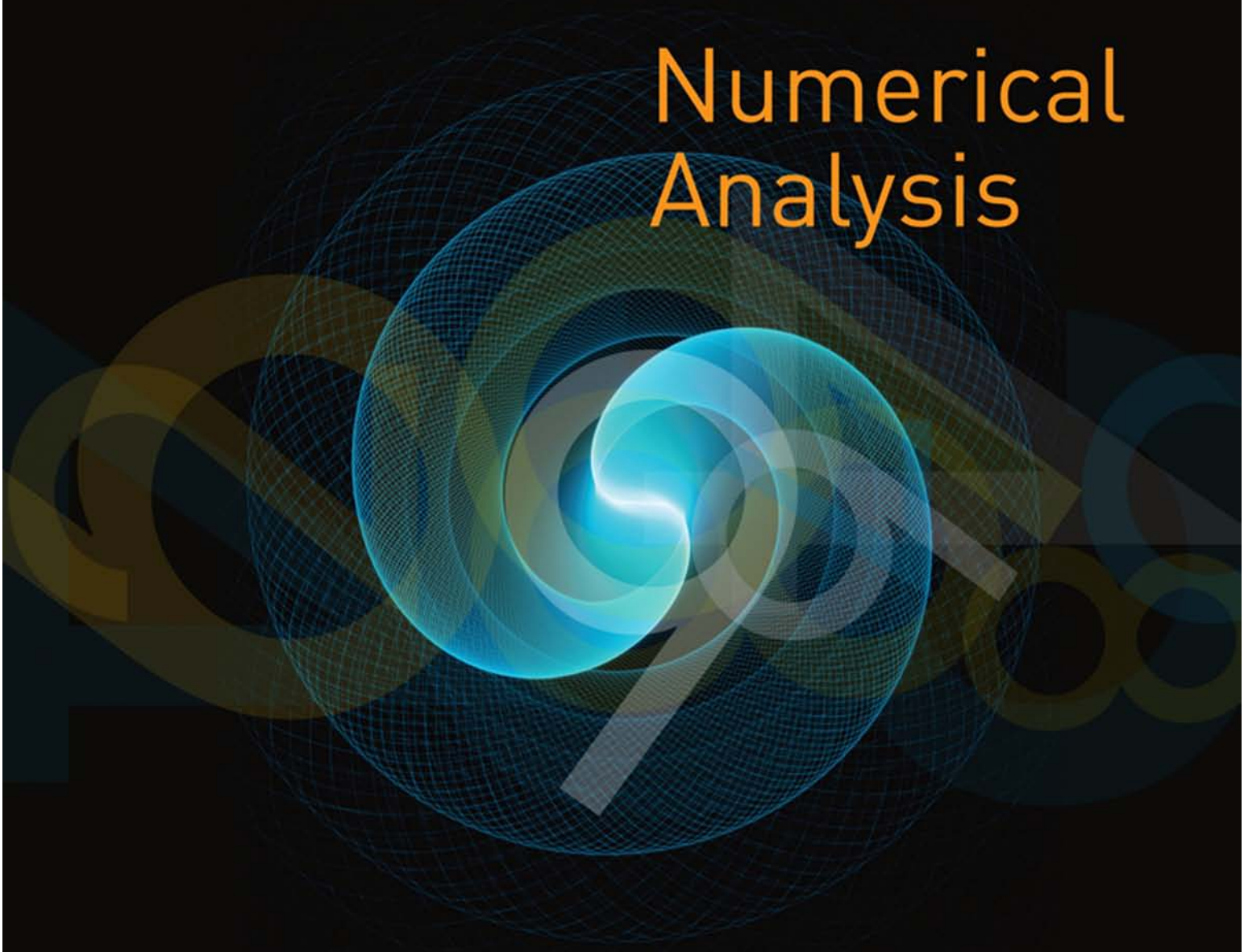
Richard L. Burden
J. Douglas Faires

Numerical Analysis



Ninth Edition

CHAPTER

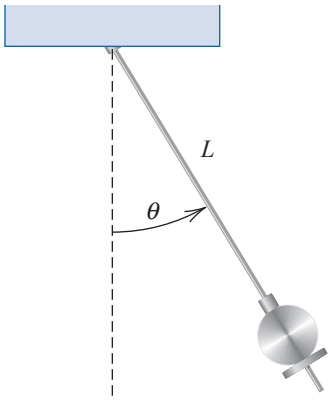
5

Initial-Value Problems for Ordinary Differential Equations

Introduction

The motion of a swinging pendulum under certain simplifying assumptions is described by the second-order differential equation

$$\frac{d^2\theta}{dt^2} + \frac{g}{L}\sin\theta = 0,$$



where L is the length of the pendulum, $g \approx 32.17$ ft/s^2 is the gravitational constant of the earth, and θ is the angle the pendulum makes with the vertical. If, in addition, we specify the position of the pendulum when the motion begins, $\theta(t_0) = \theta_0$, and its velocity at that point, $\theta'(t_0) = \theta_0'$, we have what is called an *initial-value problem*.

For small values of θ, the approximation $\theta \approx \sin\theta$ can be used to simplify this problem to the linear initial-value problem

$$\frac{d^2\theta}{dt^2} + \frac{g}{L}\theta = 0, \quad \theta(t_0) = \theta_0, \quad \theta'(t_0) = \theta_0'.$$

This problem can be solved by a standard differential-equation technique. For larger values of θ, the assumption that $\theta = \sin\theta$ is not reasonable so approximation methods must be used. A problem of this type is considered in Exercise 8 of Section 5.9.

Any textbook on ordinary differential equations details a number of methods for explicitly finding solutions to first-order initial-value problems. In practice, however, few of the problems originating from the study of physical phenomena can be solved exactly.

The first part of this chapter is concerned with approximating the solution $y(t)$ to a problem of the form

$$\frac{dy}{dt} = f(t, y), \quad \text{for } a \le t \le b,$$

subject to an initial condition $y(a) = \alpha$. Later in the chapter we deal with the extension of these methods to a system of first-order differential equations in the form

$$\begin{aligned}
\frac{dy_1}{dt} &= f_1(t, y_1, y_2, \ldots, y_n),\\
\frac{dy_2}{dt} &= f_2(t, y_1, y_2, \ldots, y_n),\\
&\vdots\\
\frac{dy_n}{dt} &= f_n(t, y_1, y_2, \ldots, y_n),
\end{aligned}$$

for $a \le t \le b$, subject to the initial conditions

$$y_1(a) = \alpha_1, \quad y_2(a) = \alpha_2, \quad \ldots, \quad y_n(a) = \alpha_n.$$

We also examine the relationship of a system of this type to the general nth-order initial-value problem of the form

$$y^{(n)} = f(t, y, y', y'', \ldots, y^{(n-1)}),$$

for $a \le t \le b$, subject to the initial conditions

$$y(a) = \alpha_1, \quad y'(a) = \alpha_2, \quad \ldots, \quad y^{n-1}(a) = \alpha_n.$$

5.1 The Elementary Theory of Initial-Value Problems

Differential equations are used to model problems in science and engineering that involve the change of some variable with respect to another. Most of these problems require the solution of an *initial-value problem*, that is, the solution to a differential equation that satisfies a given initial condition.

In common real-life situations, the differential equation that models the problem is too complicated to solve exactly, and one of two approaches is taken to approximate the solution. The first approach is to modify the problem by simplifying the differential equation to one that can be solved exactly and then use the solution of the simplified equation to approximate the solution to the original problem. The other approach, which we will examine in this chapter, uses methods for approximating the solution of the original problem. This is the approach that is most commonly taken because the approximation methods give more accurate results and realistic error information.

The methods that we consider in this chapter do not produce a continuous approximation to the solution of the initial-value problem. Rather, approximations are found at certain specified, and often equally spaced, points. Some method of interpolation, commonly Hermite, is used if intermediate values are needed.

We need some definitions and results from the theory of ordinary differential equations before considering methods for approximating the solutions to initial-value problems.

Definition 5.1 A function $f(t, y)$ is said to satisfy a **Lipschitz condition** in the variable y on a set $D \subset \mathbb{R}^2$ if a constant $L > 0$ exists with

$$|f(t, y_1) - f(t, y_2,)| \leq L|y_1 - y_2|,$$

whenever (t, y_1) and (t, y_2) are in D. The constant L is called a **Lipschitz constant** for f. ■

Example 1 Show that $f(t, y) = t|y|$ satisfies a Lipschitz condition on the interval $D = \{(t, y) \mid 1 \leq t \leq 2 \text{ and } -3 \leq y \leq 4\}$.

Solution For each pair of points (t, y_1) and (t, y_2) in D we have

$$|f(t, y_1) - f(t, y_2)| = |t|y_1| - t|y_2|| = |t||y_1| - |y_2|| \leq 2|y_1 - y_2|.$$

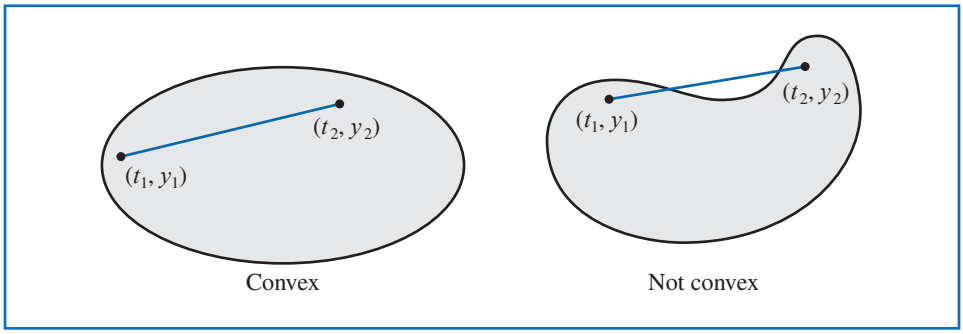
Thus f satisfies a Lipschitz condition on D in the variable y with Lipschitz constant 2. The smallest value possible for the Lipschitz constant for this problem is $L = 2$, because, for example,

$$|f(2, 1) - f(2, 0)| = |2 - 0| = 2|1 - 0|.$$ ■

Definition 5.2 A set $D \subset \mathbb{R}^2$ is said to be **convex** if whenever (t_1, y_1) and (t_2, y_2) belong to D, then $((1 - \lambda)t_1 + \lambda t_2, (1 - \lambda)y_1 + \lambda y_2)$ also belongs to D for every λ in $[0, 1]$. ■

In geometric terms, Definition 5.2 states that a set is convex provided that whenever two points belong to the set, the entire straight-line segment between the points also belongs to the set. (See Figure 5.1.) The sets we consider in this chapter are generally of the form $D = \{(t, y) \mid a \leq t \leq b \text{ and } -\infty < y < \infty\}$ for some constants a and b. It is easy to verify (see Exercise 7) that these sets are convex.

Figure 5.1



Theorem 5.3 Suppose $f(t, y)$ is defined on a convex set $D \subset \mathbb{R}^2$. If a constant $L > 0$ exists with

$$\left|\frac{\partial f}{\partial y}(t, y)\right| \leq L, \quad \text{for all } (t, y) \in D, \tag{5.1}$$

then f satisfies a Lipschitz condition on D in the variable y with Lipschitz constant L. ■

Rudolf Lipschitz (1832–1903) worked in many branches of mathematics, including number theory, Fourier series, differential equations, analytical mechanics, and potential theory. He is best known for this generalization of the work of Augustin-Louis Cauchy (1789–1857) and Guiseppe Peano (1856–1932).

The proof of Theorem 5.3 is discussed in Exercise 6; it is similar to the proof of the corresponding result for functions of one variable discussed in Exercise 27 of Section 1.1.

As the next theorem will show, it is often of significant interest to determine whether the function involved in an initial-value problem satisfies a Lipschitz condition in its second

variable, and condition (5.1) is generally easier to apply than the definition. We should note, however, that Theorem 5.3 gives only sufficient conditions for a Lipschitz condition to hold. The function in Example 1, for instance, satisfies a Lipschitz condition, but the partial derivative with respect to y does not exist when $y = 0$.

The following theorem is a version of the fundamental existence and uniqueness theorem for first-order ordinary differential equations. Although the theorem can be proved with the hypothesis reduced somewhat, this form of the theorem is sufficient for our purposes. (The proof of the theorem, in approximately this form, can be found in [BiR], pp. 142–155.)

Theorem 5.4 Suppose that $D = \{(t, y) \mid a \le t \le b \text{ and } -\infty < y < \infty\}$ and that $f(t, y)$ is continuous on D. If f satisfies a Lipschitz condition on D in the variable y, then the initial-value problem

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

has a unique solution $y(t)$ for $a \le t \le b$. ■

Example 2 Use Theorem 5.4 to show that there is a unique solution to the initial-value problem

$$y' = 1 + t \sin(ty), \quad 0 \le t \le 2, \quad y(0) = 0.$$

Solution Holding t constant and applying the Mean Value Theorem to the function

$$f(t, y) = 1 + t \sin(ty),$$

we find that when $y_1 < y_2$, a number ξ in (y_1, y_2) exists with

$$\frac{f(t, y_2) - f(t, y_1)}{y_2 - y_1} = \frac{\partial}{\partial y} f(t, \xi) = t^2 \cos(\xi t).$$

Thus

$$|f(t, y_2) - f(t, y_1)| = |y_2 - y_1||t^2 \cos(\xi t)| \le 4|y_2 - y_1|,$$

and f satisfies a Lipschitz condition in the variable y with Lipschitz constant $L = 4$. Additionally, $f(t, y)$ is continuous when $0 \le t \le 2$ and $-\infty < y < \infty$, so Theorem 5.4 implies that a unique solution exists to this initial-value problem.

If you have completed a course in differential equations you might try to find the exact solution to this problem. ■

Well-Posed Problems

Now that we have, to some extent, taken care of the question of when initial-value problems have unique solutions, we can move to the second important consideration when approximating the solution to an initial-value problem. Initial-value problems obtained by observing physical phenomena generally only approximate the true situation, so we need to know whether small changes in the statement of the problem introduce correspondingly small changes in the solution. This is also important because of the introduction of round-off error when numerical methods are used. That is,

- Question: How do we determine whether a particular problem has the property that small changes, or perturbations, in the statement of the problem introduce correspondingly small changes in the solution?

As usual, we first need to give a workable definition to express this concept.

Definition 5.5 The initial-value problem

$$\frac{dy}{dt} = f(t, y), \quad a \le t \le b, \quad y(a) = \alpha, \tag{5.2}$$

is said to be a **well-posed problem** if:

- A unique solution, $y(t)$, to the problem exists, and
- There exist constants $\varepsilon_0 > 0$ and $k > 0$ such that for any ε, with $\varepsilon_0 > \varepsilon > 0$, whenever $\delta(t)$ is continuous with $|\delta(t)| < \varepsilon$ for all t in $[a, b]$, and when $|\delta_0| < \varepsilon$, the initial-value problem

$$\frac{dz}{dt} = f(t, z) + \delta(t), \quad a \le t \le b, \quad z(a) = \alpha + \delta_0, \tag{5.3}$$

has a unique solution $z(t)$ that satisfies

$$|z(t) - y(t)| < k\varepsilon \quad \text{for all } t \text{ in } [a, b]. \qquad \blacksquare$$

The problem specified by (5.3) is called a **perturbed problem** associated with the original problem (5.2). It assumes the possibility of an error being introduced in the statement of the differential equation, as well as an error δ_0 being present in the initial condition.

Numerical methods will always be concerned with solving a perturbed problem because any round-off error introduced in the representation perturbs the original problem. Unless the original problem is well-posed, there is little reason to expect that the numerical solution to a perturbed problem will accurately approximate the solution to the original problem.

The following theorem specifies conditions that ensure that an initial-value problem is well-posed. The proof of this theorem can be found in [BiR], pp. 142–147.

Theorem 5.6 Suppose $D = \{(t, y) \mid a \le t \le b \text{ and } -\infty < y < \infty\}$. If f is continuous and satisfies a Lipschitz condition in the variable y on the set D, then the initial-value problem

$$\frac{dy}{dt} = f(t, y), \quad a \le t \le b, \quad y(a) = \alpha$$

is well-posed. $\blacksquare$

Example 3 Show that the initial-value problem

$$\frac{dy}{dt} = y - t^2 + 1, \quad 0 \le t \le 2, \quad y(0) = 0.5. \tag{5.4}$$

is well posed on $D = \{(t, y) \mid 0 \le t \le 2 \text{ and } -\infty < y < \infty\}$.

Solution Because

$$\left| \frac{\partial (y - t^2 + 1)}{\partial y} \right| = |1| = 1,$$

Theorem 5.3 implies that $f(t, y) = y - t^2 + 1$ satisfies a Lipschitz condition in y on D with Lipschitz constant 1. Since f is continuous on D, Theorem 5.6 implies that the problem is well-posed.

As an illustration, consider the solution to the perturbed problem

$$\frac{dz}{dt} = z - t^2 + 1 + \delta, \quad 0 \le t \le 2, \quad z(0) = 0.5 + \delta_0, \tag{5.5}$$

where δ and δ_0 are constants. The solutions to Eqs. (5.4) and (5.5) are

$$y(t) = (t+1)^2 - 0.5e^t \quad \text{and} \quad z(t) = (t+1)^2 + (\delta + \delta_0 - 0.5)e^t - \delta,$$

respectively.

Suppose that ε is a positive number. If $|\delta| < \varepsilon$ and $|\delta_0| < \varepsilon$, then

$$|y(t) - z(t)| = |(\delta + \delta_0)e^t - \delta| \le |\delta + \delta_0|e^2 + |\delta| \le (2e^2 + 1)\varepsilon,$$

for all t. This implies that problem (5.4) is well-posed with $k(\varepsilon) = 2e^2 + 1$ for all $\varepsilon > 0$. ■

Maple can be used to solve many initial-value problems. Consider the problem

$$\frac{dy}{dt} = y - t^2 + 1, \quad 0 \le t \le 2, \quad y(0) = 0.5.$$

Maple reserves the letter *D* to represent differentiation.

To define the differential equation and initial condition, enter

$deq := D(y)(t) = y(t) - t^2 + 1; \; init := y(0) = 0.5$

The names *deq* and *init* have been chosen by the user. The command to solve the initial-value problems is

$deqsol := dsolve(\{deq, init\}, y(t))$

and Maple responds with

$$y(t) = 1 + t^2 + 2t - \frac{1}{2}e^t$$

To use the solution to obtain a specific value, such as $y(1.5)$, we enter

$q := rhs(deqsol) : \; evalf(subs(t = 1.5, q))$

which gives

4.009155465

The function *rhs* (for *r*ight *h*and *s*ide) is used to assign the solution of the initial-value problem to the function q, which we then evaluate at $t = 1.5$.

The function *dsolve* can fail if an explicit solution to the initial-value problem cannot be found. For example, for the initial-value problem given in Example 2, the command

$deqsol2 := dsolve(\{D(y)(t) = 1 + t \cdot \sin(t \cdot y(t)), \; y(0) = 0\}, y(t))$

does not succeed because an explicit solution cannot be found. In this case a numerical method must be used.

