1) is nonlinear, then the interval of existence of the solution may be difficult to determine and may have no simple relationship to the function f. However, in all our discussions we assume that there is a unique solution of the initial value problem (1), (2) in the interval of interest.

In Section 2.7 we described the oldest and simplest numerical approximation method, namely, the Euler or tangent line method. To derive this method, let us write the differential equation (1) at the point $t = t_n$ in the form

$$\frac{d\phi}{dt}(t_n) = f(t_n, \phi(t_n)). \tag{3}$$

Then we approximate the derivative in equation (3) by the corresponding (forward) difference quotient, obtaining

$$\frac{\phi(t_{n+1}) - \phi(t_n)}{t_{n+1} - t_n} \cong f(t_n, \phi(t_n)). \tag{4}$$

Finally, if we replace $\phi(t_{n+1})$ and $\phi(t_n)$ by their approximate values y_{n+1} and y_n, respectively, and solve for y_{n+1}, we obtain the Euler formula

$$y_{n+1} = y_n + f(t_n, y_n)(t_{n+1} - t_n), \quad n = 0, 1, 2, \ldots. \tag{5}$$

If the step size $t_{n+1} - t_n$ has a uniform value h for all n and if we denote $f(t_n, y_n)$ by f_n, then equation (5) simplifies to

$$y_{n+1} = y_n + hf_n, \quad n = 0, 1, 2, \ldots. \tag{6}$$

Euler's method consists of repeatedly evaluating equation (5) or (6), using the result of each step to execute the next step. In this way we obtain a sequence of values $y_0, y_1, y_2, \ldots, y_n, \ldots$ that approximate the values of the solution $\phi(t)$ at the points $t_0, t_1, t_2, \ldots, t_n, \ldots$.

A computer program for Euler's method has a structure such as that shown below. The specific instructions can be written in any convenient programming language.

The Euler Method

Step 1. **define** $f(t, y)$
Step 2. **input** initial values $t = t0$ and $y = y0$
Step 3. **input** step size h and number of steps n
Step 4. **output** $t0$ and $y0$
Step 5. **for** j from 1 to n **do**
Step 6. $f_n = f(t, y)$
$y = y + h * f_n$
$t = t + h$
Step 7. **output** t and y
Step 8. **end**

Some examples of Euler's method appear in Section 2.7. As another example, consider the initial value problem

$$y' = 1 - t + 4y, \tag{7}$$

$$y(0) = 1. \tag{8}$$

Equation (7) is a first-order linear equation, and you can easily verify that the solution satisfying the initial condition (8) is

$$y = \phi(t) = \frac{1}{4}t - \frac{3}{16} + \frac{19}{16}e^{4t}. \tag{9}$$

Since the exact solution is known, we do not need numerical methods to approximate the solution of the initial value problem (7), (8). On the other hand, the availability of the exact solution makes it easy to monitor the accuracy of any numerical procedure that we use on this problem. We will use this problem throughout the chapter to illustrate and to compare different numerical methods. The solutions of equation (7) diverge rather rapidly from each other, so we should expect that it will be fairly difficult to approximate the solution (9) well over any interval of moderate length. Indeed, this is the reason for choosing this particular problem; it will be relatively easy to observe the benefits of using more efficient methods.

EXAMPLE 1

Using the Euler formula (6) and step sizes $h = 0.05, 0.025, 0.01$, and 0.001, determine approximate values of the solution $y = \phi(t)$ of the problem (7), (8) on the interval $0 \le t \le 2$.

Solution:

The indicated calculations were carried out on a computer, and some of the results are shown in Table 8.1.1. Their accuracy is not particularly impressive. For $h = 0.01$ the percentage error is 3.85% at $t = 0.5$, 7.49% at $t = 1.0$, and 14.4% at $t = 2.0$. The corresponding percentage errors for $h = 0.001$ are 0.40%, 0.79%, and 1.58%, respectively. Observe that if $h = 0.001$, then it requires 2000 steps to traverse the interval from $t = 0$ to $t = 2$. Thus considerable computation is needed to obtain even reasonably good accuracy for this problem using the Euler method. When we discuss other numerical approximation methods later in this chapter, we will find that it is possible to obtain comparable or better accuracy with much larger step sizes and many fewer computational steps.

where $\overline{t_n}$ is some point in $t_n < \overline{t_n} < t_n + h$. Then, noting that $\phi(t_n + h) = \phi(t_{n+1})$ and $\phi'(t_n) = f(t_n, \phi(t_n))$, we can rewrite equation (19) as

$$\phi(t_{n+1}) = \phi(t_n) + hf(t_n, \phi(t_n)) + \frac{1}{2}\phi''(\overline{t_n})h^2. \tag{20}$$

Now let us use the Euler formula to calculate an approximation to $\phi(t_{n+1})$ under the assumption that we know the correct value for y_n at t_n, namely $y_n = \phi(t_n)$. The result is

$$y^*_{n+1} = \phi(t_n) + hf(t_n, \phi(t_n)), \tag{21}$$

where the asterisk is used to designate this hypothetical approximate value for $\phi(t_{n+1})$. The difference between $\phi(t_{n+1})$ and y^*_{n+1} is the local truncation error for the $(n+1)^{\text{st}}$ step in the Euler method, which we will denote by e_{n+1}. Thus, by subtracting equation (21) from equation (20), we find that

$$e_{n+1} = \phi(t_{n+1}) - y^*_{n+1} = \frac{1}{2}\phi''(\overline{t_n})h^2, \tag{22}$$

since the remaining terms in equations (20) and (21) cancel.

Thus the local truncation error for the Euler method is proportional to the square of the step size h, and the proportionality factor depends on the second derivative of the solution ϕ. The expression given by equation (22) depends on n and, in general, is different for each step. A uniform bound, valid on an interval $[a, b]$, is given by

$$|e_n| \le \frac{1}{2}Mh^2, \tag{23}$$

where M is the maximum of $|\phi''(t)|$ on the interval $[a, b]$. Since equation (23) is based on a consideration of the worst possible case—that is, the largest possible value of $|\phi''(t)|$—it may well be a considerable overestimate of the actual local truncation error in some parts of the interval $[a, b]$.

One use of equation (23) is to choose a step size that will result in a local truncation error no greater than some given tolerance level. For example, if the local truncation error must be no greater than ϵ, then from equation (23) we have

$$\frac{1}{2}Mh^2 \le \epsilon \text{ or } h \le \sqrt{\frac{2\epsilon}{M}}. \tag{24}$$

The primary difficulty in using equation (22), (23), or (24) lies in estimating $|\phi''(t)|$ or M. However, the central fact expressed by these equations is that the local truncation error is proportional to h^2. For example, if a new value of h is used that is one-half of its original value, then the resulting error will be reduced to one-fourth of its previous value.

More important than the local truncation error is the global truncation error E_n. The analysis for estimating E_n is much more difficult than that for e_n. Nevertheless, it can be shown that the global truncation error in using the Euler method on a finite interval is no greater than a constant times h. Thus

$$|E_n| \le Kh \tag{25}$$

for some constant K; see Problem 20 for more details. The Euler method is called a **first-order method** because its global truncation error is proportional to the first power of the step size.

Because it is more accessible, we will hereafter use the local truncation error as our principal measure of the accuracy of a numerical method and for comparing different methods. If we have *a priori* information about the solution of the given initial value problem, we can use the result (22) to obtain more precise information about how the local truncation error varies with t.