EXERCISE SET 5.1

1. Use Theorem 5.4 to show that each of the following initial-value problems has a unique solution, and find the solution.

a. $y' = y \cos t, \quad 0 \le t \le 1, \quad y(0) = 1.$

b. $y' = \frac{2}{t}y + t^2 e^t, \quad 1 \le t \le 2, \quad y(1) = 0.$

c. $y' = -\frac{2}{t}y + t^2 e^t, \quad 1 \le t \le 2, \quad y(1) = \sqrt{2}e.$

d. $y' = \frac{4t^3 y}{1 + t^4}, \quad 0 \le t \le 1, \quad y(0) = 1.$

2. Show that each of the following initial-value problems has a unique solution and find the solution. Can Theorem 5.4 be applied in each case?

a. $y' = e^{t-y}, \quad 0 \le t \le 1, \quad y(0) = 1.$

b. $y' = t^{-2}(\sin 2t - 2ty), \quad 1 \le t \le 2, \quad y(1) = 2.$

c. $y' = -y + ty^{1/2}, \quad 2 \le t \le 3, \quad y(2) = 2.$

d. $y' = \dfrac{ty + y}{ty + t}, \quad 2 \le t \le 4, \quad y(2) = 4.$

3. For each choice of $f(t, y)$ given in parts (a)–(d):

i. Does f satisfy a Lipschitz condition on $D = \{(t, y) \mid 0 \le t \le 1, -\infty < y < \infty\}$?

ii. Can Theorem 5.6 be used to show that the initial-value problem

$$y' = f(t, y), \quad 0 \le t \le 1, \quad y(0) = 1,$$

is well-posed?

a. $f(t, y) = t^2 y + 1$ **b.** $f(t, y) = ty$ **c.** $f(t, y) = 1 - y$ **d.** $f(t, y) = -ty + \dfrac{4t}{y}$

4. For each choice of $f(t, y)$ given in parts (a)–(d):

i. Does f satisfy a Lipschitz condition on $D = \{(t, y) \mid 0 \le t \le 1, -\infty < y < \infty\}$?

ii. Can Theorem 5.6 be used to show that the initial-value problem

$$y' = f(t, y), \quad 0 \le t \le 1, \quad y(0) = 1,$$

is well-posed?

a. $f(t, y) = e^{t-y}$ **b.** $f(t, y) = \dfrac{1 + y}{1 + t}$ **c.** $f(t, y) = \cos(yt)$ **d.** $f(t, y) = \dfrac{y^2}{1 + t}$

5. For the following initial-value problems, show that the given equation implicitly defines a solution. Approximate $y(2)$ using Newton's method.

a. $y' = -\dfrac{y^3 + y}{(3y^2 + 1)t}, \quad 1 \le t \le 2, \quad y(1) = 1; \quad y^3 t + yt = 2$

b. $y' = -\dfrac{y \cos t + 2te^y}{\sin t + t^2 e^y + 2}, \quad 1 \le t \le 2, \quad y(1) = 0; \quad y \sin t + t^2 e^y + 2y = 1$

6. Prove Theorem 5.3 by applying the Mean Value Theorem 1,8 to $f(t, y)$, holding t fixed.

7. Show that, for any constants a and b, the set $D = \{(t, y) \mid a \le t \le b, -\infty < y < \infty\}$ is convex.

8. Suppose the perturbation $\delta(t)$ is proportional to t, that is, $\delta(t) = \delta t$ for some constant δ. Show directly that the following initial-value problems are well-posed.

a. $y' = 1 - y, \quad 0 \le t \le 2, \quad y(0) = 0$

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

c. $y' = \dfrac{2}{t} y + t^2 e^t, \quad 1 \le t \le 2, \quad y(1) = 0$

d. $y' = -\dfrac{2}{t} y + t^2 e^t, \quad 1 \le t \le 2, \quad y(1) = \sqrt{2}e$

9. *Picard's method* for solving the initial-value problem

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

is described as follows: Let $y_0(t) = \alpha$ for each t in $[a, b]$. Define a sequence $\{y_k(t)\}$ of functions by

$$y_k(t) = \alpha + \int_a^t f(\tau, y_{k-1}(\tau))\, d\tau, \quad k = 1, 2, \ldots.$$

a. Integrate $y' = f(t, y(t))$, and use the initial condition to derive Picard's method.

b. Generate $y_0(t)$, $y_1(t)$, $y_2(t)$, and $y_3(t)$ for the initial-value problem

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

c. Compare the result in part (b) to the Maclaurin series of the actual solution $y(t) = t + e^{-t}$.

5.2 Euler's Method

Euler's method is the most elementary approximation technique for solving initial-value problems. Although it is seldom used in practice, the simplicity of its derivation can be used to illustrate the techniques involved in the construction of some of the more advanced techniques, without the cumbersome algebra that accompanies these constructions.

The object of Euler's method is to obtain approximations to the well-posed initial-value problem

$$\frac{dy}{dt} = f(t, y), \quad a \le t \le b, \quad y(a) = \alpha. \tag{5.6}$$

A continuous approximation to the solution $y(t)$ will not be obtained; instead, approximations to y will be generated at various values, called **mesh points**, in the interval $[a, b]$. Once the approximate solution is obtained at the points, the approximate solution at other points in the interval can be found by interpolation.

We first make the stipulation that the mesh points are equally distributed throughout the interval $[a, b]$. This condition is ensured by choosing a positive integer N and selecting the mesh points

$$t_i = a + ih, \quad \text{for each } i = 0, 1, 2, \ldots, N.$$

The common distance between the points $h = (b - a)/N = t_{i+1} - t_i$ is called the **step size**.

The use of elementary difference methods to approximate the solution to differential equations was one of the numerous mathematical topics that was first presented to the mathematical public by the most prolific of mathematicians, Leonhard Euler (1707–1783).

We will use Taylor's Theorem to derive Euler's method. Suppose that $y(t)$, the unique solution to (5.6), has two continuous derivatives on $[a, b]$, so that for each $i = 0, 1, 2, \ldots, N - 1$,

$$y(t_{i+1}) = y(t_i) + (t_{i+1} - t_i)y'(t_i) + \frac{(t_{i+1} - t_i)^2}{2}y''(\xi_i),$$

for some number ξ_i in (t_i, t_{i+1}). Because $h = t_{i+1} - t_i$, we have

$$y(t_{i+1}) = y(t_i) + hy'(t_i) + \frac{h^2}{2}y''(\xi_i),$$

and, because $y(t)$ satisfies the differential equation (5.6),

$$y(t_{i+1}) = y(t_i) + hf(t_i, y(t_i)) + \frac{h^2}{2}y''(\xi_i). \tag{5.7}$$

Euler's method constructs $w_i \approx y(t_i)$, for each $i = 1, 2, \ldots, N$, by deleting the remainder term. Thus Euler's method is

$$\begin{aligned} w_0 &= \alpha, \\ w_{i+1} &= w_i + hf(t_i, w_i), \quad \text{for each } i = 0, 1, \ldots, N - 1. \end{aligned} \tag{5.8}$$

Illustration In Example 1 we will use an algorithm for Euler's method to approximate the solution to

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

at $t = 2$. Here we will simply illustrate the steps in the technique when we have $h = 0.5$.

For this problem $f(t, y) = y - t^2 + 1$, so

$$w_0 = y(0) = 0.5;$$

$$w_1 = w_0 + 0.5\left(w_0 - (0.0)^2 + 1\right) = 0.5 + 0.5(1.5) = 1.25;$$

$$w_2 = w_1 + 0.5\left(w_1 - (0.5)^2 + 1\right) = 1.25 + 0.5(2.0) = 2.25;$$

$$w_3 = w_2 + 0.5\left(w_2 - (1.0)^2 + 1\right) = 2.25 + 0.5(2.25) = 3.375;$$

and

$$y(2) \approx w_4 = w_3 + 0.5\left(w_3 - (1.5)^2 + 1\right) = 3.375 + 0.5(2.125) = 4.4375. \quad \square$$

Equation (5.8) is called the **difference equation** associated with Euler's method. As we will see later in this chapter, the theory and solution of difference equations parallel, in many ways, the theory and solution of differential equations. Algorithm 5.1 implements Euler's method.

ALGORITHM
5.1

Euler's

To approximate the solution of the initial-value problem

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

at $(N + 1)$ equally spaced numbers in the interval $[a, b]$:

INPUT endpoints a, b; integer N; initial condition α.

OUTPUT approximation w to y at the $(N + 1)$ values of t.

Step 1 Set $h = (b - a)/N$;
$t = a$;
$w = \alpha$;
OUTPUT (t, w).

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

Step 3 Set $w = w + hf(t, w)$; (*Compute* w_i.)
$t = a + ih$. (*Compute* t_i.)

Step 4 OUTPUT (t, w).

Step 5 STOP. ■

To interpret Euler's method geometrically, note that when w_i is a close approximation to $y(t_i)$, the assumption that the problem is well-posed implies that

$$f(t_i, w_i) \approx y'(t_i) = f(t_i, y(t_i)).$$

The graph of the function highlighting $y(t_i)$ is shown in Figure 5.2. One step in Euler's method appears in Figure 5.3, and a series of steps appears in Figure 5.4.

Figure 5.2

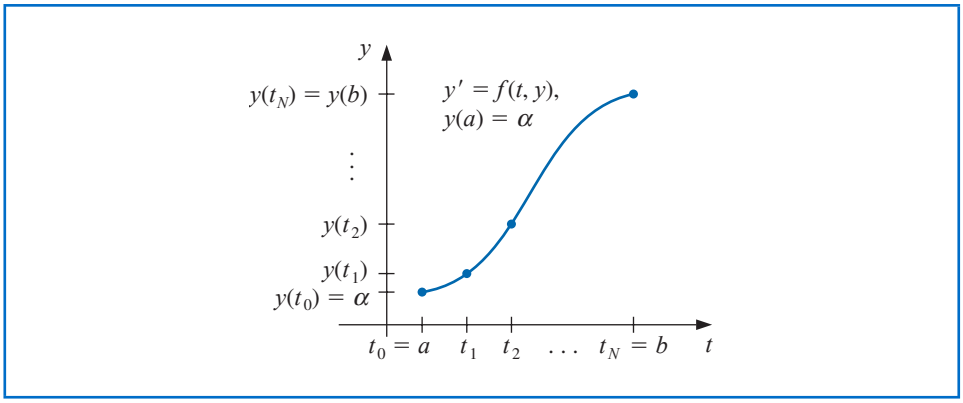


Figure 5.3

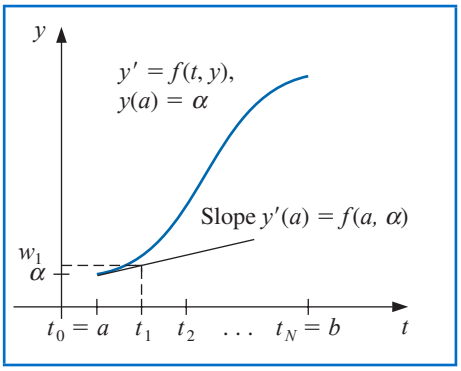
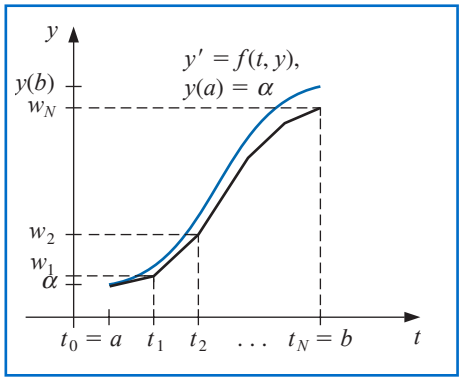


Figure 5.4



Example 1 Euler's method was used in the first illustration with $h = 0.5$ to approximate the solution to the initial-value problem

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

Use Algorithm 5.1 with $N = 10$ to determine approximations, and compare these with the exact values given by $y(t) = (t + 1)^2 - 0.5e^t$.

Solution With $N = 10$ we have $h = 0.2$, $t_i = 0.2i$, $w_0 = 0.5$, and

$$w_{i+1} = w_i + h(w_i - t_i^2 + 1) = w_i + 0.2[w_i - 0.04i^2 + 1] = 1.2w_i - 0.008i^2 + 0.2,$$

for $i = 0, 1, \ldots, 9$. So

$$w_1 = 1.2(0.5) - 0.008(0)^2 + 0.2 = 0.8; \quad w_2 = 1.2(0.8) - 0.008(1)^2 + 0.2 = 1.152;$$

and so on. Table 5.1 shows the comparison between the approximate values at t_i and the actual values. ■

Table 5.1

t_i	w_i	$y_i = y(t_i)$	$\lvert y_i - w_i \rvert$
0.0	0.5000000	0.5000000	0.0000000
0.2	0.8000000	0.8292986	0.0292986
0.4	1.1520000	1.2140877	0.0620877
0.6	1.5504000	1.6489406	0.0985406
0.8	1.9884800	2.1272295	0.1387495
1.0	2.4581760	2.6408591	0.1826831
1.2	2.9498112	3.1799415	0.2301303
1.4	3.4517734	3.7324000	0.2806266
1.6	3.9501281	4.2834838	0.3333557
1.8	4.4281538	4.8151763	0.3870225
2.0	4.8657845	5.3054720	0.4396874

Note that the error grows slightly as the value of t increases. This controlled error growth is a consequence of the stability of Euler's method, which implies that the error is expected to grow in no worse than a linear manner.

Maple has implemented Euler's method as an option with the command *InitialValueProblem* within the *NumericalAnalysis* subpackage of the *Student* package. To use it for the problem in Example 1 first load the package and the differential equation.

$with(Student[NumericalAnalysis]): deq := diff(y(t), t) = y(t) - t^2 + 1$

Then issue the command

$C := InitialValueProblem(deq, y(0) = 0.5, t = 2, method = euler, numsteps = 10, output = information, digits = 8)$

Maple produces

$$\begin{bmatrix} 1..12 \times 1..4\ Array \\ Data\ Type:\ anything \\ Storage:\ rectangular \\ Order:\ Fortran\_order \end{bmatrix}$$

Double clicking on the output brings up a table that gives the values of t_i, actual solution values $y(t_i)$, the Euler approximations w_i, and the absolute errors $|\, y(t_i) - w_i|$. These agree with the values in Table 5.1.

To print the Maple table we can issue the commands

for k **from** 1 **to** 12 **do**
$print(C[k, 1], C[k, 2], C[k, 3], C[k, 4])$
end do

The options within the *InitialValueProblem* command are the specification of the first order differential equation to be solved, the initial condition, the final value of the independent variable, the choice of method, the number of steps used to determine that $h = (2 - 0)/(numsteps)$, the specification of form of the output, and the number of digits of rounding to be used in the computations. Other output options can specify a particular value of t or a plot of the solution.

Error Bounds for Euler's Method

Although Euler's method is not accurate enough to warrant its use in practice, it is sufficiently elementary to analyze the error that is produced from its application. The error analysis for

the more accurate methods that we consider in subsequent sections follows the same pattern but is more complicated.

To derive an error bound for Euler's method, we need two computational lemmas.

Lemma 5.7 For all $x \geq -1$ and any positive m, we have $0 \leq (1+x)^m \leq e^{mx}$. ■

Proof Applying Taylor's Theorem with $f(x) = e^x$, $x_0 = 0$, and $n = 1$ gives

$$e^x = 1 + x + \frac{1}{2}x^2 e^{\xi},$$

where ξ is between x and zero. Thus

$$0 \leq 1 + x \leq 1 + x + \frac{1}{2}x^2 e^{\xi} = e^x,$$

and, because $1 + x \geq 0$, we have

$$0 \leq (1+x)^m \leq (e^x)^m = e^{mx}. \quad ■ ■ ■$$

Lemma 5.8 If s and t are positive real numbers, $\{a_i\}_{i=0}^k$ is a sequence satisfying $a_0 \geq -t/s$, and

$$a_{i+1} \leq (1+s)a_i + t, \quad \text{for each } i = 0, 1, 2, \ldots, k-1, \tag{5.9}$$

then

$$a_{i+1} \leq e^{(i+1)s}\left(a_0 + \frac{t}{s}\right) - \frac{t}{s}. \quad ■$$

Proof For a fixed integer i, Inequality (5.9) implies that

$$\begin{aligned}
a_{i+1} &\leq (1+s)a_i + t \\
&\leq (1+s)[(1+s)a_{i-1} + t] + t = (1+s)^2 a_{i-1} + [1 + (1+s)]t \\
&\leq (1+s)^3 a_{i-2} + \left[1 + (1+s) + (1+s)^2\right]t \\
&\vdots \\
&\leq (1+s)^{i+1} a_0 + \left[1 + (1+s) + (1+s)^2 + \cdots + (1+s)^i\right]t.
\end{aligned}$$

But

$$1 + (1+s) + (1+s)^2 + \cdots + (1+s)^i = \sum_{j=0}^{i} (1+s)^j$$

is a geometric series with ratio $(1+s)$ that sums to

$$\frac{1 - (1+s)^{i+1}}{1 - (1+s)} = \frac{1}{s}[(1+s)^{i+1} - 1].$$

Thus

$$a_{i+1} \leq (1+s)^{i+1} a_0 + \frac{(1+s)^{i+1} - 1}{s} t = (1+s)^{i+1}\left(a_0 + \frac{t}{s}\right) - \frac{t}{s},$$

and using Lemma 5.7 with $x = 1 + s$ gives

$$a_{i+1} \leq e^{(i+1)s}\left(a_0 + \frac{t}{s}\right) - \frac{t}{s}. \quad ■ ■ ■$$

Theorem 5.9 Suppose f is continuous and satisfies a Lipschitz condition with constant L on

$$D = \{(t, y) \mid a \le t \le b \text{ and } -\infty < y < \infty\}$$

and that a constant M exists with

$$|y''(t)| \le M, \quad \text{for all } t \in [a, b],$$

where $y(t)$ denotes the unique solution to the initial-value problem

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

Let $w_0, w_1, \ldots, w_N$ be the approximations generated by Euler's method for some positive integer N. Then, for each $i = 0, 1, 2, \ldots, N$,

$$|y(t_i) - w_i| \le \frac{hM}{2L}\left[e^{L(t_i - a)} - 1\right]. \tag{5.10}$$

■

Proof When $i = 0$ the result is clearly true, since $y(t_0) = w_0 = \alpha$.

From Eq. (5.7), we have

$$y(t_{i+1}) = y(t_i) + hf(t_i, y(t_i)) + \frac{h^2}{2}y''(\xi_i),$$

for $i = 0, 1, \ldots, N - 1$, and from the equations in (5.8),

$$w_{i+1} = w_i + hf(t_i, w_i).$$

Using the notation $y_i = y(t_i)$ and $y_{i+1} = y(t_{i+1})$, we subtract these two equations to obtain

$$y_{i+1} - w_{i+1} = y_i - w_i + h[f(t_i, y_i) - f(t_i, w_i)] + \frac{h^2}{2}y''(\xi_i)$$

Hence

$$|y_{i+1} - w_{i+1}| \le |y_i - w_i| + h|f(t_i, y_i) - f(t_i, w_i)| + \frac{h^2}{2}|y''(\xi_i)|.$$

Now f satisfies a Lipschitz condition in the second variable with constant L, and $|y''(t)| \le M$, so

$$|y_{i+1} - w_{i+1}| \le (1 + hL)|y_i - w_i| + \frac{h^2M}{2}.$$

Referring to Lemma 5.8 and letting $s = hL$, $t = h^2M/2$, and $a_j = |y_j - w_j|$, for each $j = 0, 1, \ldots, N$, we see that

$$|y_{i+1} - w_{i+1}| \le e^{(i+1)hL}\left(|y_0 - w_0| + \frac{h^2M}{2hL}\right) - \frac{h^2M}{2hL}.$$

Because $|y_0 - w_0| = 0$ and $(i + 1)h = t_{i+1} - t_0 = t_{i+1} - a$, this implies that

$$|y_{i+1} - w_{i+1}| \le \frac{hM}{2L}(e^{(t_{i+1} - a)L} - 1),$$

for each $i = 0, 1, \ldots, N - 1$. ■ ■ ■

The weakness of Theorem 5.9 lies in the requirement that a bound be known for the second derivative of the solution. Although this condition often prohibits us from obtaining a realistic error bound, it should be noted that if $\partial f/\partial t$ and $\partial f/\partial y$ both exist, the chain rule for partial differentiation implies that

$$y''(t) = \frac{dy'}{dt}(t) = \frac{df}{dt}(t, y(t)) = \frac{\partial f}{\partial t}(t, y(t)) + \frac{\partial f}{\partial y}(t, y(t)) \cdot f(t, y(t)).$$

So it is at times possible to obtain an error bound for $y''(t)$ without explicitly knowing $y(t)$.

Example 2 The solution to the initial-value problem

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

was approximated in Example 1 using Euler's method with $h = 0.2$. Use the inequality in Theorem 5.9 to find a bounds for the approximation errors and compare these to the actual errors.

Solution Because $f(t, y) = y - t^2 + 1$, we have $\partial f(t, y)/\partial y = 1$ for all y, so $L = 1$. For this problem, the exact solution is $y(t) = (t + 1)^2 - 0.5e^t$, so $y''(t) = 2 - 0.5e^t$ and

$$|y''(t)| \le 0.5e^2 - 2, \quad \text{for all } t \in [0, 2].$$

Using the inequality in the error bound for Euler's method with $h = 0.2$, $L = 1$, and $M = 0.5e^2 - 2$ gives

$$|y_i - w_i| \le 0.1(0.5e^2 - 2)(e^{t_i} - 1).$$

Hence

$$|y(0.2) - w_1| \le 0.1(0.5e^2 - 2)(e^{0.2} - 1) = 0.03752;$$
$$|y(0.4) - w_2| \le 0.1(0.5e^2 - 2)(e^{0.4} - 1) = 0.08334;$$

and so on. Table 5.2 lists the actual error found in Example 1, together with this error bound. Note that even though the true bound for the second derivative of the solution was used, the error bound is considerably larger than the actual error, especially for increasing values of t. ■

Table 5.2

t_i	0.2	0.4	0.6	0.8	1.0	1.2	1.4	1.6	1.8	2.0
Actual Error	0.02930	0.06209	0.09854	0.13875	0.18268	0.23013	0.28063	0.33336	0.38702	0.43969
Error Bound	0.03752	0.08334	0.13931	0.20767	0.29117	0.39315	0.51771	0.66985	0.85568	1.08264

The principal importance of the error-bound formula given in Theorem 5.9 is that the bound depends linearly on the step size h. Consequently, diminishing the step size should give correspondingly greater accuracy to the approximations.

Neglected in the result of Theorem 5.9 is the effect that round-off error plays in the choice of step size. As h becomes smaller, more calculations are necessary and more round-off error is expected. In actuality then, the difference-equation form

$$w_0 = \alpha,$$
$$w_{i+1} = w_i + hf(t_i, w_i), \quad \text{for each } i = 0, 1, \ldots, N - 1,$$

is not used to calculate the approximation to the solution y_i at a mesh point t_i. We use instead an equation of the form

$$u_0 = \alpha + \delta_0,$$
$$u_{i+1} = u_i + hf(t_i, u_i) + \delta_{i+1}, \quad \text{for each } i = 0, 1, \ldots, N-1, \tag{5.11}$$

where δ_i denotes the round-off error associated with u_i. Using methods similar to those in the proof of Theorem 5.9, we can produce an error bound for the finite-digit approximations to y_i given by Euler's method.

Theorem 5.10 Let $y(t)$ denote the unique solution to the initial-value problem

$$y' = f(t, y), \quad a \le t \le b, \quad y(a) = \alpha \tag{5.12}$$

and $u_0, u_1, \ldots, u_N$ be the approximations obtained using (5.11). If $|\delta_i| < \delta$ for each $i = 0, 1, \ldots, N$ and the hypotheses of Theorem 5.9 hold for (5.12), then

$$|y(t_i) - u_i| \le \frac{1}{L}\left(\frac{hM}{2} + \frac{\delta}{h}\right)[e^{L(t_i - a)} - 1] + |\delta_0| e^{L(t_i - a)}, \tag{5.13}$$

for each $i = 0, 1, \ldots, N$. ■

The error bound (5.13) is no longer linear in h. In fact, since

$$\lim_{h \to 0}\left(\frac{hM}{2} + \frac{\delta}{h}\right) = \infty,$$

the error would be expected to become large for sufficiently small values of h. Calculus can be used to determine a lower bound for the step size h. Letting $E(h) = (hM/2) + (\delta/h)$ implies that $E'(h) = (M/2) - (\delta/h^2)$.

If $h < \sqrt{2\delta/M}$, then $E'(h) < 0$ and $E(h)$ is decreasing.

If $h > \sqrt{2\delta/M}$, then $E'(h) > 0$ and $E(h)$ is increasing.

The minimal value of $E(h)$ occurs when

$$h = \sqrt{\frac{2\delta}{M}}. \tag{5.14}$$

Decreasing h beyond this value tends to increase the total error in the approximation. Normally, however, the value of δ is sufficiently small that this lower bound for h does not affect the operation of Euler's method.

EXERCISE SET 5.2

1. Use Euler's method to approximate the solutions for each of the following initial-value problems.

a. $y' = te^{3t} - 2y, \quad 0 \le t \le 1, \quad y(0) = 0,$ with $h = 0.5$

b. $y' = 1 + (t - y)^2, \quad 2 \le t \le 3, \quad y(2) = 1,$ with $h = 0.5$

c. $y' = 1 + y/t, \quad 1 \le t \le 2, \quad y(1) = 2,$ with $h = 0.25$

d. $y' = \cos 2t + \sin 3t, \quad 0 \le t \le 1, \quad y(0) = 1,$ with $h = 0.25$

2. Use Euler's method to approximate the solutions for each of the following initial-value problems.

a. $y' = e^{t-y}, \quad 0 \le t \le 1, \quad y(0) = 1, \text{ with } h = 0.5$

b. $y' = \dfrac{1+t}{1+y}, \quad 1 \le t \le 2, \quad y(1) = 2, \text{ with } h = 0.5$

c. $y' = -y + ty^{1/2}, \quad 2 \le t \le 3, \quad y(2) = 2, \text{ with } h = 0.25$

d. $y' = t^{-2}(\sin 2t - 2ty), \quad 1 \le t \le 2, \quad y(1) = 2, \text{ with } h = 0.25$

3. The actual solutions to the initial-value problems in Exercise 1 are given here. Compare the actual error at each step to the error bound.

a. $y(t) = \dfrac{1}{5}te^{3t} - \dfrac{1}{25}e^{3t} + \dfrac{1}{25}e^{-2t}$

b. $y(t) = t + \dfrac{1}{1-t}$

c. $y(t) = t\ln t + 2t$

d. $y(t) = \dfrac{1}{2}\sin 2t - \dfrac{1}{3}\cos 3t + \dfrac{4}{3}$

4. The actual solutions to the initial-value problems in Exercise 2 are given here. Compute the actual error and compare this to the error bound if Theorem 5.9 can be applied.

a. $y(t) = \ln(e^t + e - 1)$

b. $y(t) = \sqrt{t^2 + 2t + 6} - 1$

c. $y(t) = \left(t - 2 + \sqrt{2}ee^{-t/2}\right)^2$

d. $y(t) = \dfrac{4 + \cos 2 - \cos 2t}{2t^2}$

5. Use Euler's method to approximate the solutions for each of the following initial-value problems.

a. $y' = y/t - (y/t)^2, \quad 1 \le t \le 2, \quad y(1) = 1, \text{ with } h = 0.1$

b. $y' = 1 + y/t + (y/t)^2, \quad 1 \le t \le 3, \quad y(1) = 0, \text{ with } h = 0.2$

c. $y' = -(y+1)(y+3), \quad 0 \le t \le 2, \quad y(0) = -2, \text{ with } h = 0.2$

d. $y' = -5y + 5t^2 + 2t, \quad 0 \le t \le 1, \quad y(0) = \frac{1}{3}, \text{ with } h = 0.1$

6. Use Euler's method to approximate the solutions for each of the following initial-value problems.

a. $y' = \dfrac{2 - 2ty}{t^2 + 1}, \quad 0 \le t \le 1, \quad y(0) = 1, \text{ with } h = 0.1$

b. $y' = \dfrac{y^2}{1+t}, \quad 1 \le t \le 2, \quad y(1) = -(\ln 2)^{-1}, \text{ with } h = 0.1$

c. $y' = (y^2 + y)/t, \quad 1 \le t \le 3, \quad y(1) = -2, \text{ with } h = 0.2$

d. $y' = -ty + 4ty^{-1}, \quad 0 \le t \le 1, \quad y(0) = 1, \text{ with } h = 0.1$

7. The actual solutions to the initial-value problems in Exercise 5 are given here. Compute the actual error in the approximations of Exercise 5.

a. $y(t) = \dfrac{t}{1 + \ln t}$

b. $y(t) = t\tan(\ln t)$

c. $y(t) = -3 + \dfrac{2}{1 + e^{-2t}}$

d. $y(t) = t^2 + \dfrac{1}{3}e^{-5t}$

8. The actual solutions to the initial-value problems in Exercise 6 are given here. Compute the actual error in the approximations of Exercise 6.

a. $y(t) = \dfrac{2t+1}{t^2+1}$

b. $y(t) = \dfrac{-1}{\ln(t+1)}$

c. $y(t) = \dfrac{2t}{1-2t}$

d. $y(t) = \sqrt{4 - 3e^{-t^2}}$

9. Given the initial-value problem

$$y' = \frac{2}{t}y + t^2e^t, \quad 1 \le t \le 2, \quad y(1) = 0,$$

with exact solution $y(t) = t^2(e^t - e)$:

a. Use Euler's method with $h = 0.1$ to approximate the solution, and compare it with the actual values of y.

b. Use the answers generated in part (a) and linear interpolation to approximate the following values of y, and compare them to the actual values.

i. $y(1.04)$ **ii.** $y(1.55)$ **iii.** $y(1.97)$

c. Compute the value of h necessary for $|y(t_i) - w_i| \leq 0.1$, using Eq. (5.10).

10. Given the initial-value problem

$$y' = \frac{1}{t^2} - \frac{y}{t} - y^2, \quad 1 \leq t \leq 2, \quad y(1) = -1,$$

with exact solution $y(t) = -1/t$:

a. Use Euler's method with $h = 0.05$ to approximate the solution, and compare it with the actual values of y.

b. Use the answers generated in part (a) and linear interpolation to approximate the following values of y, and compare them to the actual values.

i. $y(1.052)$ **ii.** $y(1.555)$ **iii.** $y(1.978)$

c. Compute the value of h necessary for $|y(t_i) - w_i| \leq 0.05$ using Eq. (5.10).

11. Given the initial-value problem

$$y' = -y + t + 1, \quad 0 \leq t \leq 5, \quad y(0) = 1,$$

with exact solution $y(t) = e^{-t} + t$:

a. Approximate $y(5)$ using Euler's method with $h = 0.2$, $h = 0.1$, and $h = 0.05$.

b. Determine the optimal value of h to use in computing $y(5)$, assuming $\delta = 10^{-6}$ and that Eq. (5.14) is valid.

12. Consider the initial-value problem

$$y' = -10y, \quad 0 \leq t \leq 2, \quad y(0) = 1,$$

which has solution $y(t) = e^{-10t}$. What happens when Euler's method is applied to this problem with $h = 0.1$? Does this behavior violate Theorem 5.9?

13. Use the results of Exercise 5 and linear interpolation to approximate the following values of $y(t)$. Compare the approximations obtained to the actual values obtained using the functions given in Exercise 7.

a. $y(1.25)$ and $y(1.93)$ **b.** $y(2.1)$ and $y(2.75)$

c. $y(1.3)$ and $y(1.93)$ **d.** $y(0.54)$ and $y(0.94)$

14. Use the results of Exercise 6 and linear interpolation to approximate the following values of $y(t)$. Compare the approximations obtained to the actual values obtained using the functions given in Exercise 8.

a. $y(0.25)$ and $y(0.93)$ **b.** $y(1.25)$ and $y(1.93)$

c. $y(2.10)$ and $y(2.75)$ **d.** $y(0.54)$ and $y(0.94)$

15. Let $E(h) = \dfrac{hM}{2} + \dfrac{\delta}{h}$.

a. For the initial-value problem

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

compute the value of h to minimize $E(h)$. Assume $\delta = 5 \times 10^{-(n+1)}$ if you will be using n-digit arithmetic in part (c).

b. For the optimal h computed in part (a), use Eq. (5.13) to compute the minimal error obtainable.

c. Compare the actual error obtained using $h = 0.1$ and $h = 0.01$ to the minimal error in part (b). Can you explain the results?

16. In a circuit with impressed voltage $\mathcal{E}$ having resistance R, inductance L, and capacitance C in parallel, the current i satisfies the differential equation

$$\frac{di}{dt} = C\frac{d^2\mathcal{E}}{dt^2} + \frac{1}{R}\frac{d\mathcal{E}}{dt} + \frac{1}{L}\mathcal{E}.$$

Suppose $C = 0.3$ farads, $R = 1.4$ ohms, $L = 1.7$ henries, and the voltage is given by

$$\mathcal{E}(t) = e^{-0.06\pi t}\sin(2t - \pi).$$

If $i(0) = 0$, find the current i for the values $t = 0.1j$, where $j = 0, 1, \ldots, 100$.

17. In a book entitled *Looking at History Through Mathematics*, Rashevsky [Ra], pp. 103–110, considers a model for a problem involving the production of nonconformists in society. Suppose that a society has a population of $x(t)$ individuals at time t, in years, and that all nonconformists who mate with other nonconformists have offspring who are also nonconformists, while a fixed proportion r of all other offspring are also nonconformist. If the birth and death rates for all individuals are assumed to be the constants b and d, respectively, and if conformists and nonconformists mate at random, the problem can be expressed by the differential equations

$$\frac{dx(t)}{dt} = (b - d)x(t) \quad \text{and} \quad \frac{dx_n(t)}{dt} = (b - d)x_n(t) + rb(x(t) - x_n(t)),$$

where $x_n(t)$ denotes the number of nonconformists in the population at time t.

a. Suppose the variable $p(t) = x_n(t)/x(t)$ is introduced to represent the proportion of nonconformists in the society at time t. Show that these equations can be combined and simplified to the single differential equation

$$\frac{dp(t)}{dt} = rb(1 - p(t)).$$

b. Assuming that $p(0) = 0.01$, $b = 0.02$, $d = 0.015$, and $r = 0.1$, approximate the solution $p(t)$ from $t = 0$ to $t = 50$ when the step size is $h = 1$ year.

c. Solve the differential equation for $p(t)$ exactly, and compare your result in part (b) when $t = 50$ with the exact value at that time.

5.3 Higher-Order Taylor Methods

Since the object of a numerical techniques is to determine accurate approximations with minimal effort, we need a means for comparing the efficiency of various approximation methods. The first device we consider is called the *local truncation error* of the method.

The local truncation error at a specified step measures the amount by which the exact solution to the differential equation fails to satisfy the difference equation being used for the approximation at that step. This might seem like an unlikely way to compare the error of various methods. We really want to know how well the approximations generated by the methods satisfy the differential equation, not the other way around. However, we don't know the exact solution so we cannot generally determine this, and the local truncation will serve quite well to determine not only the local error of a method but the actual approximation error.

Consider the initial value problem

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

Definition 5.11 The difference method

$$w_0 = \alpha$$
$$w_{i+1} = w_i + h\phi(t_i, w_i), \quad \text{for each } i = 0, 1, \ldots, N - 1,$$

has **local truncation error**

$$\tau_{i+1}(h) = \frac{y_{i+1} - (y_i + h\phi(t_i, y_i))}{h} = \frac{y_{i+1} - y_i}{h} - \phi(t_i, y_i),$$

for each $i = 0, 1, \ldots, N-1$, where y_i and y_{i+1} denote the solution at t_i and t_{i+1}, respectively. ■

For example, Euler's method has local truncation error at the ith step

$$\tau_{i+1}(h) = \frac{y_{i+1} - y_i}{h} - f(t_i, y_i), \quad \text{for each } i = 0, 1, \ldots, N-1.$$

This error is a *local error* because it measures the accuracy of the method at a specific step, assuming that the method was exact at the previous step. As such, it depends on the differential equation, the step size, and the particular step in the approximation.

By considering Eq. (5.7) in the previous section, we see that Euler's method has

$$\tau_{i+1}(h) = \frac{h}{2} y''(\xi_i), \quad \text{for some } \xi_i \text{ in } (t_i, t_{i+1}).$$

When $y''(t)$ is known to be bounded by a constant M on $[a, b]$, this implies

$$|\tau_{i+1}(h)| \le \frac{h}{2} M,$$

so the local truncation error in Euler's method is $O(h)$.

One way to select difference-equation methods for solving ordinary differential equations is in such a manner that their local truncation errors are $O(h^p)$ for as large a value of p as possible, while keeping the number and complexity of calculations of the methods within a reasonable bound.

Since Euler's method was derived by using Taylor's Theorem with $n = 1$ to approximate the solution of the differential equation, our first attempt to find methods for improving the convergence properties of difference methods is to extend this technique of derivation to larger values of n.

The methods in this section use Taylor polynomials and the knowledge of the derivative at a node to approximate the value of the function at a new node.

Suppose the solution $y(t)$ to the initial-value problem

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

has $(n+1)$ continuous derivatives. If we expand the solution, $y(t)$, in terms of its nth Taylor polynomial about t_i and evaluate at t_{i+1}, we obtain

$$y(t_{i+1}) = y(t_i) + hy'(t_i) + \frac{h^2}{2} y''(t_i) + \cdots + \frac{h^n}{n!} y^{(n)}(t_i) + \frac{h^{n+1}}{(n+1)!} y^{(n+1)}(\xi_i), \tag{5.15}$$

for some ξ_i in (t_i, t_{i+1}).

Successive differentiation of the solution, $y(t)$, gives

$$y'(t) = f(t, y(t)), \quad y''(t) = f'(t, y(t)), \quad \text{and, generally,} \quad y^{(k)}(t) = f^{(k-1)}(t, y(t)).$$

Substituting these results into Eq. (5.15) gives

$$y(t_{i+1}) = y(t_i) + hf(t_i, y(t_i)) + \frac{h^2}{2} f'(t_i, y(t_i)) + \cdots + \frac{h^n}{n!} f^{(n-1)}(t_i, y(t_i)) + \frac{h^{n+1}}{(n+1)!} f^{(n)}(\xi_i, y(\xi_i)). \tag{5.16}$$

The difference-equation method corresponding to Eq. (5.16) is obtained by deleting the remainder term involving ξ_i.

Taylor method of order *n*

$$w_0 = \alpha,$$
$$w_{i+1} = w_i + hT^{(n)}(t_i, w_i), \quad \text{for each } i = 0, 1, \ldots, N-1, \tag{5.17}$$

where

$$T^{(n)}(t_i, w_i) = f(t_i, w_i) + \frac{h}{2} f'(t_i, w_i) + \cdots + \frac{h^{n-1}}{n!} f^{(n-1)}(t_i, w_i).$$

Euler's method is Taylor's method of order one.

Example 1 Apply Taylor's method of orders **(a)** two and **(b)** four with $N = 10$ to the initial-value problem

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

Solution **(a)** For the method of order two we need the first derivative of $f(t, y(t)) = y(t) - t^2 + 1$ with respect to the variable t. Because $y' = y - t^2 + 1$ we have

$$f'(t, y(t)) = \frac{d}{dt}(y - t^2 + 1) = y' - 2t = y - t^2 + 1 - 2t,$$

so

$$T^{(2)}(t_i, w_i) = f(t_i, w_i) + \frac{h}{2} f'(t_i, w_i) = w_i - t_i^2 + 1 + \frac{h}{2}(w_i - t_i^2 + 1 - 2t_i)$$
$$= \left(1 + \frac{h}{2}\right)(w_i - t_i^2 + 1) - ht_i$$

Because $N = 10$ we have $h = 0.2$, and $t_i = 0.2i$ for each $i = 1, 2, \ldots, 10$. Thus the second-order method becomes

$$w_0 = 0.5,$$
$$w_{i+1} = w_i + h\left[\left(1 + \frac{h}{2}\right)(w_i - t_i^2 + 1) - ht_i\right]$$
$$= w_i + 0.2\left[\left(1 + \frac{0.2}{2}\right)(w_i - 0.04i^2 + 1) - 0.04i\right]$$
$$= 1.22w_i - 0.0088i^2 - 0.008i + 0.22.$$

Table 5.3

t_i	Taylor Order 2 w_i	Error $\lvert y(t_i) - w_i \rvert$
0.0	0.500000	0
0.2	0.830000	0.000701
0.4	1.215800	0.001712
0.6	1.652076	0.003135
0.8	2.132333	0.005103
1.0	2.648646	0.007787
1.2	3.191348	0.011407
1.4	3.748645	0.016245
1.6	4.306146	0.022663
1.8	4.846299	0.031122
2.0	5.347684	0.042212

The first two steps give the approximations

$$y(0.2) \approx w_1 = 1.22(0.5) - 0.0088(0)^2 - 0.008(0) + 0.22 = 0.83;$$
$$y(0.4) \approx w_2 = 1.22(0.83) - 0.0088(0.2)^2 - 0.008(0.2) + 0.22 = 1.2158$$

All the approximations and their errors are shown in Table 5.3

(b) For Taylor's method of order four we need the first three derivatives of $f(t, y(t))$ with respect to t. Again using $y' = y - t^2 + 1$ we have

$$f'(t, y(t)) = y - t^2 + 1 - 2t,$$
$$f''(t, y(t)) = \frac{d}{dt}(y - t^2 + 1 - 2t) = y' - 2t - 2$$
$$= y - t^2 + 1 - 2t - 2 = y - t^2 - 2t - 1,$$

and

$$f'''(t, y(t)) = \frac{d}{dt}(y - t^2 - 2t - 1) = y' - 2t - 2 = y - t^2 - 2t - 1,$$

so

$$\begin{aligned} T^{(4)}(t_i, w_i) &= f(t_i, w_i) + \frac{h}{2} f'(t_i, w_i) + \frac{h^2}{6} f''(t_i, w_i) + \frac{h^3}{24} f'''(t_i, w_i) \\ &= w_i - t_i^2 + 1 + \frac{h}{2}(w_i - t_i^2 + 1 - 2t_i) + \frac{h^2}{6}(w_i - t_i^2 - 2t_i - 1) \\ &\quad + \frac{h^3}{24}(w_i - t_i^2 - 2t_i - 1) \\ &= \left(1 + \frac{h}{2} + \frac{h^2}{6} + \frac{h^3}{24}\right)(w_i - t_i^2) - \left(1 + \frac{h}{3} + \frac{h^2}{12}\right)(ht_i) \\ &\quad + 1 + \frac{h}{2} - \frac{h^2}{6} - \frac{h^3}{24}. \end{aligned}$$

Hence Taylor's method of order four is

$$\begin{aligned} w_0 &= 0.5, \\ w_{i+1} &= w_i + h\left[\left(1 + \frac{h}{2} + \frac{h^2}{6} + \frac{h^3}{24}\right)(w_i - t_i^2) - \left(1 + \frac{h}{3} + \frac{h^2}{12}\right)ht_i \right. \\ &\quad \left. + 1 + \frac{h}{2} - \frac{h^2}{6} - \frac{h^3}{24}\right], \end{aligned}$$

for $i = 0, 1, \ldots, N - 1$.