As an example, consider the illustrative problem

$$y' = 1 - t + 4y, \quad y(0) = 1 \tag{26}$$

on the interval $0 \le t \le 2$. Let $y = \phi(t)$ be the solution of the initial value problem (26). Then, as noted previously,

$$\phi(t) = \frac{1}{16}\left(4t - 3 + 19e^{4t}\right)$$

and therefore

$$\phi''(t) = 19e^{4t}.$$

Equation (22) then states that

$$e_{n+1} = \frac{19e^{4\overline{t_n}}h^2}{2}, \quad t_n < \overline{t_n} < t_n + h. \tag{27}$$

The appearance of the factor 19 and the rapid growth of e^{4t} explain why the results in Table 8.1.1 are not very accurate.

For instance, for $h = 0.05$ the error in the first step is

$$e_1 = \phi(t_1) - y_1 = \frac{19e^{4\overline{t_0}}(0.0025)}{2}, \quad 0 < \overline{t_0} < 0.05.$$

It is clear that e_1 is positive, and since $e^{4\overline{t_0}} < e^{0.2}$, we have

$$e_1 \le \frac{19e^{0.2}(0.0025)}{2} \cong 0.02901. \tag{28}$$

Note also that $e^{4\overline{t_0}} > 1$; hence $e_1 > \frac{19}{2}(0.0025) = 0.02375$. The actual error is 0.02542. It follows from equation (27) that the error becomes progressively worse with increasing t; this is also clearly shown by the results in Table 8.1.1. Similar computations for bounds for the local truncation error give

$$1.0617 \cong \frac{19e^{3.8}(0.0025)}{2} \le e_{20} \le \frac{19e^{4}(0.0025)}{2} \cong 1.2967 \tag{29}$$

in going from 0.95 to 1.0 and

$$57.96 \cong \frac{19e^{7.8}(0.0025)}{2} \le e_{40} \le \frac{19e^{8}(0.0025)}{2} \cong 70.80 \tag{30}$$

in going from 1.95 to 2.0.

These results indicate that for this problem, the local truncation error is about 2500 times larger near $t = 2$ than near $t = 0$. Thus, to reduce the local truncation error to an acceptable level throughout $0 \le t \le 2$, we must choose a step size h based on an analysis near $t = 2$. Of course, this step size will be much smaller than necessary near $t = 0$. For example, to achieve a local truncation error of 0.01 for this problem, we need a step size of about 0.00059 near $t = 2$ and a step size of about 0.032 near $t = 0$. The use of a uniform step size that is smaller than necessary over much of the interval results in more calculations than necessary, more time consumed, and possibly more danger of unacceptable round-off errors.

Another approach is to keep the local truncation error approximately constant throughout the interval by gradually reducing the step size as t increases. In the example problem, we would need to reduce h by a factor of about 50 in going from $t = 0$ to $t = 2$. A method that provides for variations in the step size is called **adaptive**. All modern computer codes for solving differential equations have the capability of adjusting the step size as needed. We will return to this question in the next section.

Problems

N 1. Complete the calculations leading to the entries in columns three and four of Table 8.1.1.

N 2. Complete the calculations leading to the entries in columns three and four of Table 8.1.2.

In each of Problems 3 through 7, find approximate values of the solution of the initial value problem at $t = 0.1, 0.2, 0.3$, and 0.4.

N a. Use the Euler method with $h = 0.05$.

N b. Use the Euler method with $h = 0.025$.

N c. Use the backward Euler method with $h = 0.05$.

N d. Use the backward Euler method with $h = 0.025$.

3. $y' = 5t - 3\sqrt{y}, \quad y(0) = 2$

4. $y' = 2y - 3t, \quad y(0) = 1$

5. $y' = 2t + e^{-ty}, \quad y(0) = 1$

6. $y' = (y^2 + 2ty)/(3 + t^2), \quad y(0) = 0.5$

7. $y' = (t^2 - y^2)\sin y, \quad y(0) = -1$

In each of Problems 8 through 12, find approximate values of the solution of the initial value problem at $t = 0.5$, 1.0, 1.5, and 2.0.

N **a.** Use the Euler method with $h = 0.025$.

N **b.** Use the Euler method with $h = 0.0125$.

N **c.** Use the backward Euler method with $h = 0.025$.

N **d.** Use the backward Euler method with $h = 0.0125$.

8. $y' = 0.5 - t + 2y, \quad y(0) = 1$

9. $y' = 5t - 3\sqrt{y}, \quad y(0) = 2$

10. $y' = 2t + e^{-ty}, \quad y(0) = 1$

11. $y' = (4 - ty)/(1 + y^2), \quad y(0) = -2$

12. $y' = (y^2 + 2ty)/(3 + t^2), \quad y(0) = 0.5$

13. Using three terms in the Taylor series given in equation (12) and taking $h = 0.1$, determine approximate values of the solution of the illustrative example $y' = 1 - t + 4y$, $y(0) = 1$ at $t = 0.1$ and 0.2. Compare the results with those using the Euler method and with the exact values. *Hint:* If $y' = f(t, y)$, what is y''?

In each of Problems 14 and 15,

N **a.** Estimate the local truncation error for the Euler method in terms of the solution $y = \phi(t)$.

N **b.** Obtain a bound for e_{n+1} in terms of t and $\phi(t)$ that is valid on the interval $0 \le t \le 1$.

N **c.** By using a formula for the solution, obtain a more accurate error bound for e_{n+1}.

N **d.** For $h = 0.1$ compute a bound for e_1 and compare it with the actual error at $t = 0.1$.

N **e.** Compute a bound for the error e_4 in the fourth step.

14. $y' = 2y - 1, \quad y(0) = 1$

15. $y' = \frac{1}{2} - t + 2y, \quad y(0) = 1$

In each of Problems 16 through 18, obtain a formula for the local truncation error for the Euler method in terms of t and the exact solution $y = \phi(t)$.

16. $y' = 5t - 3\sqrt{y}, \quad y(0) = 2$

17. $y' = \sqrt{t + y}, \quad y(1) = 3$

18. $y' = 2t + e^{-ty}, \quad y(0) = 1$

19. Consider the initial value problem

$$y' = \cos(5\pi t), \quad y(0) = 1.$$

N **a.** Determine approximate values of $\phi(t)$ at $t = 0.2$, 0.4, and 0.6 using the Euler method with $h = 0.2$.

N **b.** Determine the solution $y = \phi(t)$, and draw a graph of $y = \phi(t)$ for $0 \le t \le 1$.

G **c.** Draw a broken-line graph for the approximate solution, and compare it with the graph of the exact solution.

N **d.** Repeat the computation of part a for $0 \le t \le 0.4$, but take $h = 0.1$.

N **e.** Show by computing the local truncation error that neither of these step sizes is sufficiently small.

N **f.** Determine a value of h to ensure that the local truncation error is less than 0.05 throughout the interval $0 \le t \le 1$. That such a small value of h is required results from the fact that $\max |\phi''(t)|$ is large.

20. In this problem we discuss the global truncation error associated with the Euler method for the initial value problem $y' = f(t, y)$, $y(t_0) = y_0$. When the functions f and f_y are continuous in a closed, bounded region R of the ty-plane that includes the point (t_0, y_0), it can be shown that there exists a constant L such that $|f(t, y) - f(t, \tilde{y})| \le L|y - \tilde{y}|$, where (t, y) and $(t, \tilde{y})$ are any two points in R with the same t coordinate (see Problem 14 of Section 2.8). Further, we assume that f_t is continuous, so the solution ϕ has a continuous second derivative.

a. Using equation (20), show that

$$\begin{aligned} |E_{n+1}| &\le |E_n| + h|f(t_n, \phi(t_n)) - f(t_n, y_n)| \\ &\quad + \frac{1}{2}h^2|\phi''(\bar{t}_n)| \\ &\le \alpha|E_n| + \beta h^2, \end{aligned} \tag{31}$$

where $\alpha = 1 + hL$ and $\beta = \max \frac{1}{2}|\phi''(t)|$ on $t_0 \le t \le t_n$.

b. Assume that if $E_0 = 0$, and if $|E_n|$ satisfies equation (31), then $|E_n| \le \beta h^2(\alpha^n - 1)/(\alpha - 1)$ for $\alpha \ne 1$. Use this result to show that

$$|E_n| \le \frac{(1 + hL)^n - 1}{L}\beta h. \tag{32}$$

Equation (32) gives a bound for $|E_n|$ in terms of h, L, n, and β. Notice that for a fixed h, this error bound increases with increasing n; that is, the error bound increases with distance from the starting point t_0.

c. Show that $(1 + hL)^n \le e^{nhL}$; hence

$$|E_n| \le \frac{e^{nhL} - 1}{L}\beta h.$$

If we select an ending point T greater than t_0 and then choose the step size h so that n steps are required to traverse the interval $[t_0, T]$, then $nh = T - t_0$, and

$$|E_n| \le \frac{e^{(T - t_0)L} - 1}{L}\beta h = Kh,$$

which is equation (25). Note that K depends on the length $T - t_0$ of the interval and on the constants L and β that are determined from the function f.

21. Derive an expression analogous to equation (22) for the local truncation error for the backward Euler formula. *Hint:* Construct a suitable Taylor approximation to $\phi(t)$ about $t = t_{n+1}$.

22. Using a step size $h = 0.05$ and the Euler method, but retaining only three digits throughout the computations, determine approximate values of the solution at $t = 0.1$, 0.2, 0.3, and 0.4 for each of the following initial value problems:

N **a.** $y' = 1 - t + 4y, \quad y(0) = 1$

N **b.** $y' = 3 + t - y, \quad y(0) = 1$

N **c.** $y' = 2y - 3t, \quad y(0) = 1$

Compare the results of a with those obtained in Example 1 and in Problem 1 and the results of c with those obtained in Problem 4. The small differences between some of those results rounded to three digits and the present results are due to round-off error. The round-off error would become important if the computation required many steps.

23. The following problem illustrates a danger that occurs because of round-off error when nearly equal numbers are subtracted and the difference is then multiplied by a large number. Evaluate the quantity

$$1000 \cdot \begin{vmatrix} 6.010 & 18.04 \\ 2.004 & 6.000 \end{vmatrix}$$

in the following ways:

N **a.** First round each entry in the determinant to two digits.

N **b.** First round each entry in the determinant to three digits.

N **c.** Retain all four digits. Compare this value with the results in parts a and b.

24. The distributive law $a(b - c) = ab - ac$ does not hold, in general, if the products are rounded off to a smaller number of digits. To show this in a specific case, take $a = 0.22, b = 3.19$, and $c = 2.17$. After each multiplication, round off the last digit.

8.2 Improvements on the Euler Method

For many problems the Euler method requires a very small step size to produce sufficiently accurate results. Much effort has been devoted to the development of more accurate methods. In the next three sections, we will discuss some of these methods. Consider the initial value problem

$$y' = f(t, y), \quad y(t_0) = y_0 \tag{1}$$

and let $y = \phi(t)$ denote its solution. Recall from equation (10) of Section 8.1 that by integrating the given differential equation from t_n to t_{n+1}, we obtain

$$\phi(t_{n+1}) = \phi(t_n) + \int_{t_n}^{t_{n+1}} f(t, \phi(t))\, dt. \tag{2}$$

The Euler formula

$$y_{n+1} = y_n + hf(t_n, y_n) \tag{3}$$

is obtained by replacing the integrand $f(t, \phi(t))$ in equation (2) by its approximate value $f(t_n, y_n)$ at the left endpoint of the interval of integration. Other approximations of the definite integral lead to other numerical solution methods for initial value problems.

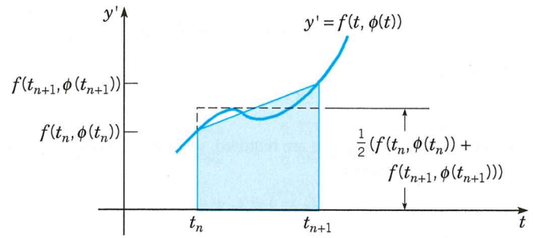

FIGURE 8.2.1 Derivation of the improved Euler method.

Improved Euler Formula. A better approximate formula for the solution of initial value problem (1) can be obtained if the definite integral in equation (2) is approximated more accurately. One way to do this is to replace the integrand by the average of its values at the two endpoints, namely, $\frac{1}{2}\big(f(t_n, \phi(t_n)) + f(t_{n+1}, \phi(t_{n+1}))\big)$. This is equivalent to approximating the area under the curve in Figure 8.2.1 between $t = t_n$ and $t = t_{n+1}$ by the area of the shaded trapezoid. Further, we replace $\phi(t_n)$ and $\phi(t_{n+1})$ by their respective approximate values y_n and y_{n+1}. In this way we obtain, from equation (2),

$$y_{n+1} = y_n + \frac{f(t_n, y_n) + f(t_{n+1}, y_{n+1})}{2}h. \tag{4}$$

Since the unknown y_{n+1} appears as one of the arguments of f on the right-hand side of equation (4), this equation defines y_{n+1} implicitly rather than explicitly. Depending on the nature of the function f, it might be fairly difficult to solve equation (4) for y_{n+1}. This difficulty

Variation of Step Size. In Section 8.1 we mentioned the possibility of adjusting the step size as a calculation proceeds so as to maintain the local truncation error at a more or less constant level. The goal is to use no more steps than necessary and, at the same time, to keep some control over the accuracy of the approximation. Here, we will describe how this can be done. First, we choose the error tolerance ϵ, which is the local truncation error that we are willing to accept. Suppose that after n steps we have reached the point (t_n, y_n). We choose a step size h and calculate y_{n+1}. Next we need to estimate the error we have made in calculating y_{n+1}. Not knowing the actual solution, the best that we can do is to use a more accurate method and repeat the calculation starting from (t_n, y_n). For example, if we used the Euler method for the original calculation, we might repeat it with the improved Euler method. Then the difference between the two calculated values is an estimate e_{n+1}^{est} of the error in using the original method. If the estimated error is larger than the error tolerance ϵ, then we adjust the step size and repeat the calculation. The key to making this adjustment efficiently is knowing how the local truncation error e_{n+1} depends on the step size h. For the Euler method, the local truncation error is proportional to h^2, so to bring the estimated error down (or up) to the tolerance level ϵ, we must multiply the original step size by the factor $\sqrt{\epsilon / e_{n+1}^{\text{est}}}$.

To illustrate this procedure, consider the example problem (7):

$$y' = 1 - t + 4y, \quad y(0) = 1.$$

Suppose that we choose the error tolerance ϵ to be 0.05. You can verify that after one step with $h = 0.1$, we obtain the values 1.5 and 1.595 from the Euler method and the improved Euler method, respectively. Thus the estimated error in using the Euler method is 0.095. Since this is larger than the tolerance level of 0.05, we need to adjust the step size downward by the factor $\sqrt{0.05/0.095} \cong 0.73$. Rounding downward to be conservative, let us choose the adjusted step size $h = 0.07$. Then, from the Euler formula, we obtain

$$y_1 = 1 + (0.07)\, f(0, 1) = 1.35 \cong \phi(0.07).$$

Then, using the improved Euler method with $h = 0.07$, we obtain $y_1 = 1.39655$, so the estimated error in using the Euler formula is 0.04655, which is slightly less than the specified tolerance. The actual error, based on a comparison with the exact solution, is somewhat greater, namely, 0.05122.

We can follow the same procedure at each step of the calculation, thereby keeping the local truncation error approximately constant throughout the entire numerical process. Modern adaptive codes for solving differential equations adjust the step size in very much this way as they proceed, although they usually use more accurate formulas than the Euler and improved Euler formulas. Consequently, they are able to achieve both efficiency and accuracy by using very small steps only where they are really needed.