Because $N = 10$ and $h = 0.2$ the method becomes

$$\begin{aligned} w_{i+1} &= w_i + 0.2\left[\left(1 + \frac{0.2}{2} + \frac{0.04}{6} + \frac{0.008}{24}\right)(w_i - 0.04i^2) \right. \\ &\quad \left. - \left(1 + \frac{0.2}{3} + \frac{0.04}{12}\right)(0.04i) + 1 + \frac{0.2}{2} - \frac{0.04}{6} - \frac{0.008}{24}\right] \\ &= 1.2214w_i - 0.008856i^2 - 0.00856i + 0.2186, \end{aligned}$$

for each $i = 0, 1, \ldots, 9$. The first two steps give the approximations

$$y(0.2) \approx w_1 = 1.2214(0.5) - 0.008856(0)^2 - 0.00856(0) + 0.2186 = 0.8293;$$

$$y(0.4) \approx w_2 = 1.2214(0.8293) - 0.008856(0.2)^2 - 0.00856(0.2) + 0.2186 = 1.214091$$

All the approximations and their errors are shown in Table 5.4.

Table 5.4

t_i	Taylor Order 4 w_i	Error $\lvert y(t_i) - w_i \rvert$
0.0	0.500000	0
0.2	0.829300	0.000001
0.4	1.214091	0.000003
0.6	1.648947	0.000006
0.8	2.127240	0.000010
1.0	2.640874	0.000015
1.2	3.179964	0.000023
1.4	3.732432	0.000032
1.6	4.283529	0.000045
1.8	4.815238	0.000062
2.0	5.305555	0.000083

Compare these results with those of Taylor's method of order 2 in Table 5.4 and you will see that the fourth-order results are vastly superior.

The results from Table 5.4 indicate the Taylor's method of order 4 results are quite accurate at the nodes 0.2, 0.4, etc. But suppose we need to determine an approximation to an intermediate point in the table, for example, at $t = 1.25$. If we use linear interpolation on the Taylor method of order four approximations at $t = 1.2$ and $t = 1.4$, we have

$$y(1.25) \approx \left(\frac{1.25 - 1.4}{1.2 - 1.4}\right) 3.1799640 + \left(\frac{1.25 - 1.2}{1.4 - 1.2}\right) 3.7324321 = 3.3180810.$$

Hermite interpolation requires both the value of the function and its derivative at each node. This makes it a natural interpolation method for approximating differential equations since these data are all available.

The true value is $y(1.25) = 3.3173285$, so this approximation has an error of 0.0007525, which is nearly 30 times the average of the approximation errors at 1.2 and 1.4.

We can significantly improve the approximation by using cubic Hermite interpolation. To determine this approximation for $y(1.25)$ requires approximations to $y'(1.2)$ and $y'(1.4)$ as well as approximations to $y(1.2)$ and $y(1.4)$. However, the approximations for $y(1.2)$ and $y(1.4)$ are in the table, and the derivative approximations are available from the differential equation, because $y'(t) = f(t, y(t))$. In our example $y'(t) = y(t) - t^2 + 1$, so

$$y'(1.2) = y(1.2) - (1.2)^2 + 1 \approx 3.1799640 - 1.44 + 1 = 2.7399640$$

and

$$y'(1.4) = y(1.4) - (1.4)^2 + 1 \approx 3.7324327 - 1.96 + 1 = 2.7724321.$$

The divided-difference procedure in Section 3.4 gives the information in Table 5.5. The underlined entries come from the data, and the other entries use the divided-difference formulas.

Table 5.5

1.2	3.1799640			
		2.7399640		
1.2	3.1799640		0.1118825	
		2.7623405		−0.3071225
1.4	3.7324321		0.0504580	
		2.7724321		
1.4	3.7324321			

The cubic Hermite polynomial is

$$y(t) \approx 3.1799640 + (t - 1.2)2.7399640 + (t - 1.2)^2 0.1118825 + (t - 1.2)^2(t - 1.4)(-0.3071225),$$

so

$$y(1.25) \approx 3.1799640 + 0.1369982 + 0.0002797 + 0.0001152 = 3.3173571,$$

a result that is accurate to within 0.0000286. This is about the average of the errors at 1.2 and at 1.4, and only 4% of the error obtained using linear interpolation. This improvement in accuracy certainly justifies the added computation required for the Hermite method. ■

Theorem 5.12 If Taylor's method of order n is used to approximate the solution to

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

with step size h and if $y \in C^{n+1}[a, b]$, then the local truncation error is $O(h^n)$. ■

Proof Note that Eq. (5.16) on page 277 can be rewritten

$$y_{i+1} - y_i - hf(t_i, y_i) - \frac{h^2}{2} f'(t_i, y_i) - \cdots - \frac{h^n}{n!} f^{(n-1)}(t_i, y_i) = \frac{h^{n+1}}{(n+1)!} f^{(n)}(\xi_i, y(\xi_i)),$$

for some ξ_i in (t_i, t_{i+1}). So the local truncation error is

$$\tau_{i+1}(h) = \frac{y_{i+1} - y_i}{h} - T^{(n)}(t_i, y_i) = \frac{h^n}{(n+1)!} f^{(n)}(\xi_i, y(\xi_i)),$$

for each $i = 0, 1, \ldots, N-1$. Since $y \in C^{n+1}[a, b]$, we have $y^{(n+1)}(t) = f^{(n)}(t, y(t))$ bounded on $[a, b]$ and $\tau_i(h) = O(h^n)$, for each $i = 1, 2, \ldots, N$. ■ ■ ■

Taylor's methods are options within the Maple command *InitialValueProblem*. The form and output for Taylor's methods are the same as available under Euler's method, as discussed in Section 5.1. To obtain Taylor's method of order 2 for the problem in Example 1, first load the package and the differential equation.

$with(Student[NumericalAnalysis]) : deq := diff(y(t), t) = y(t) - t^2 + 1$

Then issue

$C := InitialValueProblem(deq, y(0) = 0.5, t = 2, method = taylor, order = 2,$
$numsteps = 10, output = information, digits = 8)$

Maple responds with an array of data similar to that produced with Euler's method. Double clicking on the output will bring up a table that gives the values of t_i, actual solution values $y(t_i)$, the Taylor approximations w_i, and the absolute errors $|y(t_i) - w_i|$. These agree with the values in Table 5.3.

To print the table issue the commands

for k **from** 1 **to** 12 **do**
$print(C[k, 1], C[k, 2], C[k, 3], C[k, 4])$
end do

EXERCISE SET 5.3

1. Use Taylor's method of order two to approximate the solutions for each of the following initial-value problems.

a. $y' = te^{3t} - 2y, \quad 0 \le t \le 1, \quad y(0) = 0,$ with $h = 0.5$

b. $y' = 1 + (t - y)^2, \quad 2 \le t \le 3, \quad y(2) = 1,$ with $h = 0.5$

c. $y' = 1 + y/t, \quad 1 \le t \le 2, \quad y(1) = 2,$ with $h = 0.25$

d. $y' = \cos 2t + \sin 3t, \quad 0 \le t \le 1, \quad y(0) = 1,$ with $h = 0.25$

2. Use Taylor's method of order two to approximate the solutions for each of the following initial-value problems.

a. $y' = e^{t-y}, \quad 0 \le t \le 1, \quad y(0) = 1,$ with $h = 0.5$

b. $y' = \dfrac{1 + t}{1 + y}, \quad 1 \le t \le 2, \quad y(1) = 2,$ with $h = 0.5$

c. $y' = -y + ty^{1/2}, \quad 2 \le t \le 3, \quad y(2) = 2,$ with $h = 0.25$

d. $y' = t^{-2}(\sin 2t - 2ty), \quad 1 \le t \le 2, \quad y(1) = 2,$ with $h = 0.25$

3. Repeat Exercise 1 using Taylor's method of order four.

4. Repeat Exercise 2 using Taylor's method of order four.

5. Use Taylor's method of order two to approximate the solution for each of the following initial-value problems.

a. $y' = y/t - (y/t)^2, \quad 1 \le t \le 1.2, \quad y(1) = 1,$ with $h = 0.1$

b. $y' = \sin t + e^{-t}, \quad 0 \le t \le 1, \quad y(0) = 0,$ with $h = 0.5$

c. $y' = (y^2 + y)/t, \quad 1 \le t \le 3, \quad y(1) = -2,$ with $h = 0.5$

d. $y' = -ty + 4ty^{-1}, \quad 0 \le t \le 1, \quad y(0) = 1,$ with $h = 0.25$

6. Use Taylor's method of order two to approximate the solution for each of the following initial-value problems.

a. $y' = \dfrac{2 - 2ty}{t^2 + 1}, \quad 0 \le t \le 1, \quad y(0) = 1,$ with $h = 0.1$

b. $y' = \dfrac{y^2}{1 + t}, \quad 1 \le t \le 2, \quad y(1) = -(\ln 2)^{-1},$ with $h = 0.1$

c. $y' = (y^2 + y)/t, \quad 1 \le t \le 3, \quad y(1) = -2,$ with $h = 0.2$

d. $y' = -ty + 4t/y, \quad 0 \le t \le 1, \quad y(0) = 1,$ with $h = 0.1$

7. Repeat Exercise 5 using Taylor's method of order four.

8. Repeat Exercise 6 using Taylor's method of order four.

9. Given the initial-value problem

$$y' = \frac{2}{t}y + t^2e^t, \quad 1 \le t \le 2, \quad y(1) = 0,$$

with exact solution $y(t) = t^2(e^t - e)$:

a. Use Taylor's method of order two with $h = 0.1$ to approximate the solution, and compare it with the actual values of y.

b. Use the answers generated in part (a) and linear interpolation to approximate y at the following values, and compare them to the actual values of y.

i. $y(1.04)$ **ii.** $y(1.55)$ **iii.** $y(1.97)$

c. Use Taylor's method of order four with $h = 0.1$ to approximate the solution, and compare it with the actual values of y.

d. Use the answers generated in part (c) and piecewise cubic Hermite interpolation to approximate y at the following values, and compare them to the actual values of y.

i. $y(1.04)$ **ii.** $y(1.55)$ **iii.** $y(1.97)$

10. Given the initial-value problem

$$y' = \frac{1}{t^2} - \frac{y}{t} - y^2, \quad 1 \le t \le 2, \quad y(1) = -1,$$

with exact solution $y(t) = -1/t$:

a. Use Taylor's method of order two with $h = 0.05$ to approximate the solution, and compare it with the actual values of y.

b. Use the answers generated in part (a) and linear interpolation to approximate the following values of y, and compare them to the actual values.

i. $y(1.052)$ **ii.** $y(1.555)$ **iii.** $y(1.978)$

c. Use Taylor's method of order four with $h = 0.05$ to approximate the solution, and compare it with the actual values of y.

d. Use the answers generated in part (c) and piecewise cubic Hermite interpolation to approximate the following values of y, and compare them to the actual values.

i. $y(1.052)$ **ii.** $y(1.555)$ **iii.** $y(1.978)$

11. A projectile of mass $m = 0.11$ kg shot vertically upward with initial velocity $v(0) = 8$ m/s is slowed due to the force of gravity, $F_g = -mg$, and due to air resistance, $F_r = -kv|v|$, where $g = 9.8$ m/s^2 and $k = 0.002$ kg/m. The differential equation for the velocity v is given by

$$mv' = -mg - kv|v|.$$

a. Find the velocity after $0.1, 0.2, \ldots, 1.0$ s.

b. To the nearest tenth of a second, determine when the projectile reaches its maximum height and begins falling.

12. Use the Taylor method of order two with $h = 0.1$ to approximate the solution to

$$y' = 1 + t\sin(ty), \quad 0 \le t \le 2, \quad y(0) = 0.$$

5.4 Runge-Kutta Methods

The Taylor methods outlined in the previous section have the desirable property of high-order local truncation error, but the disadvantage of requiring the computation and evaluation of the derivatives of $f(t, y)$. This is a complicated and time-consuming procedure for most problems, so the Taylor methods are seldom used in practice.

In the later 1800s, Carl Runge (1856–1927) used methods similar to those in this section to derive numerous formulas for approximating the solution to initial-value problems.

Runge-Kutta methods have the high-order local truncation error of the Taylor methods but eliminate the need to compute and evaluate the derivatives of $f(t, y)$. Before presenting the ideas behind their derivation, we need to consider Taylor's Theorem in two variables. The proof of this result can be found in any standard book on advanced calculus (see, for example, [Fu], p. 331).

Theorem 5.13 Suppose that $f(t, y)$ and all its partial derivatives of order less than or equal to $n + 1$ are continuous on $D = \{(t, y) \mid a \le t \le b, c \le y \le d\}$, and let $(t_0, y_0) \in D$. For every $(t, y) \in D$, there exists ξ between t and t_0 and μ between y and y_0 with

$$f(t, y) = P_n(t, y) + R_n(t, y),$$

In 1901, Martin Wilhelm Kutta (1867–1944) generalized the methods that Runge developed in 1895 to incorporate systems of first-order differential equations. These techniques differ slightly from those we currently call Runge-Kutta methods.

where

$$\begin{aligned} P_n(t, y) = f(t_0, y_0) &+ \left[(t - t_0)\frac{\partial f}{\partial t}(t_0, y_0) + (y - y_0)\frac{\partial f}{\partial y}(t_0, y_0)\right] \\ &+ \left[\frac{(t - t_0)^2}{2}\frac{\partial^2 f}{\partial t^2}(t_0, y_0) + (t - t_0)(y - y_0)\frac{\partial^2 f}{\partial t \partial y}(t_0, y_0)\right. \\ &\left. + \frac{(y - y_0)^2}{2}\frac{\partial^2 f}{\partial y^2}(t_0, y_0)\right] + \cdots \\ &+ \left[\frac{1}{n!}\sum_{j=0}^{n}\binom{n}{j}(t - t_0)^{n-j}(y - y_0)^j\frac{\partial^n f}{\partial t^{n-j}\partial y^j}(t_0, y_0)\right] \end{aligned}$$

and

$$R_n(t, y) = \frac{1}{(n + 1)!}\sum_{j=0}^{n+1}\binom{n + 1}{j}(t - t_0)^{n+1-j}(y - y_0)^j\frac{\partial^{n+1} f}{\partial t^{n+1-j}\partial y^j}(\xi, \mu).$$

The function $P_n(t, y)$ is called the ***n*th Taylor polynomial in two variables** for the function f about (t_0, y_0), and $R_n(t, y)$ is the remainder term associated with $P_n(t, y)$. ■

Example 1 Use Maple to determine $P_2(t, y)$, the second Taylor polynomial about $(2, 3)$ for the function

$$f(t, y) = \exp\left[-\frac{(t - 2)^2}{4} - \frac{(y - 3)^2}{4}\right]\cos(2t + y - 7)$$

Solution To determine $P_2(t, y)$ we need the values of f and its first and second partial derivatives at $(2, 3)$. The evaluation of the function is easy

$$f(2, 3) = e^{\left(-0^2/4 - 0^2/4\right)}\cos(4 + 3 - 7) = 1,$$

but the computations involved with the partial derivatives are quite tedious. However, higher dimensional Taylor polynomials are available in the *MultivariateCalculus* subpackage of the *Student* package, which is accessed with the command

with(*Student*[*MultivariateCalculus*])

The first option of the *TaylorApproximation* command is the function, the second specifies the point (t_0, y_0) where the polynomial is centered, and the third specifies the degree of the polynomial. So we issue the command

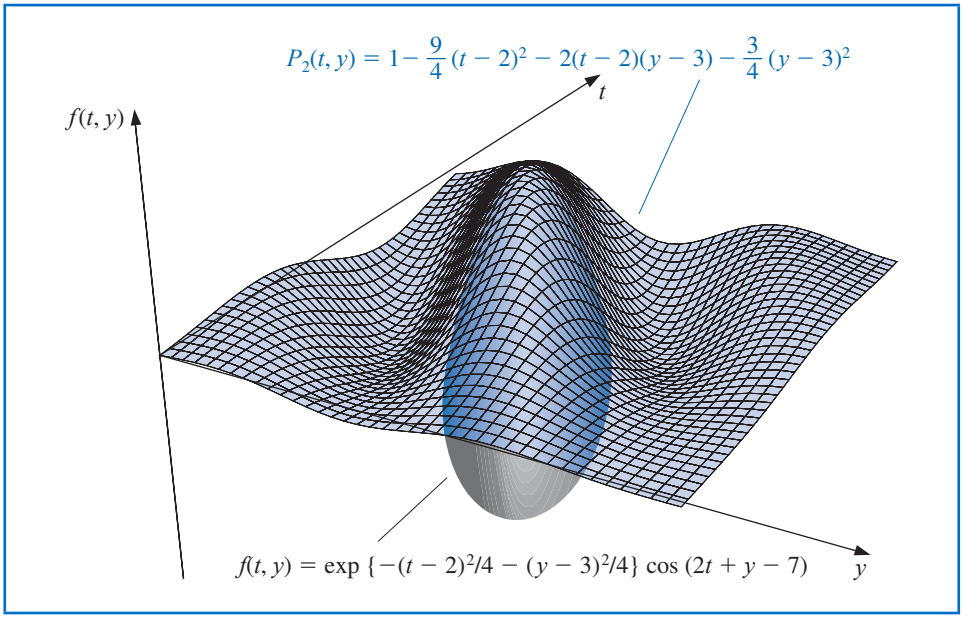
$$TaylorApproximation\left(e^{-\frac{(t-2)^2}{4}-\frac{(y-3)^2}{4}}\cos(2t+y-7),[t,y]=[2,3],2\right)$$

The response from this Maple command is the polynomial

$$1-\frac{9}{4}(t-2)^2-2(t-2)(y-3)-\frac{3}{4}(y-3)^2$$

A plot option is also available by adding a fourth option to the *TaylorApproximation* command in the form *output* = *plot*. The plot in the default form is quite crude, however, because not many points are plotted for the function and the polynomial. A better illustration is seen in Figure 5.5.

Figure 5.5



The final parameter in this command indicates that we want the second multivariate Taylor polynomial, that is, the quadratic polynomial. If this parameter is 2, we get the quadratic polynomial, and if it is 0 or 1, we get the constant polynomial 1, because there are no linear terms. When this parameter is omitted, it defaults to 6 and gives the sixth Taylor polynomial. ■

Runge-Kutta Methods of Order Two

The first step in deriving a Runge-Kutta method is to determine values for a_1, α_1, and β_1 with the property that $a_1 f(t+\alpha_1, y+\beta_1)$ approximates

$$T^{(2)}(t,y) = f(t,y) + \frac{h}{2}f'(t,y),$$

with error no greater than $O(h^2)$, which is same as the order of the local truncation error for the Taylor method of order two. Since

$$f'(t,y) = \frac{df}{dt}(t,y) = \frac{\partial f}{\partial t}(t,y) + \frac{\partial f}{\partial y}(t,y)\cdot y'(t) \quad \text{and} \quad y'(t) = f(t,y),$$

we have

$$T^{(2)}(t,y) = f(t,y) + \frac{h}{2}\frac{\partial f}{\partial t}(t,y) + \frac{h}{2}\frac{\partial f}{\partial y}(t,y)\cdot f(t,y). \tag{5.18}$$

Expanding $f(t+\alpha_1, y+\beta_1)$ in its Taylor polynomial of degree one about (t,y) gives

$$\begin{aligned} a_1 f(t+\alpha_1, y+\beta_1) = a_1 f(t,y) + a_1\alpha_1\frac{\partial f}{\partial t}(t,y) \\ + a_1\beta_1\frac{\partial f}{\partial y}(t,y) + a_1\cdot R_1(t+\alpha_1, y+\beta_1), \end{aligned} \tag{5.19}$$

where

$$R_1(t+\alpha_1, y+\beta_1) = \frac{\alpha_1^2}{2}\frac{\partial^2 f}{\partial t^2}(\xi,\mu) + \alpha_1\beta_1\frac{\partial^2 f}{\partial t\partial y}(\xi,\mu) + \frac{\beta_1^2}{2}\frac{\partial^2 f}{\partial y^2}(\xi,\mu), \tag{5.20}$$

for some ξ between t and $t+\alpha_1$ and μ between y and $y+\beta_1$.