Problems

N **1.** Complete the calculations leading to the entries in columns four and five of Table 8.2.1.

In each of Problems 2 through 6, find approximate values of the solution of the given initial value problem at $t = 0.1$, 0.2, 0.3, and 0.4. Compare the results with those obtained by the Euler method and the backward Euler method in Section 8.1 and with the exact solution (if available).

N **a.** Use the improved Euler method with $h = 0.05$.
N **b.** Use the improved Euler method with $h = 0.025$.
N **c.** Use the improved Euler method with $h = 0.0125$.

2. $y' = 3 + t - y, \quad y(0) = 1$
3. $y' = 2y - 3t, \quad y(0) = 1$
4. $y' = 2t + e^{-ty}, \quad y(0) = 1$
5. $y' = (y^2 + 2ty)/(3 + t^2), \quad y(0) = 0.5$
6. $y' = (t^2 - y^2) \sin y, \quad y(0) = -1$

In each of Problems 7 through 11, find approximate values of the solution of the initial value problem at $t = 0.5$, 1.0, 1.5, and 2.0.

N **a.** Use the improved Euler method with $h = 0.025$.
N **b.** Use the improved Euler method with $h = 0.0125$.

7. $y' = 0.5 - t + 2y, \quad y(0) = 1$
8. $y' = 5t - 3\sqrt{y}, \quad y(0) = 2$
9. $y' = \sqrt{t + y}, \quad y(0) = 3$
10. $y' = 2t + e^{-ty}, \quad y(0) = 1$
11. $y' = (y^2 + 2ty)/(3 + t^2), \quad y(0) = 0.5$

12. In this problem we establish that the local truncation error for the improved Euler formula is proportional to h^3. If we assume that the solution ϕ of the initial value problem $y' = f(t, y)$, $y(t_0) = y_0$ has derivatives that are continuous through the third order (f has continuous second partial derivatives), then it follows that

$$\phi(t_n + h) = \phi(t_n) + \phi'(t_n)h + \frac{\phi''(t_n)}{2!}h^2 + \frac{\phi'''(\overline{t_n})}{3!}h^3,$$

where $t_n < \overline{t_n} < t_n + h$. Assume that $y_n = \phi(t_n)$.

a. Show that, for y_{n+1} as given by equation (5),

$$e_{n+1} = \phi(t_{n+1}) - y_{n+1} = \frac{\phi''(t_n)h - (f(t_n + h, y_n + hf(t_n, y_n)) - f(t_n, y_n))}{2!}h + \frac{\phi'''(\overline{t_n})h^3}{3!}. \tag{10}$$

b. Use the facts that $\phi''(t) = f_t(t, \phi(t)) + f_y(t, \phi(t))\phi'(t)$ and that the Taylor approximation with a remainder for a function $F(t, y)$ of two variables is

$$F(a + h, b + k) = F(a, b) + F_t(a, b)h + F_y(a, b)k + \frac{1}{2!}(h^2 F_{tt} + 2hkF_{ty} + k^2 F_{yy})\Big|_{t=\xi, y=\eta},$$

where ξ lies between a and $a + h$, and η lies between b and $b + k$, to show that the first term on the right-hand side of equation (10) is proportional to h^3 plus higher-order terms. This is the critical estimate needed to prove that the local truncation error is proportional to h^3.

c. Show that if $f(t, y)$ is linear in t and y, then $e_{n+1} = \frac{1}{6}\phi'''(\overline{t_n})h^3$ for some $\overline{t_n}$ with $t_n < \overline{t_n} < t_{n+1}$.

Hint: What are f_{tt}, f_{ty}, and f_{yy}?

13. Consider the improved Euler method for solving the illustrative initial value problem $y' = 1 - t + 4y$, $y(0) = 1$.

a. Using the result of Problem 12c and the exact solution of the initial value problem, determine e_{n+1} and a bound for the error at any step on $0 \le t \le 2$.
b. Compare the error found in a with the one obtained in equation (27) of Section 8.1 using the Euler method.
c. Also obtain a bound for e_1 for $h = 0.05$, and compare it with equation (28) of Section 8.1.

In each of Problems 14 and 15,

a. Use the actual solution $\phi(t)$ to determine e_{n+1} and a bound for e_{n+1} at any step on $0 \le t \le 1$ for the improved Euler method for the given initial value problem.
b. Also obtain a bound for e_1 for $h = 0.1$, and compare it with the similar estimate for the Euler method and with the actual error for the improved Euler method.

14. $y' = 2y - 1, \quad y(0) = 1$
15. $y' = 0.5 - t + 2y, \quad y(0) = 1$

In each of Problems 16 through 19, carry out one step of the Euler method and of the improved Euler method, using the step size $h = 0.1$. Suppose that a local truncation error no greater than $\epsilon = 0.0025$ is required. Estimate the step size that is needed for the Euler method to satisfy this requirement at the first step.

N **16.** $y' = 0.5 - t + 2y, \quad y(0) = 1$
N **17.** $y' = 5t - 3\sqrt{y}, \quad y(0) = 2$
N **18.** $y' = \sqrt{t + y}, \quad y(0) = 3$
N **19.** $y' = (y^2 + 2ty)/(3 + t^2), \quad y(0) = 0.5$

20. The **modified Euler formula** for the initial value problem $y' = f(t, y)$, $y(t_0) = y_0$ is given by

$$y_{n+1} = y_n + hf\left(t_n + \frac{1}{2}h, y_n + \frac{1}{2}hf(t_n, y_n)\right).$$

Following the procedure outlined in Problem 12, show that the local truncation error in the modified Euler formula is proportional to h^3.

In each of Problems 21 through 24, use the modified Euler formula of Problem 20 with $h = 0.05$ to compute approximate values of the solution of the given initial value problem at $t = 0.1$, 0.2, 0.3, and 0.4.

N **21.** $y' = 3 + t - y, \quad y(0) = 1$ (Compare with Problem 2)
N **22.** $y' = 5t - 3\sqrt{y}, \quad y(0) = 2$
N **23.** $y' = 2y - 3t, \quad y(0) = 1$ (Compare with Problem 3)
N **24.** $y' = 2t + e^{-ty}, \quad y(0) = 1$ (Compare with Problem 4)

25. Show that the modified Euler formula of Problem 20 is identical to the improved Euler formula of equation (5) for $y' = f(t, y)$ if f is linear in both t and y.

8.3 The Runge-Kutta Method

The Euler formula, the backward Euler formula, and the improved Euler formula were introduced, in Sections 8.1 and 8.2, as ways to numerically approximate the solution of the initial value problem

$$y' = f(t, y), \quad y(t_0) = y_0. \tag{1}$$

The local truncation errors for these methods are proportional to h^2, h^2, and h^3, respectively. The Euler and improved Euler methods belong to what is now called the Runge-Kutta[2] class of methods.

In this section we discuss the method originally developed by Runge and Kutta. This method is now called the classic **fourth-order four-stage Runge-Kutta method**, but it is often referred to simply as *the* Runge-Kutta method, and we will follow this practice for

[2]Carl David Runge (1856–1927), a German mathematician and physicist, worked for many years in spectroscopy. The analysis of data led him to consider problems in numerical computation, and the Runge-Kutta method originated in his paper on the numerical solution of differential equations in 1895. The method was extended to systems of equations in 1901 by Martin Wilhelm Kutta (1867–1944). Kutta was a German mathematician and aerodynamicist who is also well known for his important contributions to classical airfoil theory.

This has stimulated the development of adaptive Runge-Kutta methods that provide for modifying the step size automatically as the computation proceeds, so as to maintain the local truncation error near or below a specified tolerance level. As explained in Section 8.2, this requires the estimation of the local truncation error at each step. One way to do this is to repeat the computation with a fifth-order method—which has a local truncation error proportional to h^6—and then to use the difference between the two results as an estimate of the error. If this is done in a straightforward manner, then the use of the fifth-order method requires at least five more evaluations of f at each step, in addition to those required originally by the fourth-order method. However, if we make an appropriate choice of the intermediate points and the weighting coefficients in the expressions for $k_{n1}, \ldots, k_{n4}$ in a certain fourth-order Runge-Kutta method, then these expressions can be used again, together with one additional stage, in a corresponding fifth-order method. This results in a substantial gain in efficiency. It turns out that this can be done in more than one way.

The first fourth- and fifth-order Runge-Kutta pair was developed by Erwin Fehlberg[5] in the late 1960s and is now called the Runge-Kutta-Fehlberg, or RKF,[6] method. The popularity of the RKF method was considerably enhanced by the appearance in 1977 of its Fortran implementation RKF45 by Lawrence F. Shampine and H. A. Watts. The RKF method and other adaptive Runge-Kutta methods are very powerful and efficient means of approximating numerically the solutions of an enormous class of initial value problems. Specific implementations of one or more of them are widely available in commercial software packages.

Problems

N **1.** Confirm the results in Table 8.3.1 by executing the indicated computations.

In each of Problems 2 through 6, find approximate values of the solution of the given initial value problem at $t = 0.1$, 0.2, 0.3, and 0.4. Compare the results with those obtained by using other methods and with the exact solution (if available).

N **a.** Use the Runge-Kutta method with $h = 0.1$.

N **b.** Use the Runge-Kutta method with $h = 0.05$.

2. $y' = 3 + t - y, \quad y(0) = 1$

3. $y' = 5t - 3\sqrt{y}, \quad y(0) = 2$

4. $y' = 2t + e^{-ty}, \quad y(0) = 1$

5. $y' = (y^2 + 2ty)/(3 + t^2), \quad y(0) = 0.5$

6. $y' = (t^2 - y^2)\sin y, \quad y(0) = -1$

In each of Problems 7 through 11, find approximate values of the solution of the given initial value problem at $t = 0.5$, 1.0, 1.5, and 2.0. Compare the results with those obtained by other methods and with the exact solution (if available).

N **a.** Use the Runge-Kutta method with $h = 0.1$.

N **b.** Use the Runge-Kutta method with $h = 0.05$.

7. $y' = 0.5 - t + 2y, \quad y(0) = 1$

8. $y' = 5t - 3\sqrt{y}, \quad y(0) = 2$

9. $y' = \sqrt{t + y}, \quad y(0) = 3$

10. $y' = 2t + e^{-ty}, \quad y(0) = 1$

11. $y' = (y^2 + 2ty)/(3 + t^2), \quad y(0) = 0.5$

12. Consider the initial value problem

$$y' = 3t^2/(3y^2 - 4), \quad y(0) = 0.$$

Let t_M be the right-hand endpoint of the interval of existence of this solution.

G **a.** Draw a direction field for this equation.

b. Use the direction field created in a to estimate t_M. What happens at t_M to prevent the solution from continuing farther?

N **c.** Use the Runge-Kutta method with various step sizes to determine an approximate value of t_M.

d. If you continue the Runge-Kutta computation for $t > t_m$, you can continue to generate values of y. What significance, if any, do these values have?

N **e.** Suppose that the initial condition is changed to $y(0) = 1$. Repeat parts b and c for this problem.

[5]Erwin Fehlberg (1911–1990) was born in Germany, received his doctorate from the Technical University of Berlin in 1942, emigrated to the United States after World War II, and was employed by NASA for many years. The Runge-Kutta-Fehlberg method was first published in a NASA Technical Report in 1969.

[6]The details of the RKF method may be found, for example, in the books by Ascher and Petzold and by Mattheij and Molenaar that are listed in the References.

8.4 Multistep Methods

In previous sections we have discussed numerical procedures for approximating the solution of the initial value problem

$$y' = f(t, y), \quad y(t_0) = y_0, \tag{1}$$

in which data at the point $t = t_n$ are used to calculate an approximate value of the solution $\phi(t_{n+1})$ at the next mesh point $t = t_{n+1}$. In other words, the calculated value of the exact solution ϕ at any mesh point depends only on the data at the preceding mesh point. Such methods are called **one-step methods**. However, once approximate values of the exact solution $y = \phi(t)$ have been obtained at a few points beyond t_0, it is natural to ask whether we can make use of more of this information—not just the value at the last point—to calculate the value of $\phi(t)$ at the next point. Specifically, if y_1 at t_1, y_2 at t_2, $\ldots$, y_n at t_n are known, how can we use this information to determine y_{n+1} at t_{n+1}? Methods that use information at more than the last mesh point are referred to as **multistep methods**. In this section we will describe two types of multistep methods: Adams[7] methods and backward differentiation methods. Within each type, we can achieve various levels of accuracy, depending on the number of preceding data points that are used. For simplicity, we will assume throughout our discussion that the step size h is constant.

Adams Methods. Recall that the solution $\phi(t)$ of the initial value problem (1) satisfies

$$\phi(t_{n+1}) - \phi(t_n) = \int_{t_n}^{t_{n+1}} \phi'(t)\, dt. \tag{2}$$

The basic idea of an Adams method is to approximate $\phi'(t)$ by a polynomial $P_k(t)$ of degree k and to use the polynomial to evaluate the integral on the right-hand side of equation (2). The coefficients in $P_k(t)$ are determined by using $k + 1$ previously calculated data points.

For example, suppose that we wish to use a first-degree polynomial $P_1(t) = At + B$. Then we need only the two data points (t_n, y_n) and (t_{n-1}, y_{n-1}). For P_1 to interpolate ϕ' at both $t = t_n$ and $t = t_{n-1}$, we require both that $P_1(t_n) = \phi'(t_n) = f(t_n, y_n)$ and that $P_1(t_{n-1}) = \phi'(t_{n-1}) = f(t_{n-1}, y_{n-1})$. Recall that we denote $f(t_j, y_j)$ by f_j for an integer j. Thus A and B must satisfy the equations

$$\begin{aligned} At_n + B &= f_n, \\ At_{n-1} + B &= f_{n-1}. \end{aligned} \tag{3}$$

Solving for A and B, we obtain

$$A = \frac{f_n - f_{n-1}}{h} \quad \text{and} \quad B = \frac{f_{n-1}t_n - f_n t_{n-1}}{h}. \tag{4}$$

Replacing $\phi'(t)$ by $P_1(t)$ and evaluating the integral in equation (2), we find that

$$\phi(t_{n+1}) - \phi(t_n) = \int_{t_{n+1}}^{t_n} (At + B)\, dt = \frac{A}{2}\left(t_{n+1}^2 - t_n^2\right) + B(t_{n+1} - t_n).$$

Finally, we replace $\phi(t_{n+1})$ and $\phi(t_n)$ by y_{n+1} and y_n, respectively, and carry out some algebraic simplification. For a constant step size h, we obtain

$$y_{n+1} = y_n + \frac{3}{2}hf_n - \frac{1}{2}hf_{n-1}. \tag{5}$$

Equation (5) is the **second-order Adams-Bashforth[8] formula**. It is an explicit formula for y_{n+1} in terms of y_n and y_{n-1} and has a local truncation error proportional to h^3.

[7]John Couch Adams (1819–1892), an English mathematician and astronomer, is most famous as codiscoverer, with Joseph Leverrier, of the planet Neptune in 1846. He was associated with Cambridge University for most of his life, as student (1839–1843), fellow, Lowdean Professor, and director of the Observatory. Adams was extremely skilled at computation; his procedure for numerical integration of differential equations appeared in 1883 in a book on capillary action written with Francis Bashforth.