Matching the coefficients of f and its derivatives in Eqs. (5.18) and (5.19) gives the three equations

$$f(t,y):\ a_1 = 1; \quad \frac{\partial f}{\partial t}(t,y):\ a_1\alpha_1 = \frac{h}{2}; \quad \text{and} \quad \frac{\partial f}{\partial y}(t,y):\ a_1\beta_1 = \frac{h}{2}f(t,y).$$

The parameters a_1, α_1, and β_1 are therefore

$$a_1 = 1, \quad \alpha_1 = \frac{h}{2}, \quad \text{and} \quad \beta_1 = \frac{h}{2}f(t,y),$$

so

$$T^{(2)}(t,y) = f\left(t+\frac{h}{2}, y+\frac{h}{2}f(t,y)\right) - R_1\left(t+\frac{h}{2}, y+\frac{h}{2}f(t,y)\right),$$

and from Eq. (5.20),

$$\begin{aligned} R_1\left(t+\frac{h}{2}, y+\frac{h}{2}f(t,y)\right) = \frac{h^2}{8}\frac{\partial^2 f}{\partial t^2}(\xi,\mu) + \frac{h^2}{4}f(t,y)\frac{\partial^2 f}{\partial t\partial y}(\xi,\mu) \\ + \frac{h^2}{8}(f(t,y))^2\frac{\partial^2 f}{\partial y^2}(\xi,\mu). \end{aligned}$$

If all the second-order partial derivatives of f are bounded, then

$$R_1\left(t+\frac{h}{2}, y+\frac{h}{2}f(t,y)\right)$$

is $O(h^2)$. As a consequence:

- The order of error for this new method is the same as that of the Taylor method of order two.

The difference-equation method resulting from replacing $T^{(2)}(t,y)$ in Taylor's method of order two by $f(t+(h/2), y+(h/2)f(t,y))$ is a specific Runge-Kutta method known as the *Midpoint method*.

Midpoint Method

$$w_0 = \alpha,$$
$$w_{i+1} = w_i + hf\left(t_i + \frac{h}{2}, w_i + \frac{h}{2} f(t_i, w_i)\right), \quad \text{for } i = 0, 1, \ldots, N-1.$$

Only three parameters are present in $a_1 f(t + \alpha_1, y + \beta_1)$ and all are needed in the match of $T^{(2)}$. So a more complicated form is required to satisfy the conditions for any of the higher-order Taylor methods.

The most appropriate four-parameter form for approximating

$$T^{(3)}(t, y) = f(t, y) + \frac{h}{2} f'(t, y) + \frac{h^2}{6} f''(t, y)$$

is

$$a_1 f(t, y) + a_2 f(t + \alpha_2, y + \delta_2 f(t, y)); \tag{5.21}$$

and even with this, there is insufficient flexibility to match the term

$$\frac{h^2}{6}\left[\frac{\partial f}{\partial y}(t, y)\right]^2 f(t, y),$$

resulting from the expansion of $(h^2/6) f''(t, y)$. Consequently, the best that can be obtained from using (5.21) are methods with $O(h^2)$ local truncation error.

The fact that (5.21) has four parameters, however, gives a flexibility in their choice, so a number of $O(h^2)$ methods can be derived. One of the most important is the *Modified Euler method*, which corresponds to choosing $a_1 = a_2 = \frac{1}{2}$ and $\alpha_2 = \delta_2 = h$. It has the following difference-equation form.

Modified Euler Method

$$w_0 = \alpha,$$
$$w_{i+1} = w_i + \frac{h}{2}[f(t_i, w_i) + f(t_{i+1}, w_i + hf(t_i, w_i))], \quad \text{for} \quad i = 0, 1, \ldots, N-1.$$

Example 2 Use the Midpoint method and the Modified Euler method with $N = 10$, $h = 0.2$, $t_i = 0.2i$, and $w_0 = 0.5$ to approximate the solution to our usual example,

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

Solution The difference equations produced from the various formulas are

$$\text{Midpoint method:} \quad w_{i+1} = 1.22w_i - 0.0088i^2 - 0.008i + 0.218;$$
$$\text{Modified Euler method:} \quad w_{i+1} = 1.22w_i - 0.0088i^2 - 0.008i + 0.216,$$

for each $i = 0, 1, \ldots, 9$. The first two steps of these methods give

$$\text{Midpoint method:} \quad w_1 = 1.22(0.5) - 0.0088(0)^2 - 0.008(0) + 0.218 = 0.828;$$
$$\text{Modified Euler method:} \quad w_1 = 1.22(0.5) - 0.0088(0)^2 - 0.008(0) + 0.216 = 0.826,$$

and

$$\text{Midpoint method:} \quad w_2 = 1.22(0.828) - 0.0088(0.2)^2 - 0.008(0.2) + 0.218 = 1.21136;$$

$$\text{Modified Euler method:} \quad w_2 = 1.22(0.826) - 0.0088(0.2)^2 - 0.008(0.2) + 0.216 = 1.20692,$$

Table 5.6 lists all the results of the calculations. For this problem, the Midpoint method is superior to the Modified Euler method. ■

Table 5.6

t_i	$y(t_i)$	Midpoint Method	Error	Modified Euler Method	Error
0.0	0.5000000	0.5000000	0	0.5000000	0
0.2	0.8292986	0.8280000	0.0012986	0.8260000	0.0032986
0.4	1.2140877	1.2113600	0.0027277	1.2069200	0.0071677
0.6	1.6489406	1.6446592	0.0042814	1.6372424	0.0116982
0.8	2.1272295	2.1212842	0.0059453	2.1102357	0.0169938
1.0	2.6408591	2.6331668	0.0076923	2.6176876	0.0231715
1.2	3.1799415	3.1704634	0.0094781	3.1495789	0.0303627
1.4	3.7324000	3.7211654	0.0112346	3.6936862	0.0387138
1.6	4.2834838	4.2706218	0.0128620	4.2350972	0.0483866
1.8	4.8151763	4.8009586	0.0142177	4.7556185	0.0595577
2.0	5.3054720	5.2903695	0.0151025	5.2330546	0.0724173

Runge-Kutta methods are also options within the Maple command *InitialValueProblem*. The form and output for Runge-Kutta methods are the same as available under the Euler's and Taylor's methods, as discussed in Sections 5.1 and 5.2.

Higher-Order Runge-Kutta Methods

The term $T^{(3)}(t, y)$ can be approximated with error $O(h^3)$ by an expression of the form

$$f(t + \alpha_1, y + \delta_1 f(t + \alpha_2, y + \delta_2 f(t, y))),$$

involving four parameters, the algebra involved in the determination of $\alpha_1, \delta_1, \alpha_2$, and δ_2 is quite involved. The most common $O(h^3)$ is Heun's method, given by

Karl Heun (1859–1929) was a professor at the Technical University of Karlsruhe. He introduced this technique in a paper published in 1900. [Heu]

$$w_0 = \alpha$$

$$w_{i+1} = w_i + \frac{h}{4}\left(f(t_i, w_i) + 3f\left(t_i + \frac{2h}{3}, w_i + \frac{2h}{3}f\left(t_i + \frac{h}{3}, w_i + \frac{h}{3}f(t_i, w_i)\right)\right)\right),$$

$$\text{for} \quad i = 0, 1, \ldots, N - 1.$$

Illustration Applying Heun's method with $N = 10$, $h = 0.2$, $t_i = 0.2i$, and $w_0 = 0.5$ to approximate the solution to our usual example,

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

gives the values in Table 5.7. Note the decreased error throughout the range over the Midpoint and Modified Euler approximations. □

Table 5.7

t_i	$y(t_i)$	Heun's Method	Error
0.0	0.5000000	0.5000000	0
0.2	0.8292986	0.8292444	0.0000542
0.4	1.2140877	1.2139750	0.0001127
0.6	1.6489406	1.6487659	0.0001747
0.8	2.1272295	2.1269905	0.0002390
1.0	2.6408591	2.6405555	0.0003035
1.2	3.1799415	3.1795763	0.0003653
1.4	3.7324000	3.7319803	0.0004197
1.6	4.2834838	4.2830230	0.0004608
1.8	4.8151763	4.8146966	0.0004797
2.0	5.3054720	5.3050072	0.0004648

Runge-Kutta methods of order three are not generally used. The most common Runge-Kutta method in use is of order four in difference-equation form, is given by the following.

Runge-Kutta Order Four

$$\begin{aligned} w_0 &= \alpha, \\ k_1 &= hf(t_i, w_i), \\ k_2 &= hf\left(t_i + \frac{h}{2}, w_i + \frac{1}{2}k_1\right), \\ k_3 &= hf\left(t_i + \frac{h}{2}, w_i + \frac{1}{2}k_2\right), \\ k_4 &= hf(t_{i+1}, w_i + k_3), \\ w_{i+1} &= w_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4), \end{aligned}$$

for each $i = 0, 1, \ldots, N-1$. This method has local truncation error $O(h^4)$, provided the solution $y(t)$ has five continuous derivatives. We introduce the notation k_1, k_2, k_3, k_4 into the method is to eliminate the need for successive nesting in the second variable of $f(t, y)$. Exercise 32 shows how complicated this nesting becomes.

Algorithm 5.2 implements the Runge-Kutta method of order four.

ALGORITHM
5.2

Runge-Kutta (Order Four)

To approximate the solution of the initial-value problem

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

at $(N + 1)$ equally spaced numbers in the interval $[a, b]$:

INPUT endpoints a, b; integer N; initial condition α.

OUTPUT approximation w to y at the $(N + 1)$ values of t.

Step 1 Set $h = (b - a)/N$;
$t = a$;
$w = \alpha$;
OUTPUT (t, w).

Step 2 For $i = 1, 2, \ldots, N$ do Steps 3–5.

Step 3 Set $K_1 = hf(t, w)$;
$K_2 = hf(t + h/2, w + K_1/2)$;
$K_3 = hf(t + h/2, w + K_2/2)$;
$K_4 = hf(t + h, w + K_3)$.

Step 4 Set $w = w + (K_1 + 2K_2 + 2K_3 + K_4)/6$; (*Compute* w_i.)
$t = a + ih$. (*Compute* t_i.)

Step 5 OUTPUT (t, w).

Step 6 STOP. ■

Example 3 Use the Runge-Kutta method of order four with $h = 0.2$, $N = 10$, and $t_i = 0.2i$ to obtain approximations to the solution of the initial-value problem

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

Solution The approximation to $y(0.2)$ is obtained by

$$w_0 = 0.5$$

$$k_1 = 0.2f(0, 0.5) = 0.2(1.5) = 0.3$$

$$k_2 = 0.2f(0.1, 0.65) = 0.328$$

$$k_3 = 0.2f(0.1, 0.664) = 0.3308$$

$$k_4 = 0.2f(0.2, 0.8308) = 0.35816$$

$$w_1 = 0.5 + \frac{1}{6}(0.3 + 2(0.328) + 2(0.3308) + 0.35816) = 0.8292933.$$

The remaining results and their errors are listed in Table 5.8. ■

Table 5.8

t_i	Exact $y_i = y(t_i)$	Runge-Kutta Order Four w_i	Error $\lvert y_i - w_i \rvert$
0.0	0.5000000	0.5000000	0
0.2	0.8292986	0.8292933	0.0000053
0.4	1.2140877	1.2140762	0.0000114
0.6	1.6489406	1.6489220	0.0000186
0.8	2.1272295	2.1272027	0.0000269
1.0	2.6408591	2.6408227	0.0000364
1.2	3.1799415	3.1798942	0.0000474
1.4	3.7324000	3.7323401	0.0000599
1.6	4.2834838	4.2834095	0.0000743
1.8	4.8151763	4.8150857	0.0000906
2.0	5.3054720	5.3053630	0.0001089

To obtain Runge-Kutta order 4 method results with *InitialValueProblem* use the option *method = rungekutta*, *submethod = rk4*. The results produced from the following call for out standard example problem agree with those in Table 5.6.

$C := InitialValueProblem(deq, y(0) = 0.5, t = 2, method = rungekutta, submethod = rk4, numsteps = 10, output = information, digits = 8)$

Computational Comparisons

The main computational effort in applying the Runge-Kutta methods is the evaluation of f. In the second-order methods, the local truncation error is $O(h^2)$, and the cost is two function evaluations per step. The Runge-Kutta method of order four requires 4 evaluations per step, and the local truncation error is $O(h^4)$. Butcher (see [But] for a summary) has established the relationship between the number of evaluations per step and the order of the local truncation error shown in Table 5.9. This table indicates why the methods of order less than five with smaller step size are used in preference to the higher-order methods using a larger step size.

Table 5.9

Evaluations per step	2	3	4	$5 \le n \le 7$	$8 \le n \le 9$	$10 \le n$
Best possible local truncation error	$O(h^2)$	$O(h^3)$	$O(h^4)$	$O(h^{n-1})$	$O(h^{n-2})$	$O(h^{n-3})$

One measure of comparing the lower-order Runge-Kutta methods is described as follows:

- The Runge-Kutta method of order four requires four evaluations per step, whereas Euler's method requires only one evaluation. Hence if the Runge-Kutta method of order four is to be superior it should give more accurate answers than Euler's method with one-fourth the step size. Similarly, if the Runge-Kutta method of order four is to be superior to the second-order Runge-Kutta methods, which require two evaluations per step, it should give more accuracy with step size h than a second-order method with step size $h/2$.

The following illustrates the superiority of the Runge-Kutta fourth-order method by this measure for the initial-value problem that we have been considering.

Illustration For the problem

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

Euler's method with $h = 0.025$, the Midpoint method with $h = 0.05$, and the Runge-Kutta fourth-order method with $h = 0.1$ are compared at the common mesh points of these methods 0.1, 0.2, 0.3, 0.4, and 0.5. Each of these techniques requires 20 function evaluations to determine the values listed in Table 5.10 to approximate $y(0.5)$. In this example, the fourth-order method is clearly superior. □

Table 5.10

t_i	Exact	Euler $h = 0.025$	Modified Euler $h = 0.05$	Runge-Kutta Order Four $h = 0.1$
0.0	0.5000000	0.5000000	0.5000000	0.5000000
0.1	0.6574145	0.6554982	0.6573085	0.6574144
0.2	0.8292986	0.8253385	0.8290778	0.8292983
0.3	1.0150706	1.0089334	1.0147254	1.0150701
0.4	1.2140877	1.2056345	1.2136079	1.2140869
0.5	1.4256394	1.4147264	1.4250141	1.4256384

EXERCISE SET 5.4

1. Use the Modified Euler method to approximate the solutions to each of the following initial-value problems, and compare the results to the actual values.

a. $y' = te^{3t} - 2y, \quad 0 \le t \le 1, \quad y(0) = 0,$ with $h = 0.5;$ actual solution $y(t) = \frac{1}{5}te^{3t} - \frac{1}{25}e^{3t} + \frac{1}{25}e^{-2t}.$

b. $y' = 1 + (t - y)^2, \quad 2 \le t \le 3, \quad y(2) = 1,$ with $h = 0.5;$ actual solution $y(t) = t + \frac{1}{1-t}.$

c. $y' = 1 + y/t, \quad 1 \le t \le 2, \quad y(1) = 2,$ with $h = 0.25;$ actual solution $y(t) = t \ln t + 2t.$

d. $y' = \cos 2t + \sin 3t, \quad 0 \le t \le 1, \quad y(0) = 1,$ with $h = 0.25;$ actual solution $y(t) = \frac{1}{2}\sin 2t - \frac{1}{3}\cos 3t + \frac{4}{3}.$

2. Use the Modified Euler method to approximate the solutions to each of the following initial-value problems, and compare the results to the actual values.

a. $y' = e^{t-y}, \quad 0 \le t \le 1, \quad y(0) = 1,$ with $h = 0.5;$ actual solution $y(t) = \ln(e^t + e - 1).$

b. $y' = \dfrac{1+t}{1+y}, \quad 1 \le t \le 2, \quad y(1) = 2,$ with $h = 0.5;$ actual solution $y(t) = \sqrt{t^2 + 2t + 6} - 1.$

c. $y' = -y + ty^{1/2}, \quad 2 \le t \le 3, \quad y(2) = 2,$ with $h = 0.25;$ actual solution $y(t) = \left(t - 2 + \sqrt{2}ee^{-t/2}\right)^2.$

d. $y' = t^{-2}(\sin 2t - 2ty), \quad 1 \le t \le 2, \quad y(1) = 2,$ with $h = 0.25;$ actual solution $y(t) = \frac{1}{2}t^{-2}(4 + \cos 2 - \cos 2t).$

3. Use the Modified Euler method to approximate the solutions to each of the following initial-value problems, and compare the results to the actual values.

a. $y' = y/t - (y/t)^2, \quad 1 \le t \le 2, \quad y(1) = 1,$ with $h = 0.1;$ actual solution $y(t) = t/(1 + \ln t).$

b. $y' = 1 + y/t + (y/t)^2, \quad 1 \le t \le 3, \quad y(1) = 0,$ with $h = 0.2;$ actual solution $y(t) = t\tan(\ln t).$

c. $y' = -(y + 1)(y + 3), \quad 0 \le t \le 2, \quad y(0) = -2,$ with $h = 0.2;$ actual solution $y(t) = -3 + 2(1 + e^{-2t})^{-1}.$

d. $y' = -5y + 5t^2 + 2t, \quad 0 \le t \le 1, \quad y(0) = \frac{1}{3},$ with $h = 0.1;$ actual solution $y(t) = t^2 + \frac{1}{3}e^{-5t}.$

4. Use the Modified Euler method to approximate the solutions to each of the following initial-value problems, and compare the results to the actual values.

a. $y' = \dfrac{2 - 2ty}{t^2 + 1}, \quad 0 \le t \le 1, \quad y(0) = 1,$ with $h = 0.1;$ actual solution $y(t) = \dfrac{2t + 1}{t^2 + 1}.$

b. $y' = \dfrac{y^2}{1 + t}, \quad 1 \le t \le 2, \quad y(1) = -(\ln 2)^{-1},$ with $h = 0.1;$ actual solution $y(t) = \dfrac{-1}{\ln(t + 1)}.$

c. $y' = (y^2 + y)/t, \quad 1 \le t \le 3, \quad y(1) = -2,$ with $h = 0.2;$ actual solution $y(t) = \dfrac{2t}{1 - 2t}.$

d. $y' = -ty + 4t/y, \quad 0 \le t \le 1, \quad y(0) = 1,$ with $h = 0.1;$ actual solution $y(t) = \sqrt{4 - 3e^{-t^2}}.$

5. Repeat Exercise 1 using the Midpoint method.

6. Repeat Exercise 2 using the Midpoint method.

7. Repeat Exercise 3 using the Midpoint method.

8. Repeat Exercise 4 using the Midpoint method.

9. Repeat Exercise 1 using Heun's method.

10. Repeat Exercise 2 using Heun's method.

11. Repeat Exercise 3 using Heun's method.

12. Repeat Exercise 4 using Heun's method.

13. Repeat Exercise 1 using the Runge-Kutta method of order four.

14. Repeat Exercise 2 using the Runge-Kutta method of order four.

15. Repeat Exercise 3 using the Runge-Kutta method of order four.

16. Repeat Exercise 4 using the Runge-Kutta method of order four.

17. Use the results of Exercise 3 and linear interpolation to approximate values of $y(t)$, and compare the results to the actual values.

a. $y(1.25)$ and $y(1.93)$ **b.** $y(2.1)$ and $y(2.75)$

c. $y(1.3)$ and $y(1.93)$ **d.** $y(0.54)$ and $y(0.94)$

18. Use the results of Exercise 4 and linear interpolation to approximate values of $y(t)$, and compare the results to the actual values.

a. $y(0.54)$ and $y(0.94)$ **b.** $y(1.25)$ and $y(1.93)$

c. $y(1.3)$ and $y(2.93)$ **d.** $y(0.54)$ and $y(0.94)$

19. Repeat Exercise 17 using the results of Exercise 7.

20. Repeat Exercise 18 using the results of Exercise 8.

21. Repeat Exercise 17 using the results of Exercise 11.

22. Repeat Exercise 18 using the results of Exercise 12.

23. Repeat Exercise 17 using the results of Exercise 15.

24. Repeat Exercise 18 using the results of Exercise 16.

25. Use the results of Exercise 15 and Cubic Hermite interpolation to approximate values of $y(t)$, and compare the approximations to the actual values.

a. $y(1.25)$ and $y(1.93)$ **b.** $y(2.1)$ and $y(2.75)$

c. $y(1.3)$ and $y(1.93)$ **d.** $y(0.54)$ and $y(0.94)$

26. Use the results of Exercise 16 and Cubic Hermite interpolation to approximate values of $y(t)$, and compare the approximations to the actual values.

a. $y(0.54)$ and $y(0.94)$ **b.** $y(1.25)$ and $y(1.93)$

c. $y(1.3)$ and $y(2.93)$ **d.** $y(0.54)$ and $y(0.94)$

27. Show that the Midpoint method and the Modified Euler method give the same approximations to the initial-value problem

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

for any choice of h. Why is this true?

28. Water flows from an inverted conical tank with circular orifice at the rate

$$\frac{dx}{dt} = -0.6\pi r^2 \sqrt{2g}\frac{\sqrt{x}}{A(x)},$$

where r is the radius of the orifice, x is the height of the liquid level from the vertex of the cone, and $A(x)$ is the area of the cross section of the tank x units above the orifice. Suppose $r = 0.1$ ft, $g = 32.1$ ft/s^2, and the tank has an initial water level of 8 ft and initial volume of $512(\pi/3)$ ft^3. Use the Runge-Kutta method of order four to find the following.

a. The water level after 10 min with $h = 20$ s

b. When the tank will be empty, to within 1 min.

29. The irreversible chemical reaction in which two molecules of solid potassium dichromate ($K_2Cr_2O_7$), two molecules of water (H_2O), and three atoms of solid sulfur (S) combine to yield three molecules of the gas sulfur dioxide (SO_2), four molecules of solid potassium hydroxide (KOH), and two molecules of solid chromic oxide (Cr_2O_3) can be represented symbolically by the *stoichiometric equation*:

$$2K_2Cr_2O_7 + 2H_2O + 3S \longrightarrow 4KOH + 2Cr_2O_3 + 3SO_2.$$

If n_1 molecules of $K_2Cr_2O_7$, n_2 molecules of H_2O, and n_3 molecules of S are originally available, the following differential equation describes the amount $x(t)$ of KOH after time t:

$$\frac{dx}{dt} = k\left(n_1 - \frac{x}{2}\right)^2\left(n_2 - \frac{x}{2}\right)^2\left(n_3 - \frac{3x}{4}\right)^3,$$

where k is the velocity constant of the reaction. If $k = 6.22 \times 10^{-19}$, $n_1 = n_2 = 2 \times 10^3$, and $n_3 = 3 \times 10^3$, use the Runge-Kutta method of order four to determine how many units of potassium hydroxide will have been formed after 0.2 s?

30. Show that the difference method

$$w_0 = \alpha,$$
$$w_{i+1} = w_i + a_1 f(t_i, w_i) + a_2 f(t_i + \alpha_2, w_1 + \delta_2 f(t_i, w_i)),$$

for each $i = 0, 1, \ldots, N-1$, cannot have local truncation error $O(h^3)$ for any choice of constants a_1, a_2, α_2, and δ_2.

31. Show that Heun's method can be expressed in difference form, similar to that of the Runge-Kutta method of order four, as

$$w_0 = \alpha,$$
$$k_1 = hf(t_i, w_i),$$
$$k_2 = hf\left(t_i + \frac{h}{3}, w_i + \frac{1}{3}k_1\right),$$
$$k_3 = hf\left(t_i + \frac{2h}{3}, w_i + \frac{2}{3}k_2\right),$$
$$w_{i+1} = w_i + \frac{1}{4}(k_1 + 3k_3),$$

for each $i = 0, 1, \ldots, N-1$.

32. The Runge-Kutta method of order four can be written in the form

$$w_0 = \alpha,$$
$$w_{i+1} = w_i + \frac{h}{6}f(t_i, w_i) + \frac{h}{3}f(t_i + \alpha_1 h, w_i + \delta_1 h f(t_i, w_i))$$
$$+ \frac{h}{3}f(t_i + \alpha_2 h, w_i + \delta_2 h f(t_i + \gamma_2 h, w_i + \gamma_3 h f(t_i, w_i)))$$
$$+ \frac{h}{6}f(t_i + \alpha_3 h, w_i + \delta_3 h f(t_i + \gamma_4 h, w_i + \gamma_5 h f(t_i + \gamma_6 h, w_i + \gamma_7 h f(t_i, w_i)))).$$

Find the values of the constants

$$\alpha_1,\ \alpha_2,\ \alpha_3,\ \delta_1,\ \delta_2,\ \delta_3,\ \gamma_2,\ \gamma_3,\ \gamma_4,\ \gamma_5,\ \gamma_6, \text{ and } \gamma_7.$$

5.5 Error Control and the Runge-Kutta-Fehlberg Method

In Section 4.6 we saw that the appropriate use of varying step sizes for integral approximations produced efficient methods. In itself, this might not be sufficient to favor these methods due to the increased complication of applying them. However, they have another feature

You might like to review the Adaptive Quadrature material in Section 4.6 before considering this material.

that makes them worthwhile. They incorporate in the step-size procedure an estimate of the truncation error that does not require the approximation of the higher derivatives of the function. These methods are called *adaptive* because they adapt the number and position of the nodes used in the approximation to ensure that the truncation error is kept within a specified bound.

There is a close connection between the problem of approximating the value of a definite integral and that of approximating the solution to an initial-value problem. It is not surprising, then, that there are adaptive methods for approximating the solutions to initial-value problems and that these methods are not only efficient, but also incorporate the control of error.

Any one-step method for approximating the solution, $y(t)$, of the initial-value problem

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

can be expressed in the form

$$w_{i+1} = w_i + h_i \phi(t_i, w_i, h_i), \quad \text{for } i = 0, 1, \ldots, N-1,$$

for some function ϕ.

An ideal difference-equation method

$$w_{i+1} = w_i + h_i \phi(t_i, w_i, h_i), \quad i = 0, 1, \ldots, N-1,$$

for approximating the solution, $y(t)$, to the initial-value problem

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

would have the property that, given a tolerance $\varepsilon > 0$, a minimal number of mesh points could be used to ensure that the global error, $|y(t_i) - w_i|$, did not exceed ε for any $i = 0, 1, \ldots, N$. Having a minimal number of mesh points and also controlling the global error of a difference method is, not surprisingly, inconsistent with the points being equally spaced in the interval. In this section we examine techniques used to control the error of a difference-equation method in an efficient manner by the appropriate choice of mesh points.

Although we cannot generally determine the global error of a method, we will see in Section 5.10 that there is a close connection between the local truncation error and the global error. By using methods of differing order we can predict the local truncation error and, using this prediction, choose a step size that will keep it and the global error in check.

To illustrate the technique, suppose that we have two approximation techniques. The first is obtained from an nth-order Taylor method of the form

$$y(t_{i+1}) = y(t_i) + h\phi(t_i, y(t_i), h) + O(h^{n+1}),$$

and produces approximations with local truncation error $\tau_{i+1}(h) = O(h^n)$. It is given by

$$w_0 = \alpha$$
$$w_{i+1} = w_i + h\phi(t_i, w_i, h), \quad \text{for } i > 0.$$

In general, the method is generated by applying a Runge-Kutta modification to the Taylor method, but the specific derivation is unimportant.

The second method is similar but one order higher; it comes from an $(n+1)$st-order Taylor method of the form

$$y(t_{i+1}) = y(t_i) + h\tilde{\phi}(t_i, y(t_i), h) + O(h^{n+2}),$$

and produces approximations with local truncation error $\tilde{\tau}_{i+1}(h) = O(h^{n+1})$. It is given by

$$\tilde{w}_0 = \alpha$$

$$\tilde{w}_{i+1} = \tilde{w}_i + h\tilde{\phi}(t_i, \tilde{w}_i, h), \quad \text{for } i > 0.$$

We first make the assumption that $w_i \approx y(t_i) \approx \tilde{w}_i$ and choose a fixed step size h to generate the approximations w_{i+1} and $\tilde{w}_{i+1}$ to $y(t_{i+1})$. Then

$$\begin{aligned}\tau_{i+1}(h) &= \frac{y(t_{i+1}) - y(t_i)}{h} - \phi(t_i, y(t_i), h)\\ &= \frac{y(t_{i+1}) - w_i}{h} - \phi(t_i, w_i, h)\\ &= \frac{y(t_{i+1}) - [w_i + h\phi(t_i, w_i, h)]}{h}\\ &= \frac{1}{h}(y(t_{i+1}) - w_{i+1}).\end{aligned}$$

In a similar manner, we have

$$\tilde{\tau}_{i+1}(h) = \frac{1}{h}(y(t_{i+1}) - \tilde{w}_{i+1}).$$

As a consequence, we have

$$\begin{aligned}\tau_{i+1}(h) &= \frac{1}{h}(y(t_{i+1}) - w_{i+1})\\ &= \frac{1}{h}[(y(t_{i+1}) - \tilde{w}_{i+1}) + (\tilde{w}_{i+1} - w_{i+1})]\\ &= \tilde{\tau}_{i+1}(h) + \frac{1}{h}(\tilde{w}_{i+1} - w_{i+1}).\end{aligned}$$

But $\tau_{i+1}(h)$ is $O(h^n)$ and $\tilde{\tau}_{i+1}(h)$ is $O(h^{n+1})$, so the significant portion of $\tau_{i+1}(h)$ must come from

$$\frac{1}{h}(\tilde{w}_{i+1} - w_{i+1}).$$

This gives us an easily computed approximation for the local truncation error of the $O(h^n)$ method:

$$\tau_{i+1}(h) \approx \frac{1}{h}(\tilde{w}_{i+1} - w_{i+1}).$$

The object, however, is not simply to estimate the local truncation error but to adjust the step size to keep it within a specified bound. To do this we now assume that since $\tau_{i+1}(h)$ is $O(h^n)$, a number K, independent of h, exists with

$$\tau_{i+1}(h) \approx Kh^n.$$

Then the local truncation error produced by applying the nth-order method with a new step size qh can be estimated using the original approximations w_{i+1} and $\tilde{w}_{i+1}$:

$$\tau_{i+1}(qh) \approx K(qh)^n = q^n(Kh^n) \approx q^n\tau_{i+1}(h) \approx \frac{q^n}{h}(\tilde{w}_{i+1} - w_{i+1}).$$

To bound $\tau_{i+1}(qh)$ by ε, we choose q so that

$$\frac{q^n}{h}|\tilde{w}_{i+1} - w_{i+1}| \approx |\tau_{i+1}(qh)| \leq \varepsilon;$$

that is, so that

$$q \le \left(\frac{\varepsilon h}{|\tilde{w}_{i+1} - w_{i+1}|} \right)^{1/n}. \tag{5.22}$$

Runge-Kutta-Fehlberg Method

Erwin Fehlberg developed this and other error control techniques while working for the NASA facility in Huntsville, Alabama during the 1960s. He received the Exceptional Scientific Achievement Medal from NASA in 1969.

One popular technique that uses Inequality (5.22) for error control is the **Runge-Kutta-Fehlberg method**. (See [Fe].) This technique uses a Runge-Kutta method with local truncation error of order five,

$$\tilde{w}_{i+1} = w_i + \frac{16}{135}k_1 + \frac{6656}{12825}k_3 + \frac{28561}{56430}k_4 - \frac{9}{50}k_5 + \frac{2}{55}k_6,$$

to estimate the local error in a Runge-Kutta method of order four given by

$$w_{i+1} = w_i + \frac{25}{216}k_1 + \frac{1408}{2565}k_3 + \frac{2197}{4104}k_4 - \frac{1}{5}k_5,$$

where the coefficient equations are

$$\begin{aligned}
k_1 &= h f(t_i, w_i),\\
k_2 &= h f\left(t_i + \frac{h}{4}, w_i + \frac{1}{4}k_1\right),\\
k_3 &= h f\left(t_i + \frac{3h}{8}, w_i + \frac{3}{32}k_1 + \frac{9}{32}k_2\right),\\
k_4 &= h f\left(t_i + \frac{12h}{13}, w_i + \frac{1932}{2197}k_1 - \frac{7200}{2197}k_2 + \frac{7296}{2197}k_3\right),\\
k_5 &= h f\left(t_i + h, w_i + \frac{439}{216}k_1 - 8k_2 + \frac{3680}{513}k_3 - \frac{845}{4104}k_4\right),\\
k_6 &= h f\left(t_i + \frac{h}{2}, w_i - \frac{8}{27}k_1 + 2k_2 - \frac{3544}{2565}k_3 + \frac{1859}{4104}k_4 - \frac{11}{40}k_5\right).
\end{aligned}$$

An advantage to this method is that only six evaluations of f are required per step. Arbitrary Runge-Kutta methods of orders four and five used together (see Table 5.9 on page 290) require at least four evaluations of f for the fourth-order method and an additional six for the fifth-order method, for a total of at least ten function evaluations. So the Runge-Kutta-Fehlberg method has at least a 40% decrease in the number of function evaluations over the use of a pair of arbitrary fourth- and fifth-order methods.

In the error-control theory, an initial value of h at the ith step is used to find the first values of w_{i+1} and $\tilde{w}_{i+1}$, which leads to the determination of q for that step, and then the calculations were repeated. This procedure requires twice the number of function evaluations per step as without the error control. In practice, the value of q to be used is chosen somewhat differently in order to make the increased function-evaluation cost worthwhile. The value of q determined at the ith step is used for two purposes:

- When $q < 1$: to reject the initial choice of h at the ith step and repeat the calculations using qh, and
- When $q \ge 1$: to accept the computed value at the ith step using the step size h, but change the step size to qh for the $(i + 1)$st step.

Because of the penalty in terms of function evaluations that must be paid if the steps are repeated, q tends to be chosen conservatively. In fact, for the Runge-Kutta-Fehlberg method with $n = 4$, a common choice is

$$q = \left(\frac{\varepsilon h}{2|\tilde{w}_{i+1} - w_{i+1}|}\right)^{1/4} = 0.84\left(\frac{\varepsilon h}{|\tilde{w}_{i+1} - w_{i+1}|}\right)^{1/4}.$$

In Algorithm 5.3 for the Runge-Kutta-Fehlberg method, Step 9 is added to eliminate large modifications in step size. This is done to avoid spending too much time with small step sizes in regions with irregularities in the derivatives of y, and to avoid large step sizes, which can result in skipping sensitive regions between the steps. The step-size increase procedure could be omitted completely from the algorithm, and the step-size decrease procedure used only when needed to bring the error under control.

ALGORITHM
5.3

Runge-Kutta-Fehlberg

To approximate the solution of the initial-value problem

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

with local truncation error within a given tolerance:

INPUT endpoints a, b; initial condition α; tolerance TOL; maximum step size $hmax$; minimum step size $hmin$.

OUTPUT t, w, h where w approximates $y(t)$ and the step size h was used, or a message that the minimum step size was exceeded.

Step 1 Set $t = a$;
$w = \alpha$;
$h = hmax$;
$FLAG = 1$;
OUTPUT (t, w).

Step 2 While $(FLAG = 1)$ do Steps 3–11.

Step 3 Set $K_1 = hf\,(t, w)$;

$$K_2 = hf\left(t + \tfrac{1}{4}h, w + \tfrac{1}{4}K_1\right);$$
$$K_3 = hf\left(t + \tfrac{3}{8}h, w + \tfrac{3}{32}K_1 + \tfrac{9}{32}K_2\right);$$
$$K_4 = hf\left(t + \tfrac{12}{13}h, w + \tfrac{1932}{2197}K_1 - \tfrac{7200}{2197}K_2 + \tfrac{7296}{2197}K_3\right);$$
$$K_5 = hf\left(t + h, w + \tfrac{439}{216}K_1 - 8K_2 + \tfrac{3680}{513}K_3 - \tfrac{845}{4104}K_4\right);$$
$$K_6 = hf\left(t + \tfrac{1}{2}h, w - \tfrac{8}{27}K_1 + 2K_2 - \tfrac{3544}{2565}K_3 + \tfrac{1859}{4104}K_4 - \tfrac{11}{40}K_5\right).$$

Step 4 Set $R = \frac{1}{h}\left|\frac{1}{360}K_1 - \frac{128}{4275}K_3 - \frac{2197}{75240}K_4 + \frac{1}{50}K_5 + \frac{2}{55}K_6\right|$.
(*Note:* $R = \frac{1}{h}|\tilde{w}_{i+1} - w_{i+1}|$.)

Step 5 If $R \le TOL$ then do Steps 6 and 7.

Step 6 Set $t = t + h$; (*Approximation accepted.*)

$$w = w + \tfrac{25}{216}K_1 + \tfrac{1408}{2565}K_3 + \tfrac{2197}{4104}K_4 - \tfrac{1}{5}K_5.$$

Step 7 OUTPUT (t, w, h).

Step 8 Set $\delta = 0.84(TOL/R)^{1/4}$.

Step 9 If $\delta \le 0.1$ then set $h = 0.1h$
else if $\delta \ge 4$ then set $h = 4h$
else set $h = \delta h$. (*Calculate new h.*)

Step 10 If $h > hmax$ then set $h = hmax$.

Step 11 If $t \ge b$ then set $FLAG = 0$
else if $t + h > b$ then set $h = b - t$
else if $h < hmin$ then
set $FLAG = 0$;
OUTPUT ('*minimum h exceeded*').
(*Procedure completed unsuccessfully.*)

Step 12 (*The procedure is complete.*)
STOP. ■

Example 1 Use the Runge-Kutta-Fehlberg method with a tolerance $TOL = 10^{-5}$, a maximum step size $hmax = 0.25$, and a minimum step size $hmin = 0.01$ to approximate the solution to the initial-value problem

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

and compare the results with the exact solution $y(t) = (t + 1)^2 - 0.5e^t$.