[8]Francis Bashforth (1819–1912), English mathematician and Anglican priest, was a classmate of J. C. Adams at Cambridge. He was particularly interested in ballistics and invented the Bashforth chronograph for measuring the velocity of artillery projectiles.

The Runge-Kutta method with $h = 0.1$ gives $y_4 = 5.7927853$ with an error of -0.0014407; see Table 8.3.1. Thus, for this problem, the Runge-Kutta method is comparable in accuracy to the predictor–corrector method.

Backward Differentiation Formulas. Another type of multistep method uses a polynomial $P_k(t)$ to approximate the solution $\phi(t)$ of the initial value problem (1) rather than its derivative $\phi'(t)$, as in the Adams methods. We then differentiate $P_k(t)$ and set $P_k'(t_{n+1})$ equal to $f(t_{n+1}, y_{n+1})$ to obtain an implicit formula for y_{n+1}. These are called **backward differentiation formulas**. These methods became widely used in the 1970s because of the work of C. William Gear[10] on so-called *stiff differential equations*, whose solutions are very difficult to approximate by the methods discussed up to now; see Section 8.6.

The simplest case uses a first-degree polynomial $P_1(t) = At + B$. The coefficients are chosen so that $P_1(t_n)$ and $P_1(t_{n+1})$ agree with the computed values of the solution y_n and y_{n+1}, respectively: $P_1(t_n) = y_n$ and $P_1(t_{n+1}) = y_{n+1}$. Thus A and B must satisfy

$$\begin{aligned} At_n + B &= y_n, \\ At_{n+1} + B &= y_{n+1}. \end{aligned} \tag{12}$$

Solving the linear algebraic equations (12) for A and B yields

$$A = \frac{y_{n+1} - y_n}{h} \quad \text{and} \quad B = \frac{y_{n+1}t_n - y_n t_{n+1}}{h}. \tag{13}$$

Since $P_1'(t) = A$, the requirement that $P_1'(t_{n+1}) = f(t_{n+1}, y_{n+1})$ is just $A = f(t_{n+1}, y_{n+1})$. Equating this value of A and the value of A given in equation (13) and rearranging terms, we obtain the **first-order backward differentiation formula**

$$y_{n+1} = y_n + hf(t_{n+1}, y_{n+1}). \tag{14}$$

Note that equation (14) is just the backward Euler formula that we first saw in Section 8.1.

By using higher-order polynomials and correspondingly more data points, we can obtain backward differentiation formulas of any order. The **second-order backward differentiation formula** is

$$y_{n+1} = \frac{1}{3}\left(4y_n - y_{n-1} + 2hf(t_{n+1}, y_{n+1})\right), \tag{15}$$

and the **fourth-order backward differentiation formula** is

$$y_{n+1} = \frac{1}{25}\left(48y_n - 36y_{n-1} + 16y_{n-2} - 3y_{n-3} + 12hf(t_{n+1}, y_{n+1})\right). \tag{16}$$

These formulas have local truncation errors proportional to h^3 and h^5, respectively.

EXAMPLE 2

Use the fourth-order backward differentiation formula with $h = 0.1$ and the data given in Example 1 to determine an approximate value of the solution $y = \phi(t)$ at $t = 0.4$ for the initial value problem (11).

Solution:

Using equation (16) with $n = 3$, $h = 0.1$, and with $y_0, \ldots, y_3$ given in Example 1, we obtain the equation

$$y_4 = 4.6837842 + 0.192y_4.$$

[10] C. William Gear (1935–), born in London, England, received his undergraduate education at Cambridge University and his doctorate in 1960 from the University of Illinois. He was a member of the faculty at the University of Illinois for most of his career and made significant contributions to both computer design and numerical analysis. His influential book on numerical methods for differential equations is listed in the References.

Thus

$$y_4 = 5.7967626.$$

Comparing the calculated value with the exact value $\phi(0.4) = 5.7942260$, we find that the error is 0.0025366. This is somewhat better than the result using the fourth-order Adams-Bashforth method, but it is not as good as the result using the fourth-order predictor–corrector method, and not nearly as good as the result using the fourth-order Adams-Moulton method.

A comparison between one-step and multistep methods must take several factors into consideration. The fourth-order Runge-Kutta method requires four evaluations of f at each step, while the fourth-order Adams-Bashforth method (once past the starting values) requires only one, and the predictor–corrector method only two. Thus, for a given step size h, the latter two methods may well be considerably faster than Runge-Kutta. However, if Runge-Kutta is more accurate and therefore can use fewer steps, the difference in speed will be reduced and perhaps eliminated.

The Adams-Moulton and backward differentiation formulas also require that the difficulty in solving the implicit equation at each step be taken into account. All multistep methods have the possible disadvantage that errors in earlier steps can feed back into later calculations with unfavorable consequences. On the other hand, the underlying polynomial approximations in multistep methods make it easy to approximate the solution at points between the mesh points, should this be desirable. Multistep methods have become popular largely because it is relatively easy to estimate the error at each step and to adjust the order or the step size to control it. For a further discussion of such questions as these, see the books listed at the end of this chapter; in particular, Shampine (1994) continues to be an authoritative source.

Problems

In each of Problems 1 through 5, determine an approximate value of the solution at $t = 0.4$ and $t = 0.5$ using the specified method. For starting values, use the values given by the Runge-Kutta method; see Problems 2 through 6 of Section 8.3. Compare the results of the various methods with each other and with the actual solution (if available).

N **a.** Use the fourth-order predictor–corrector method with $h = 0.1$. Use the corrector formula once at each step.

N **b.** Use the fourth-order Adams-Moulton method with $h = 0.1$.

N **c.** Use the fourth-order backward differentiation method with $h = 0.1$.

1. $y' = 3 + t - y, \quad y(0) = 1$
2. $y' = 5t - 3\sqrt{y}, \quad y(0) = 2$
3. $y' = 2t + e^{-ty}, \quad y(0) = 1$
4. $y' = (y^2 + 2ty)/(3 + t^2), \quad y(0) = 0.5$
5. $y' = (t^2 - y^2)\sin y, \quad y(0) = -1$

In each of Problems 6 through 10, find approximate values of the solution of the given initial value problem at $t = 0.5, 1.0, 1.5$, and 2.0, using the specified method. For starting values, use the values given by the Runge-Kutta method; see Problems 7 through 11 in Section 8.3. Compare the results of the various methods with each other and with the actual solution (if available).

N **a.** Use the fourth-order predictor–corrector method with $h = 0.05$. Use the corrector formula once at each step.

N **b.** Use the fourth-order Adams-Moulton method with $h = 0.05$.

N **c.** Use the fourth-order backward differentiation method with $h = 0.05$.

6. $y' = 0.5 - t + 2y, \quad y(0) = 1$
7. $y' = 5t - 3\sqrt{y}, \quad y(0) = 2$
8. $y' = \sqrt{t + y}, \quad y(0) = 3$
9. $y' = 2t + e^{-ty}, \quad y(0) = 1$
10. $y' = (y^2 + 2ty)/(3 + t^2), \quad y(0) = 0.5$
11. **a.** Show that the first-order Adams-Bashforth method is the Euler method.
    b. Show that the first-order Adams-Moulton method is the backward Euler method.
12. Show that the third-order Adams-Bashforth formula is
$$y_{n+1} = y_n + \frac{h}{12}\left(23f_n - 16f_{n-1} + 5f_{n-2}\right).$$
13. Show that the third-order Adams-Moulton formula is
$$y_{n+1} = y_n + \frac{h}{12}\left(5f_{n+1} + 8f_n - f_{n-1}\right).$$
14. Derive the second-order backward differentiation formula given by equation (15) in this section.