Solution We will work through the first step of the calculations and then apply Algorithm 5.3 to determine the remaining results. The initial condition gives $t_0 = 0$ and $w_0 = 0.5$. To determine w_1 using $h = 0.25$, the maximum allowable stepsize, we compute

$$k_1 = hf(t_0, w_0) = 0.25\left(0.5 - 0^2 + 1\right) = 0.375;$$

$$k_2 = hf\left(t_0 + \frac{1}{4}h, w_0 + \frac{1}{4}k_1\right) = 0.25\left(\frac{1}{4}0.25, 0.5 + \frac{1}{4}0.375\right) = 0.3974609;$$

$$k_3 = hf\left(t_0 + \frac{3}{8}h, w_0 + \frac{3}{32}k_1 + \frac{9}{32}k_2\right)$$

$$= 0.25\left(0.09375, 0.5 + \frac{3}{32}0.375 + \frac{9}{32}0.3974609\right) = 0.4095383;$$

$$k_4 = hf\left(t_0 + \frac{12}{13}h, w_0 + \frac{1932}{2197}k_1 - \frac{7200}{2197}k_2 + \frac{7296}{2197}k_3\right)$$

$$= 0.25\left(0.2307692, 0.5 + \frac{1932}{2197}0.375 - \frac{7200}{2197}0.3974609 + \frac{7296}{2197}0.4095383\right)$$

$$= 0.4584971;$$

$$k_5 = hf\left(t_0 + h, w_0 + \frac{439}{216}k_1 - 8k_2 + \frac{3680}{513}k_3 - \frac{845}{4104}k_4\right)$$

$$= 0.25\left(0.25, 0.5 + \frac{439}{216}0.375 - 8(0.3974609) + \frac{3680}{513}0.4095383 - \frac{845}{4104}0.4584971\right)$$

$$= 0.4658452;$$

$$k_6 = hf\left(t_0 + \frac{1}{2}h, w_0 - \frac{8}{27}k_1 + 2k_2 - \frac{3544}{2565}k_3 + \frac{1859}{4104}k_4 - \frac{11}{40}k_5\right)$$
$$= 0.25\left(0.125, 0.5 - \frac{8}{27}0.375 + 2(0.3974609) - \frac{3544}{2565}0.4095383 + \frac{1859}{4104}0.4584971 - \frac{11}{40}0.4658452\right)$$
$$= 0.4204789.$$

The two approximations to $y(0.25)$ are then found to be

$$\tilde{w}_1 = w_0 + \frac{16}{135}k_1 + \frac{6656}{12825}k_3 + \frac{28561}{56430}k_4 - \frac{9}{50}k_5 + \frac{2}{55}k_6$$
$$= 0.5 + \frac{16}{135}0.375 + \frac{6656}{12825}0.4095383 + \frac{28561}{56430}0.4584971 - \frac{9}{50}0.4658452 + \frac{2}{55}0.4204789$$
$$= 0.9204870,$$

and

$$w_1 = w_0 + \frac{25}{216}k_1 + \frac{1408}{2565}k_3 + \frac{2197}{4104}k_4 - \frac{1}{5}k_5$$
$$= 0.5 + \frac{25}{216}0.375 + \frac{1408}{2565}0.4095383 + \frac{2197}{4104}0.4584971 - \frac{1}{5}0.4658452$$
$$= 0.9204886.$$

This also implies that

$$R = \frac{1}{0.25}\left|\frac{1}{360}k_1 - \frac{128}{4275}k_3 - \frac{2197}{75240}k_4 + \frac{1}{50}k_5 + \frac{2}{55}k_6\right|$$
$$= 4\left|\frac{1}{360}0.375 - \frac{128}{4275}0.4095383 - \frac{2197}{75240}0.4584971 + \frac{1}{50}0.4658452 + \frac{2}{55}0.4204789\right|$$
$$= 0.00000621388,$$

and

$$q = 0.84\left(\frac{\varepsilon}{R}\right)^{1/4} = 0.84\left(\frac{0.00001}{0.00000621388}\right)^{1/4} = 0.9461033291.$$

Since $q < 1$ we can accept the approximation 0.9204886 for $y(0.25)$ but we should adjust the step size for the next iteration to $h = 0.9461033291(0.25) \approx 0.2365258$. However, only the leading 5 digits of this result would be expected to be accurate because R has only about 5 digits of accuracy. Because we are effectively subtracting the nearly equal numbers w_i and $\tilde{w}_i$ when we compute R, there is a good likelihood of round-off error. This is an additional reason for being conservative when computing q.

The results from the algorithm are shown in Table 5.11. Increased accuracy has been used to ensure that the calculations are accurate to all listed places. The last two columns in Table 5.11 show the results of the fifth-order method. For small values of t, the error is less than the error in the fourth-order method, but the error exceeds that of the fourth-order method when t increases. ■

Table 5.11

		RKF-4				RKF-5	
t_i	$y_i = y(t_i)$	w_i	h_i	R_i	$\lvert y_i - w_i\rvert$	$\hat{w}_i$	$\lvert y_i - \hat{w}_i\rvert$
0	0.5	0.5			0.5		
0.2500000	0.9204873	0.9204886	0.2500000	6.2×10^{-6}	1.3×10^{-6}	0.9204870	2.424×10^{-7}
0.4865522	1.3964884	1.3964910	0.2365522	4.5×10^{-6}	2.6×10^{-6}	1.3964900	1.510×10^{-6}
0.7293332	1.9537446	1.9537488	0.2427810	4.3×10^{-6}	4.2×10^{-6}	1.9537477	3.136×10^{-6}
0.9793332	2.5864198	2.5864260	0.2500000	3.8×10^{-6}	6.2×10^{-6}	2.5864251	5.242×10^{-6}
1.2293332	3.2604520	3.2604605	0.2500000	2.4×10^{-6}	8.5×10^{-6}	3.2604599	7.895×10^{-6}
1.4793332	3.9520844	3.9520955	0.2500000	7×10^{-7}	1.11×10^{-5}	3.9520954	1.096×10^{-5}
1.7293332	4.6308127	4.6308268	0.2500000	1.5×10^{-6}	1.41×10^{-5}	4.6308272	1.446×10^{-5}
1.9793332	5.2574687	5.2574861	0.2500000	4.3×10^{-6}	1.73×10^{-5}	5.2574871	1.839×10^{-5}
2.0000000	5.3054720	5.3054896	0.0206668		1.77×10^{-5}	5.3054896	1.768×10^{-5}

An implementation of the Runge-Kutta-Fehlberg method is also available in Maple using the *InitialValueProblem* command. However, it differs from our presentation because it does not require the specification of a tolerance for the solution. For our example problem it is called with

$C := InitialValueProblem(deq, y(0) = 0.5, t = 2, method = rungekutta, submethod = rkf, numsteps = 10, output = information, digits = 8)$

As usual, the information is placed in a table that is accessed by double clicking on the output. The results can be printed in the method outlined in precious sections.

EXERCISE SET 5.5

1. Use the Runge-Kutta-Fehlberg method with tolerance $TOL = 10^{-4}$, $hmax = 0.25$, and $hmin = 0.05$ to approximate the solutions to the following initial-value problems. Compare the results to the actual values.

a. $y' = te^{3t} - 2y, \quad 0 \le t \le 1, \quad y(0) = 0;$ actual solution $y(t) = \frac{1}{5}te^{3t} - \frac{1}{25}e^{3t} + \frac{1}{25}e^{-2t}$.

b. $y' = 1 + (t - y)^2, \quad 2 \le t \le 3, \quad y(2) = 1;$ actual solution $y(t) = t + 1/(1 - t)$.

c. $y' = 1 + y/t, \quad 1 \le t \le 2, \quad y(1) = 2;$ actual solution $y(t) = t \ln t + 2t$.

d. $y' = \cos 2t + \sin 3t, \quad 0 \le t \le 1, \quad y(0) = 1;$ actual solution $y(t) = \frac{1}{2}\sin 2t - \frac{1}{3}\cos 3t + \frac{4}{3}$.

2. Use the Runge-Kutta Fehlberg Algorithm with tolerance $TOL = 10^{-4}$ to approximate the solution to the following initial-value problems.

a. $y' = (y/t)^2 + y/t, \quad 1 \le t \le 1.2, \quad y(1) = 1,$ with $hmax = 0.05$ and $hmin = 0.02$.

b. $y' = \sin t + e^{-t}, \quad 0 \le t \le 1, \quad y(0) = 0,$ with $hmax = 0.25$ and $hmin = 0.02$.

c. $y' = (y^2 + y)/t, \quad 1 \le t \le 3, \quad y(1) = -2,$ with $hmax = 0.5$ and $hmin = 0.02$.

d. $y' = t^2, \quad 0 \le t \le 2, \quad y(0) = 0,$ with $hmax = 0.5$ and $hmin = 0.02$.

3. Use the Runge-Kutta-Fehlberg method with tolerance $TOL = 10^{-6}$, $hmax = 0.5$, and $hmin = 0.05$ to approximate the solutions to the following initial-value problems. Compare the results to the actual values.

a. $y' = y/t - (y/t)^2, \quad 1 \le t \le 4, \quad y(1) = 1;$ actual solution $y(t) = t/(1 + \ln t)$.

b. $y' = 1 + y/t + (y/t)^2, \quad 1 \le t \le 3, \quad y(1) = 0;$ actual solution $y(t) = t\tan(\ln t)$.

c. $y' = -(y + 1)(y + 3), \quad 0 \le t \le 3, \quad y(0) = -2;$ actual solution $y(t) = -3 + 2(1 + e^{-2t})^{-1}$.

d. $y' = (t + 2t^3)y^3 - ty, \quad 0 \le t \le 2, \quad y(0) = \frac{1}{3};$ actual solution $y(t) = (3 + 2t^2 + 6e^{t^2})^{-1/2}$.

4. The Runge-Kutta-Verner method (see [Ve]) is based on the formulas

$$w_{i+1} = w_i + \frac{13}{160}k_1 + \frac{2375}{5984}k_3 + \frac{5}{16}k_4 + \frac{12}{85}k_5 + \frac{3}{44}k_6 \quad \text{and}$$
$$\tilde{w}_{i+1} = w_i + \frac{3}{40}k_1 + \frac{875}{2244}k_3 + \frac{23}{72}k_4 + \frac{264}{1955}k_5 + \frac{125}{11592}k_7 + \frac{43}{616}k_8,$$

where

$$k_1 = hf(t_i, w_i),$$
$$k_2 = hf\left(t_i + \frac{h}{6}, w_i + \frac{1}{6}k_1\right),$$
$$k_3 = hf\left(t_i + \frac{4h}{15}, w_i + \frac{4}{75}k_1 + \frac{16}{75}k_2\right),$$
$$k_4 = hf\left(t_i + \frac{2h}{3}, w_i + \frac{5}{6}k_1 - \frac{8}{3}k_2 + \frac{5}{2}k_3\right),$$
$$k_5 = hf\left(t_i + \frac{5h}{6}, w_i - \frac{165}{64}k_1 + \frac{55}{6}k_2 - \frac{425}{64}k_3 + \frac{85}{96}k_4\right),$$
$$k_6 = hf\left(t_i + h, w_i + \frac{12}{5}k_1 - 8k_2 + \frac{4015}{612}k_3 - \frac{11}{36}k_4 + \frac{88}{255}k_5\right),$$
$$k_7 = hf\left(t_i + \frac{h}{15}, w_i - \frac{8263}{15000}k_1 + \frac{124}{75}k_2 - \frac{643}{680}k_3 - \frac{81}{250}k_4 + \frac{2484}{10625}k_5\right),$$
$$k_8 = hf\left(t_i + h, w_i + \frac{3501}{1720}k_1 - \frac{300}{43}k_2 + \frac{297275}{52632}k_3 - \frac{319}{2322}k_4 + \frac{24068}{84065}k_5 + \frac{3850}{26703}k_7\right).$$

The sixth-order method $\tilde{w}_{i+1}$ is used to estimate the error in the fifth-order method w_{i+1}. Construct an algorithm similar to the Runge-Kutta-Fehlberg Algorithm, and repeat Exercise 3 using this new method.

5. In the theory of the spread of contagious disease (see [Ba1] or [Ba2]), a relatively elementary differential equation can be used to predict the number of infective individuals in the population at any time, provided appropriate simplification assumptions are made. In particular, let us assume that all individuals in a fixed population have an equally likely chance of being infected and once infected remain in that state. Suppose $x(t)$ denotes the number of susceptible individuals at time t and $y(t)$ denotes the number of infectives. It is reasonable to assume that the rate at which the number of infectives changes is proportional to the product of $x(t)$ and $y(t)$ because the rate depends on both the number of infectives and the number of susceptibles present at that time. If the population is large enough to assume that $x(t)$ and $y(t)$ are continuous variables, the problem can be expressed

$$y'(t) = kx(t)y(t),$$

where k is a constant and $x(t) + y(t) = m$, the total population. This equation can be rewritten involving only $y(t)$ as

$$y'(t) = k(m - y(t))y(t).$$

a. Assuming that $m = 100{,}000$, $y(0) = 1000$, $k = 2 \times 10^{-6}$, and that time is measured in days, find an approximation to the number of infective individuals at the end of 30 days.

b. The differential equation in part (a) is called a *Bernoulli equation* and it can be transformed into a linear differential equation in $u(t) = (y(t))^{-1}$. Use this technique to find the exact solution to the equation, under the same assumptions as in part (a), and compare the true value of $y(t)$ to the approximation given there. What is $\lim_{t\to\infty} y(t)$? Does this agree with your intuition?

6. In the previous exercise, all infected individuals remained in the population to spread the disease. A more realistic proposal is to introduce a third variable $z(t)$ to represent the number of individuals

who are removed from the affected population at a given time t by isolation, recovery and consequent immunity, or death. This quite naturally complicates the problem, but it can be shown (see [Ba2]) that an approximate solution can be given in the form

$$x(t) = x(0)e^{-(k_1/k_2)z(t)} \quad \text{and} \quad y(t) = m - x(t) - z(t),$$

where k_1 is the infective rate, k_2 is the removal rate, and $z(t)$ is determined from the differential equation

$$z'(t) = k_2\left(m - z(t) - x(0)e^{-(k_1/k_2)z(t)}\right).$$

The authors are not aware of any technique for solving this problem directly, so a numerical procedure must be applied. Find an approximation to $z(30)$, $y(30)$, and $x(30)$, assuming that $m = 100{,}000$, $x(0) = 99{,}000$, $k_1 = 2 \times 10^{-6}$, and $k_2 = 10^{-4}$.

5.6 Multistep Methods

The methods discussed to this point in the chapter are called **one-step methods** because the approximation for the mesh point t_{i+1} involves information from only one of the previous mesh points, t_i. Although these methods might use function evaluation information at points between t_i and t_{i+1}, they do not retain that information for direct use in future approximations. All the information used by these methods is obtained within the subinterval over which the solution is being approximated.

The approximate solution is available at each of the mesh points $t_0, t_1, \ldots, t_i$ before the approximation at t_{i+1} is obtained, and because the error $|w_j - y(t_j)|$ tends to increase with j, so it seems reasonable to develop methods that use these more accurate previous data when approximating the solution at t_{i+1}.

Methods using the approximation at more than one previous mesh point to determine the approximation at the next point are called *multistep* methods. The precise definition of these methods follows, together with the definition of the two types of multistep methods.

Definition 5.14 An ***m*-step multistep method** for solving the initial-value problem

$$y' = f(t, y), \quad a \le t \le b, \quad y(a) = \alpha, \tag{5.23}$$

has a difference equation for finding the approximation w_{i+1} at the mesh point t_{i+1} represented by the following equation, where m is an integer greater than 1:

$$\begin{aligned} w_{i+1} = {} & a_{m-1}w_i + a_{m-2}w_{i-1} + \cdots + a_0 w_{i+1-m} \\ & + h[b_m f(t_{i+1}, w_{i+1}) + b_{m-1} f(t_i, w_i) \\ & + \cdots + b_0 f(t_{i+1-m}, w_{i+1-m})], \end{aligned} \tag{5.24}$$

for $i = m-1, m, \ldots, N-1$, where $h = (b-a)/N$, the $a_0, a_1, \ldots, a_{m-1}$ and $b_0, b_1, \ldots, b_m$ are constants, and the starting values

$$w_0 = \alpha, \quad w_1 = \alpha_1, \quad w_2 = \alpha_2, \quad \ldots, \quad w_{m-1} = \alpha_{m-1}$$

are specified. ■

When $b_m = 0$ the method is called **explicit**, or **open**, because Eq. (5.24) then gives w_{i+1} explicitly in terms of previously determined values. When $b_m \neq 0$ the method is called

implicit, or **closed**, because w_{i+1} occurs on both sides of Eq. (5.243), so w_{i+1} is specified only implicitly.

For example, the equations

$$w_0 = \alpha, \quad w_1 = \alpha_1, \quad w_2 = \alpha_2, \quad w_3 = \alpha_3,$$

$$w_{i+1} = w_i + \frac{h}{24}[55f(t_i, w_i) - 59f(t_{i-1}, w_{i-1}) + 37f(t_{i-2}, w_{i-2}) - 9f(t_{i-3}, w_{i-3})], \tag{5.25}$$

The Adams-Bashforth techniques are due to John Couch Adams (1819–1892), who did significant work in mathematics and astronomy. He developed these numerical techniques to approximate the solution of a fluid-flow problem posed by Bashforth.

for each $i = 3, 4, \ldots, N-1$, define an *explicit* four-step method known as the **fourth-order Adams-Bashforth technique**. The equations

$$w_0 = \alpha, \quad w_1 = \alpha_1, \quad w_2 = \alpha_2,$$

$$w_{i+1} = w_i + \frac{h}{24}[9f(t_{i+1}, w_{i+1}) + 19f(t_i, w_i) - 5f(t_{i-1}, w_{i-1}) + f(t_{i-2}, w_{i-2})], \tag{5.26}$$

for each $i = 2, 3, \ldots, N-1$, define an *implicit* three-step method known as the **fourth-order Adams-Moulton technique**.

Forest Ray Moulton (1872–1952) was in charge of ballistics at the Aberdeen Proving Grounds in Maryland during World War I. He was a prolific author, writing numerous books in mathematics and astronomy, and developed improved multistep methods for solving ballistic equations.

The starting values in either (5.25) or (5.26) must be specified, generally by assuming $w_0 = \alpha$ and generating the remaining values by either a Runge-Kutta or Taylor method. We will see that the implicit methods are generally more accurate then the explicit methods, but to apply an implicit method such as (5.25) directly, we must solve the implicit equation for w_{i+1}. This is not always possible,and even when it can be done the solution for w_{i+1} may not be unique.

Example 1 In Example 3 of Section 5.4 (see Table 5.8 on page 289) we used the Runge-Kutta method of order four with $h = 0.2$ to approximate the solutions to the initial value problem

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

The first four approximations were found to be $y(0) = w_0 = 0.5$, $y(0.2) \approx w_1 = 0.8292933$, $y(0.4) \approx w_2 = 1.2140762$, and $y(0.6) \approx w_3 = 1.6489220$. Use these as starting values for the fourth-order Adams-Bashforth method to compute new approximations for $y(0.8)$ and $y(1.0)$, and compare these new approximations to those produced by the Runge-Kutta method of order four.

Solution For the fourth-order Adams-Bashforth we have

$$\begin{aligned} y(0.8) \approx w_4 &= w_3 + \frac{0.2}{24}(55f(0.6, w_3) - 59f(0.4, w_2) + 37f(0.2, w_1) - 9f(0, w_0)) \\ &= 1.6489220 + \frac{0.2}{24}(55f(0.6, 1.6489220) - 59f(0.4, 1.2140762) \\ &\quad + 37f(0.2, 0.8292933) - 9f(0, 0.5)) \\ &= 1.6489220 + 0.0083333(55(2.2889220) - 59(2.0540762) \\ &\quad + 37(1.7892933) - 9(1.5)) \\ &= 2.1272892, \end{aligned}$$

and

$$\begin{aligned} y(1.0) \approx w_5 &= w_4 + \frac{0.2}{24}(55f(0.8, w_4) - 59f(0.6, w_3) + 37f(0.4, w_2) - 9f(0.2, w_1)) \\ &= 2.1272892 + \frac{0.2}{24}(55f(0.8, 2.1272892) - 59f(0.6, 1.6489220) \\ &\quad + 37f(0.4, 1.2140762) - 9f(0.2, 0.8292933)) \\ &= 2.1272892 + 0.0083333(55(2.4872892) - 59(2.2889220) \\ &\quad + 37(2.0540762) - 9(1.7892933)) \\ &= 2.6410533, \end{aligned}$$

The error for these approximations at $t = 0.8$ and $t = 1.0$ are, respectively

$$|2.1272295 - 2.1272892| = 5.97 \times 10^{-5} \text{ and } |2.6410533 - 2.6408591| = 1.94 \times 10^{-4}.$$

The corresponding Runge-Kutta approximations had errors

$$|2.1272027 - 2.1272892| = 2.69 \times 10^{-5} \text{ and } |2.6408227 - 2.6408591| = 3.64 \times 10^{-5}.$$

■

Adams was particularly interested in the using his ability for accurate numerical calculations to investigate the orbits of the planets. He predicted the existence of Neptune by analyzing the irregularities in the planet Uranus, and developed various numerical integration techniques to assist in the approximation of the solution of differential equations.

To begin the derivation of a multistep method, note that the solution to the initial-value problem

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

if integrated over the interval $[t_i, t_{i+1}]$, has the property that

$$y(t_{i+1}) - y(t_i) = \int_{t_i}^{t_{i+1}} y'(t)\,dt = \int_{t_i}^{t_{i+1}} f(t, y(t))\,dt.$$

Consequently,

$$y(t_{i+1}) = y(t_i) + \int_{t_i}^{t_{i+1}} f(t, y(t))\,dt. \tag{5.27}$$

However we cannot integrate $f(t, y(t))$ without knowing $y(t)$, the solution to the problem, so we instead integrate an interpolating polynomial $P(t)$ to $f(t, y(t))$, one that is determined by some of the previously obtained data points (t_0, w_0), (t_1, w_1), $\ldots$, (t_i, w_i). When we assume, in addition, that $y(t_i) \approx w_i$, Eq. (5.27) becomes

$$y(t_{i+1}) \approx w_i + \int_{t_i}^{t_{i+1}} P(t)\,dt. \tag{5.28}$$

Although any form of the interpolating polynomial can be used for the derivation, it is most convenient to use the Newton backward-difference formula, because this form more easily incorporates the most recently calculated data.