Problems

In each of Problems 1 through 5, determine approximate values of the solution $x = \phi(t)$, $y = \psi(t)$ of the given initial value problem at $t = 0.2, 0.4, 0.6, 0.8$, and 1.0. Compare the results obtained by different methods and different step sizes.

(N) **a.** Use the Euler method with $h = 0.1$.
(N) **b.** Use the Runge-Kutta method with $h = 0.2$.
(N) **c.** Use the Runge-Kutta method with $h = 0.1$.

1. $x' = x + y + t$, $y' = 4x - 2y$; $x(0) = 1$, $y(0) = 0$

2. $x' = -tx - y - 1$, $y' = x$; $x(0) = 1$, $y(0) = 1$

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

4. $x' = x(1 - 0.5x - 0.5y)$, $y' = y(-0.25 + 0.5x)$; $x(0) = 4$, $y(0) = 1$

5. $x' = \exp(-x + y) - \cos x$, $y' = \sin(x - 3y)$; $x(0) = 1$, $y(0) = 2$

(N) **6.** Consider the example problem $x' = x - 4y$, $y' = -x + y$ with the initial conditions $x(0) = 1$ and $y(0) = 0$. Use the Runge-Kutta method to find approximate values of the solution of this problem on the interval $0 \le t \le 1$. Start with $h = 0.2$, and then repeat the calculation with step sizes $h = 0.1, 0.05, \ldots$, each half as long as in the preceding case. Continue the process until the first five digits of the solution at $t = 1$ are unchanged for successive step sizes. Determine whether these digits are accurate by comparing them with the exact solution given in equations (8) in the text.

(N) **7.** Consider the initial value problem

$$x'' + t^2x' + 3x = t, \quad x(0) = 1, \quad x'(0) = 2.$$

Convert this problem to a system of two first-order equations, and determine approximate values of the solution at $t = 0.5$ and $t = 1.0$ using the Runge-Kutta method with $h = 0.1$.

(N) **8.** Consider the general initial value problem $x' = f(t, x, y)$ and $y' = g(t, x, y)$ with $x(t_0) = x_0$ and $y(t_0) = y_0$. The Adams-Moulton predictor-corrector method of Section 8.4 generalizes to

$$x_{n+1} = x_n + \frac{1}{24}h(55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3}),$$

$$y_{n+1} = y_n + \frac{1}{24}h(55g_n - 59g_{n-1} + 37g_{n-2} - 9g_{n-3})$$

and

$$x_{n+1} = x_n + \frac{1}{24}h(9f_{n+1} + 19f_n - 5f_{n-1} + f_{n-2}),$$

$$y_{n+1} = y_n + \frac{1}{24}h(9g_{n+1} + 19g_n - 5g_{n-1} + g_{n-2}).$$

Determine an approximate value of the solution at $t = 0.4$ for the example initial value problem $x' = x - 4y$, $y' = -x + y$ with $x(0) = 1$, $y(0) = 0$. Take $h = 0.1$. Correct the predicted value once. For the values of $x_1, \ldots, y_3$ use the values of the exact solution rounded to six digits: $x_1 = 1.12735$, $x_2 = 1.32042$, $x_3 = 1.60021$, $y_1 = -0.111255$, $y_2 = -0.250847$, and $y_3 = -0.429696$.

8.6 More on Errors; Stability

In Section 8.1 we discussed some ideas related to the errors that can occur in a numerical approximation of the solution of the initial value problem

$$y' = f(t, y), \quad y(t_0) = y_0. \tag{1}$$

In this section we continue that discussion and also point out some other difficulties that can arise. Some of the points that we wish to make are fairly difficult to treat in detail, so we will illustrate them by means of examples.

Truncation and Round-Off Errors. Recall that, for the Euler method with equal time steps of size h, we showed the local truncation error is proportional to h^2 and, for a finite interval, the global truncation error is at most a constant times h. In general, for a method of order p, the local truncation error is proportional to h^{p+1} and the global truncation error on a finite interval is bounded by a constant times h^p. For example, Euler's method is an order 1 method.

To achieve high accuracy, we normally use a numerical procedure for which p is fairly large, perhaps 4 or higher. As p increases, the formula used in computing y_{n+1} normally becomes more complicated, and hence more calculations are required at each step. However, this is usually not a serious problem unless $f(t, y)$ is very complicated or the calculation must be repeated very many times.

If the step size h is decreased, the global truncation error is decreased by the same factor raised to the power p. However, as we mentioned in Section 8.1, if h is very small, a great many steps will be required to cover a fixed interval, and the global round-off error may be larger than the global truncation error. The situation is shown schematically in Figure 8.6.1. We assume that the round-off error R_n is proportional to the number of computations performed

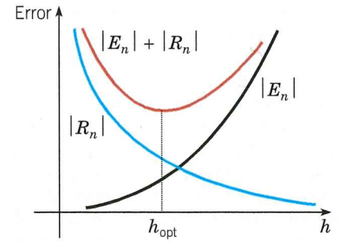

FIGURE 8.6.1 The dependence of truncation error $|E_n|$ (black), round-off error $|R_n|$ (blue), and total error $|E_n| + |R_n|$ (red) on the step size h.

and therefore is inversely proportional to the step size h. On the other hand, the truncation error E_n is proportional to a positive power of h. From equation (17) of Section 8.1, we know that the total error is bounded by $|E_n| + |R_n|$; hence we wish to choose h so as to minimize this quantity. The optimum value of h occurs when the rate of increase of the truncation error (as h increases) is balanced by the rate of decrease of the round-off error, as indicated in Figure 8.6.1.

EXAMPLE 1

Consider the example problem

$$y' = 1 - t + 4y, \quad y(0) = 1. \tag{2}$$

Using the Euler method with various step sizes, calculate approximate values for the solution $\phi(t)$ at $t = 0.5$ and $t = 1$. Try to determine the optimum step size.

Solution:

Table 8.6.1 shows the results of applying Euler's method for nine different values of h. The results were obtained using software configured to use only four significant digits. This was done on purpose to have round-off errors become significant for larger values of h than if more significant digits are used in floating-point operations. The first two columns are the step size h and the number of steps N required to traverse the interval $0 \le t \le 1$. Then $y_{N/2}$ and y_N are approximations to $\phi(0.5) = 8.712$ and $\phi(1) = 64.90$, respectively. These quantities appear in the third and fifth columns. The fourth and sixth columns display the differences between the calculated values and the actual value of the solution.

TABLE 8.6.1 Approximations to the Solution of the Initial Value Problem $y' = 1 - t + 4y, y(0) = 1$ Using the Euler Method with Different Step Sizes

h	N	$y_{N/2}$	Error	y_N	Error
0.01	100	8.390	−0.322	60.12	−4.78
0.005	200	8.551	−0.161	62.51	−2.39
0.002	500	8.633	−0.079	63.75	−1.15
0.001	1000	8.656	−0.056	63.94	−0.96
0.0008	1250	8.636	−0.076	63.78	−1.12
0.000625	1600	8.616	−0.096	64.35	−0.55
0.0005	2000	8.772	0.060	64.00	−0.90
0.0004	2500	8.507	0.205	63.40	−1.50
0.00025	4000	8.231	0.481	56.77	−8.13

You should bear in mind that the numerical values of the entries in the second column of Table 8.6.4 are extremely sensitive to slight variations in how the calculations are executed. Regardless of such details, however, the exponential growth of the approximation will be clearly evident.

Equation (18) is highly unstable, and the behavior shown in this example is typical of unstable problems. We can track a solution accurately for a while, and the interval can be extended by using smaller step sizes or more accurate methods, but eventually the instability in the problem itself takes over and leads to large errors.

Some Comments on the Selection of a Numerical Method. In this chapter we have introduced several numerical methods for approximating the solution of an initial value problem. We have tried to emphasize some important ideas while limiting the level of complexity. For one thing, except for the comments at the end of Section 8.2, we have always used a uniform step size, whereas production codes that are currently in use provide for varying the step size as the calculation proceeds.

There are several considerations that must be taken into account in choosing step sizes. Of course, one is accuracy; too large a step size leads to an inaccurate result. Normally, an error tolerance is prescribed in advance, and the step size at each step must be consistent with this requirement. As we have seen, the step size must also be chosen so that the method is stable. Otherwise, small errors could grow and render the subsequent computations worthless. Finally, for implicit methods an equation must be solved at each step, and the method used to solve the equation may impose additional restrictions on the step size.

In choosing a method, one must also balance the considerations of accuracy and stability against the amount of time required to execute each step. An implicit method, such as the Adams-Moulton method, requires more calculations for each step, but if its accuracy and stability permit a larger step size (and consequently fewer steps), then this may more than compensate for the additional calculations. The backward differentiation formulas of moderate order (say, four) are highly stable and are therefore indicated for stiff problems, for which stability is the controlling factor.

Some current production codes also permit the order of the method to be varied, as well as the step size, as the calculation proceeds. The error is estimated at each step, and the order and step size are chosen to satisfy the prescribed error tolerance. In practice, Adams methods up to order twelve and backward differentiation formulas up to order five are in use. Higher-order backward differentiation formulas are unsuitable because of a lack of stability.

Finally, we note that the smoothness of the function f—that is, the number of continuous derivatives that it possesses—is a factor in choosing the order of the method to be used. High-order methods lose some of their accuracy if f is not smooth to a corresponding order.

A numerical analysis course is likely to provide a more in-depth investigation of errors, stability, and efficiency. Similar information can be found in the References at the end of this chapter.

Problems

1. To obtain some idea of the possible dangers of small errors in the initial conditions, such as those due to round-off, consider the initial value problem

$$y' = t + y - 3, \quad y(0) = 2.$$

a. Show that the solution is $y = \phi_1(t) = 2 - t$.

b. Suppose that in the initial condition a mistake is made, and 2.001 is used instead of 2. Determine the solution $y = \phi_2(t)$ in this case, and compare the difference $\phi_2(t) - \phi_1(t)$ at $t = 1$ and as $t \to \infty$.

2. Consider the initial value problem

$$y' = t^2 + e^y, \quad y(0) = 0. \tag{26}$$

Using the Runge-Kutta method with step size h, we obtain the results in Table 8.6.5. These results suggest that the solution has a vertical asymptote between $t = 0.9$ and $t = 1.0$.

TABLE 8.6.5 Approximations to the Solution of the Initial Value Problem $y' = t^2 + e^y$, $y(0) = 0$ Using the Runge-Kutta Method

h	$y(0.9)$	$y(1.0)$
0.02	3.42985	$> 10^{38}$
0.01	3.42982	$> 10^{38}$

a. Let $y = \phi(t)$ be the solution of initial value problem (27). Further, let $y = \phi_1(t)$ be the solution of

$$y' = 1 + e^y, \quad y(0) = 0, \tag{27}$$

and let $y = \phi_2(t)$ be the solution of

$$y' = e^y, \quad y(0) = 0. \tag{28}$$

Show that

$$\phi_2(t) \le \phi(t) \le \phi_1(t) \tag{29}$$

on some interval, contained in $0 \le t \le 1$, where all three solutions exist.

b. Determine $\phi_1(t)$ and $\phi_2(t)$. Then show that $\phi(t) \to \infty$ for some t between $t = \ln 2 \cong 0.69315$ and $t = 1$.

c. Solve the differential equations $y' = e^y$ and $y' = 1 + e^y$, respectively, with the initial condition $y(0.9) = 3.4298$. Use the results to show that $\phi(t) \to \infty$ when $t \cong 0.932$.

3. Consider again the initial value problem (16) from Example 2. Investigate how small a step size h must be chosen to ensure that the error at $t = 0.05$ and at $t = 0.1$ is less than 0.0005.

N **a.** Use the Euler method.

N **b.** Use the backward Euler method.

N **c.** Use the Runge-Kutta method.

4. Consider the initial value problem

$$y' = -10y + 2.5t^2 + 0.5t, \quad y(0) = 4.$$

a. Find the solution $y = \phi(t)$ and draw its graph for $0 \le t \le 5$.

N **b.** The stability analysis in the text suggests that for this problem, the Euler method is stable only for $h < 0.2$. Confirm that this is true by applying the Euler method to this problem for $0 \le t \le 5$ with step sizes near 0.2.

N **c.** Apply the Runge-Kutta method with various step sizes to this problem for $0 \le t \le 5$. What can you conclude about the stability of this method?

N **d.** Apply the backward Euler method with various step sizes to this problem for $0 \le t \le 5$. What step size is needed to ensure that the error at $t = 5$ is less than 0.01?

In each of Problems 5 and 6:

a. Find a formula for the solution of the initial value problem, and note that it is independent of λ.

N **b.** Use the Runge-Kutta method with $h = 0.01$ to compute approximate values of the solution for $0 \le t \le 1$ for various values of λ such as $\lambda = 1, 10, 20$, and 50.

c. Explain the differences, if any, between the exact solution and the numerical approximations.

5. $y' - \lambda y = 1 - \lambda t, \quad y(0) = 0$

6. $y' - \lambda y = 2t - \lambda t^2, \quad y(0) = 0$

References

There are many books of varying degrees of sophistication that deal with numerical analysis in general and the numerical approximation of solutions of ordinary differential equations in particular. Among these are:

Ascher, Uri M., and Petzold, Linda R., *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations* (Philadelphia: Society for Industrial and Applied Mathematics, 1998).

Atkinson, Kendall E., Han, Weimin, and Stewart, David, *Numerical Solution of Ordinary Differential Equations* (Hoboken, NJ: Wiley, 2009).

Gautschi, W. *Numerical Analysis* (2nd ed.) (New York: Birkhäuser, 2011).

Gear, C. William, *Numerical Initial Value Problems in Ordinary Differential Equations* (Englewood Cliffs, NJ: Prentice-Hall, 1971).

Henrici, Peter, *Discrete Variable Methods in Ordinary Differential Equations* (New York: Wiley, 1962).

Henrici, Peter, *Error Propagation for Difference Methods* (New York: Wiley, 1963; Huntington, NY: Krieger, 1977).

Iserles, A., *A First Course in Numerical Analysis of Differential Equations* (New York: Cambridge University Press, 2009).

Mattheij, Robert, and Molenaar, Jaap, *Ordinary Differential Equations in Theory and Practice* (New York: Wiley, 1996; Philadelphia: Society for Industrial and Applied Mathematics, 2002).

Shampine, Lawrence F., *Numerical Solution of Ordinary Differential Equations* (New York: Chapman and Hall, 1994).

A detailed exposition of Adams predictor–corrector methods, including practical guidelines for implementation, may be found in

Shampine, L. F., and Gordon, M. K., *Computer Solution of Ordinary Differential Equations: The Initial Value Problem* (San Francisco: Freeman, 1975).

Many books on numerical analysis have chapters on differential equations. For example, at an elementary level, see

Burden, Richard L., and Faires, J. Douglas, *Numerical Analysis* (9th ed.) (Boston: Brooks/Cole, Cengage Learning, 2011).

The following three books are at a slightly higher level and include information on implementing the algorithms in MATLAB.

Atkinson, Kendall E., and Han, Weimin, *Elementary Numerical Analysis* (3rd ed.) (Hoboken, NJ: Wiley, 2004).

Shampine, L. F., Gladwell, I., and Thompson, S., *Solving ODEs with MATLAB* (New York: Cambridge University Press, 2003).

Stoer, J., and Bulirsch, R., *Introduction to Numerical Analysis* (3rd ed.) (New York: Springer, 2002).