To derive an Adams-Bashforth explicit m-step technique, we form the backward-difference polynomial $P_{m-1}(t)$ through

$$(t_i, f(t_i, y(t_i))), \quad (t_{i-1}, f(t_{i-1}, y(t_{i-1}))), \ldots, \quad (t_{i+1-m}, f(t_{i+1-m}, y(t_{i+1-m}))).$$

Since $P_{m-1}(t)$ is an interpolatory polynomial of degree $m-1$, some number ξ_i in (t_{i+1-m}, t_i) exists with

$$f(t, y(t)) = P_{m-1}(t) + \frac{f^{(m)}(\xi_i, y(\xi_i))}{m!}(t - t_i)(t - t_{i-1}) \cdots (t - t_{i+1-m}).$$

Introducing the variable substitution $t = t_i + sh$, with $dt = h\,ds$, into $P_{m-1}(t)$ and the error term implies that

$$\begin{aligned}\int_{t_i}^{t_{i+1}} f(t, y(t))\,dt &= \int_{t_i}^{t_{i+1}} \sum_{k=0}^{m-1} (-1)^k \binom{-s}{k} \nabla^k f(t_i, y(t_i))\,dt \\ &\quad + \int_{t_i}^{t_{i+1}} \frac{f^{(m)}(\xi_i, y(\xi_i))}{m!}(t - t_i)(t - t_{i-1}) \cdots (t - t_{i+1-m})\,dt \\ &= \sum_{k=0}^{m-1} \nabla^k f(t_i, y(t_i)) h (-1)^k \int_0^1 \binom{-s}{k} ds \\ &\quad + \frac{h^{m+1}}{m!} \int_0^1 s(s+1) \cdots (s+m-1) f^{(m)}(\xi_i, y(\xi_i))\,ds.\end{aligned}$$

The integrals $(-1)^k \int_0^1 \binom{-s}{k} ds$ for various values of k are easily evaluated and are listed in Table 5.12. For example, when $k = 3$,

$$\begin{aligned}(-1)^3 \int_0^1 \binom{-s}{3} ds &= -\int_0^1 \frac{(-s)(-s-1)(-s-2)}{1 \cdot 2 \cdot 3}\,ds \\ &= \frac{1}{6}\int_0^1 (s^3 + 3s^2 + 2s)\,ds \\ &= \frac{1}{6}\left[\frac{s^4}{4} + s^3 + s^2\right]_0^1 = \frac{1}{6}\left(\frac{9}{4}\right) = \frac{3}{8}.\end{aligned}$$

Table 5.12

k	$\int_0^1 \binom{-s}{k} ds$
0	1
1	$\frac{1}{2}$
2	$\frac{5}{12}$
3	$\frac{3}{8}$
4	$\frac{251}{720}$
5	$\frac{95}{288}$

As a consequence,

$$\begin{aligned}\int_{t_i}^{t_{i+1}} f(t, y(t))\,dt &= h\left[f(t_i, y(t_i)) + \frac{1}{2}\nabla f(t_i, y(t_i)) + \frac{5}{12}\nabla^2 f(t_i, y(t_i)) + \cdots\right] \\ &\quad + \frac{h^{m+1}}{m!}\int_0^1 s(s+1) \cdots (s+m-1) f^{(m)}(\xi_i, y(\xi_i))\,ds. \qquad (5.29)\end{aligned}$$

Because $s(s+1) \cdots (s+m-1)$ does not change sign on $[0, 1]$, the Weighted Mean Value Theorem for Integrals can be used to deduce that for some number μ_i, where $t_{i+1-m} < \mu_i < t_{i+1}$, the error term in Eq. (5.29) becomes

$$\begin{aligned}&\frac{h^{m+1}}{m!}\int_0^1 s(s+1) \cdots (s+m-1) f^{(m)}(\xi_i, y(\xi_i))\,ds \\ &\qquad = \frac{h^{m+1} f^{(m)}(\mu_i, y(\mu_i))}{m!}\int_0^1 s(s+1) \cdots (s+m-1)\,ds.\end{aligned}$$

Hence the error in (5.29) simplifies to

$$h^{m+1} f^{(m)}(\mu_i, y(\mu_i))(-1)^m \int_0^1 \binom{-s}{m} ds. \qquad (5.30)$$

But $y(t_{i+1}) - y(t_i) = \int_{t_i}^{t_{i+1}} f(t, y(t))\,dt$, so Eq. (5.27) can be written as

$$\begin{aligned}y(t_{i+1}) &= y(t_i) + h\left[f(t_i, y(t_i)) + \frac{1}{2}\nabla f(t_i, y(t_i)) + \frac{5}{12}\nabla^2 f(t_i, y(t_i)) + \cdots\right] \\ &\quad + h^{m+1} f^{(m)}(\mu_i, y(\mu_i))(-1)^m \int_0^1 \binom{-s}{m} ds. \qquad (5.31)\end{aligned}$$

Example 2 Use Eq. (5.31) with $m = 3$ to derive the three-step Adams-Bashforth technique.

Solution We have

$$\begin{aligned} y(t_{i+1}) &\approx y(t_i) + h\left[f(t_i, y(t_i)) + \frac{1}{2}\nabla f(t_i, y(t_i)) + \frac{5}{12}\nabla^2 f(t_i, y(t_i))\right] \\ &= y(t_i) + h\left\{f(t_i, y(t_i)) + \frac{1}{2}[f(t_i, y(t_i)) - f(t_{i-1}, y(t_{i-1}))]\right. \\ &\quad \left. + \frac{5}{12}[f(t_i, y(t_i)) - 2f(t_{i-1}, y(t_{i-1})) + f(t_{i-2}, y(t_{i-2}))]\right\} \\ &= y(t_i) + \frac{h}{12}[23f(t_i, y(t_i)) - 16f(t_{i-1}, y(t_{i-1})) + 5f(t_{i-2}, y(t_{i-2}))]. \end{aligned}$$

The three-step Adams-Bashforth method is, consequently,

$$w_0 = \alpha, \quad w_1 = \alpha_1, \quad w_2 = \alpha_2,$$

$$w_{i+1} = w_i + \frac{h}{12}[23f(t_i, w_i) - 16f(t_{i-1}, w_{i-1})] + 5f(t_{i-2}, w_{i-2})],$$

for $i = 2, 3, \ldots, N-1$. ■

Multistep methods can also be derived using Taylor series. An example of the procedure involved is considered in Exercise 12. A derivation using a Lagrange interpolating polynomial is discussed in Exercise 11.

The local truncation error for multistep methods is defined analogously to that of one-step methods. As in the case of one-step methods, the local truncation error provides a measure of how the solution to the differential equation fails to solve the difference equation.

Definition 5.15 If $y(t)$ is the solution to the initial-value problem

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

and

$$\begin{aligned} w_{i+1} &= a_{m-1}w_i + a_{m-2}w_{i-1} + \cdots + a_0 w_{i+1-m} \\ &\quad + h[b_m f(t_{i+1}, w_{i+1}) + b_{m-1}f(t_i, w_i) + \cdots + b_0 f(t_{i+1-m}, w_{i+1-m})] \end{aligned}$$

is the $(i+1)$st step in a multistep method, the **local truncation error** at this step is

$$\begin{aligned} \tau_{i+1}(h) = \frac{y(t_{i+1}) - a_{m-1}y(t_i) - \cdots - a_0 y(t_{i+1-m})}{h} \\ - [b_m f(t_{i+1}, y(t_{i+1})) + \cdots + b_0 f(t_{i+1-m}, y(t_{i+1-m}))], \end{aligned} \tag{5.32}$$

for each $i = m-1, m, \ldots, N-1$. ■

Example 3 Determine the local truncation error for the three-step Adams-Bashforth method derived in Example 2.

Solution Considering the form of the error given in Eq. (5.30) and the appropriate entry in Table 5.12 gives

$$h^4 f^{(3)}(\mu_i, y(\mu_i))(-1)^3 \int_0^1 \binom{-s}{3} ds = \frac{3h^4}{8} f^{(3)}(\mu_i, y(\mu_i)).$$

Using the fact that $f^{(3)}(\mu_i, y(\mu_i)) = y^{(4)}(\mu_i)$ and the difference equation derived in Example 2, we have

$$\tau_{i+1}(h) = \frac{y(t_{i+1}) - y(t_i)}{h} - \frac{1}{12}[23f(t_i, y(t_i)) - 16f(t_{i-1}, y(t_{i-1})) + 5f(t_{i-2}, y(t_{i-2}))]$$
$$= \frac{1}{h}\left[\frac{3h^4}{8} f^{(3)}(\mu_i, y(\mu_i))\right] = \frac{3h^3}{8} y^{(4)}(\mu_i), \quad \text{for some } \mu_i \in (t_{i-2}, t_{i+1}). \blacksquare$$

Adams-Bashforth Explicit Methods

Some of the explicit multistep methods together with their required starting values and local truncation errors are as follows. The derivation of these techniques is similar to the procedure in Examples 2 and 3.

Adams-Bashforth Two-Step Explicit Method

$$w_0 = \alpha, \quad w_1 = \alpha_1,$$
$$w_{i+1} = w_i + \frac{h}{2}[3f(t_i, w_i) - f(t_{i-1}, w_{i-1})], \tag{5.33}$$

where $i = 1, 2, \ldots, N-1$. The local truncation error is $\tau_{i+1}(h) = \frac{5}{12} y'''(\mu_i)h^2$, for some $\mu_i \in (t_{i-1}, t_{i+1})$.

Adams-Bashforth Three-Step Explicit Method

$$w_0 = \alpha, \quad w_1 = \alpha_1, \quad w_2 = \alpha_2,$$
$$w_{i+1} = w_i + \frac{h}{12}[23f(t_i, w_i) - 16f(t_{i-1}, w_{i-1}) + 5f(t_{i-2}, w_{i-2})], \tag{5.34}$$

where $i = 2, 3, \ldots, N-1$. The local truncation error is $\tau_{i+1}(h) = \frac{3}{8} y^{(4)}(\mu_i)h^3$, for some $\mu_i \in (t_{i-2}, t_{i+1})$.

Adams-Bashforth Four-Step Explicit Method

$$w_0 = \alpha, \quad w_1 = \alpha_1, \quad w_2 = \alpha_2, \quad w_3 = \alpha_3, \tag{5.35}$$
$$w_{i+1} = w_i + \frac{h}{24}[55f(t_i, w_i) - 59f(t_{i-1}, w_{i-1}) + 37f(t_{i-2}, w_{i-2}) - 9f(t_{i-3}, w_{i-3})],$$

where $i = 3, 4, \ldots, N-1$. The local truncation error is $\tau_{i+1}(h) = \frac{251}{720} y^{(5)}(\mu_i)h^4$, for some $\mu_i \in (t_{i-3}, t_{i+1})$.

Adams-Bashforth Five-Step Explicit Method

$$w_0 = \alpha, \quad w_1 = \alpha_1, \quad w_2 = \alpha_2, \quad w_3 = \alpha_3, \quad w_4 = \alpha_4,$$
$$w_{i+1} = w_i + \frac{h}{720}[1901f(t_i, w_i) - 2774f(t_{i-1}, w_{i-1}) \tag{5.36}$$
$$+ 2616f(t_{i-2}, w_{i-2}) - 1274f(t_{i-3}, w_{i-3}) + 251f(t_{i-4}, w_{i-4})],$$

where $i = 4, 5, \ldots, N-1$. The local truncation error is $\tau_{i+1}(h) = \frac{95}{288} y^{(6)}(\mu_i)h^5$, for some $\mu_i \in (t_{i-4}, t_{i+1})$.

Adams-Moulton Implicit Methods

Implicit methods are derived by using $(t_{i+1}, f(t_{i+1}, y(t_{i+1})))$ as an additional interpolation node in the approximation of the integral

$$\int_{t_i}^{t_{i+1}} f(t, y(t))\, dt.$$

Some of the more common implicit methods are as follows.

Adams-Moulton Two-Step Implicit Method

$$w_0 = \alpha, \quad w_1 = \alpha_1,$$
$$w_{i+1} = w_i + \frac{h}{12}[5f(t_{i+1}, w_{i+1}) + 8f(t_i, w_i) - f(t_{i-1}, w_{i-1})], \tag{5.37}$$

where $i = 1, 2, \ldots, N-1$. The local truncation error is $\tau_{i+1}(h) = -\frac{1}{24}y^{(4)}(\mu_i)h^3$, for some $\mu_i \in (t_{i-1}, t_{i+1})$.

Adams-Moulton Three-Step Implicit Method

$$w_0 = \alpha, \quad w_1 = \alpha_1, \quad w_2 = \alpha_2, \tag{5.38}$$
$$w_{i+1} = w_i + \frac{h}{24}[9f(t_{i+1}, w_{i+1}) + 19f(t_i, w_i) - 5f(t_{i-1}, w_{i-1}) + f(t_{i-2}, w_{i-2})],$$

where $i = 2, 3, \ldots, N-1$. The local truncation error is $\tau_{i+1}(h) = -\frac{19}{720}y^{(5)}(\mu_i)h^4$, for some $\mu_i \in (t_{i-2}, t_{i+1})$.

Adams-Moulton Four-Step Implicit Method

$$w_0 = \alpha, \quad w_1 = \alpha_1, \quad w_2 = \alpha_2, \quad w_3 = \alpha_3,$$
$$w_{i+1} = w_i + \frac{h}{720}[251f(t_{i+1}, w_{i+1}) + 646f(t_i, w_i) \tag{5.39}$$
$$- 264f(t_{i-1}, w_{i-1}) + 106f(t_{i-2}, w_{i-2}) - 19f(t_{i-3}, w_{i-3})],$$

where $i = 3, 4, \ldots, N-1$. The local truncation error is $\tau_{i+1}(h) = -\frac{3}{160}y^{(6)}(\mu_i)h^5$, for some $\mu_i \in (t_{i-3}, t_{i+1})$.

It is interesting to compare an m-step Adams-Bashforth explicit method with an $(m-1)$-step Adams-Moulton implicit method. Both involve m evaluations of f per step, and both have the terms $y^{(m+1)}(\mu_i)h^m$ in their local truncation errors. In general, the coefficients of the terms involving f in the local truncation error are smaller for the implicit methods than for the explicit methods. This leads to greater stability and smaller round-off errors for the implicit methods.

Example 4 Consider the initial-value problem

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

Use the exact values given from $y(t) = (t+1)^2 - 0.5e^t$ as starting values and $h = 0.2$ to compare the approximations from **(a)** by the explicit Adams-Bashforth four-step method and **(b)** the implicit Adams-Moulton three-step method.

Solution **(a)** The Adams-Bashforth method has the difference equation

$$w_{i+1} = w_i + \frac{h}{24}[55f(t_i, w_i) - 59f(t_{i-1}, w_{i-1}) + 37f(t_{i-2}, w_{i-2}) - 9f(t_{i-3}, w_{i-3})],$$

for $i = 3, 4, \ldots, 9$. When simplified using $f(t, y) = y - t^2 + 1$, $h = 0.2$, and $t_i = 0.2i$, it becomes

$$w_{i+1} = \frac{1}{24}[35w_i - 11.8w_{i-1} + 7.4w_{i-2} - 1.8w_{i-3} - 0.192i^2 - 0.192i + 4.736].$$

(b) The Adams-Moulton method has the difference equation

$$w_{i+1} = w_i + \frac{h}{24}[9f(t_{i+1}, w_{i+1}) + 19f(t_i, w_i) - 5f(t_{i-1}, w_{i-1}) + f(t_{i-2}, w_{i-2})],$$

for $i = 2, 3, \ldots, 9$. This reduces to

$$w_{i+1} = \frac{1}{24}[1.8w_{i+1} + 27.8w_i - w_{i-1} + 0.2w_{i-2} - 0.192i^2 - 0.192i + 4.736].$$

To use this method explicitly, we meed to solve the equation explicitly solve for w_{i+1}. This gives

$$w_{i+1} = \frac{1}{22.2}[27.8w_i - w_{i-1} + 0.2w_{i-2} - 0.192i^2 - 0.192i + 4.736],$$

for $i = 2, 3, \ldots, 9$.

The results in Table 5.13 were obtained using the exact values from $y(t) = (t + 1)^2 - 0.5e^t$ for α, α_1, α_2, and α_3 in the explicit Adams-Bashforth case and for α, α_1, and α_2 in the implicit Adams-Moulton case. Note that the implicit Adams-Moulton method gives consistently better results. ■

Table 5.13

t_i	Exact	Adams-Bashforth w_i	Error	Adams-Moulton w_i	Error
0.0	0.5000000				
0.2	0.8292986				
0.4	1.2140877				
0.6	1.6489406			1.6489341	0.0000065
0.8	2.1272295	2.1273124	0.0000828	2.1272136	0.0000160
1.0	2.6408591	2.6410810	0.0002219	2.6408298	0.0000293
1.2	3.1799415	3.1803480	0.0004065	3.1798937	0.0000478
1.4	3.7324000	3.7330601	0.0006601	3.7323270	0.0000731
1.6	4.2834838	4.2844931	0.0010093	4.2833767	0.0001071
1.8	4.8151763	4.8166575	0.0014812	4.8150236	0.0001527
2.0	5.3054720	5.3075838	0.0021119	5.3052587	0.0002132

Multistep methods are available as options of the *InitialValueProblem* command, in a manner similar to that of the one step methods. The command for the Adam Bashforth Four Step method applied to our usual example has the form

$C := InitialValueProblem(deq, y(0) = 0.5, t = 2, method = adamsbashforth,$
$submethod = step4, numsteps = 10, output = information, digits = 8)$

The output from this method is similar to the results in Table 5.13 except that the exact values were used in Table 5.13 and approximations were used as starting values for the Maple approximations.

To apply the Adams-Mouton Three Step method to this problem, the options would be changed to *method* = *adamsmoulton*, *submethod* = *step3*.

Predictor-Corrector Methods

In Example 4 the implicit Adams-Moulton method gave better results than the explicit Adams-Bashforth method of the same order. Although this is generally the case, the implicit methods have the inherent weakness of first having to convert the method algebraically to an explicit representation for w_{i+1}. This procedure is not always possible, as can be seen by considering the elementary initial-value problem

$$y' = e^y, \quad 0 \le t \le 0.25, \quad y(0) = 1.$$

Because $f(t, y) = e^y$, the three-step Adams-Moulton method has

$$w_{i+1} = w_i + \frac{h}{24}[9e^{w_{i+1}} + 19e^{w_i} - 5e^{w_{i-1}} + e^{w_{i-2}}]$$

as its difference equation, and this equation cannot be algebraically solved for w_{i+1}.

We could use Newton's method or the secant method to approximate w_{i+1}, but this complicates the procedure considerably. In practice, implicit multistep methods are not used as described above. Rather, they are used to improve approximations obtained by explicit methods. The combination of an explicit method to predict and an implicit to improve the prediction is called a **predictor-corrector method**.

Consider the following fourth-order method for solving an initial-value problem. The first step is to calculate the starting values w_0, w_1, w_2, and w_3 for the four-step explicit Adams-Bashforth method. To do this, we use a fourth-order one-step method, the Runge-Kutta method of order four. The next step is to calculate an approximation, w_{4p}, to $y(t_4)$ using the explicit Adams-Bashforth method as predictor:

$$w_{4p} = w_3 + \frac{h}{24}[55f(t_3, w_3) - 59f(t_2, w_2) + 37f(t_1, w_1) - 9f(t_0, w_0)].$$

This approximation is improved by inserting w_{4p} in the right side of the three-step implicit Adams-Moulton method and using that method as a corrector. This gives

$$w_4 = w_3 + \frac{h}{24}[9f(t_4, w_{4p}) + 19f(t_3, w_3) - 5f(t_2, w_2) + f(t_1, w_1)].$$

The only new function evaluation required in this procedure is $f(t_4, w_{4p})$ in the corrector equation; all the other values of f have been calculated for earlier approximations.

The value w_4 is then used as the approximation to $y(t_4)$, and the technique of using the Adams-Bashforth method as a predictor and the Adams-Moulton method as a corrector is repeated to find w_{5p} and w_5, the initial and final approximations to $y(t_5)$. This process is continued until we obtain an approximation w_c to $y(t_N) = y(b)$.

Improved approximations to $y(t_{i+1})$ might be obtained by iterating the Adams-Moulton formula, but these converge to the approximation given by the implicit formula rather than to the solution $y(t_{i+1})$. Hence it is usually more efficient to use a reduction in the step size if improved accuracy is needed.

Algorithm 5.4 is based on the fourth-order Adams-Bashforth method as predictor and one iteration of the Adams-Moulton method as corrector, with the starting values obtained from the fourth-order Runge-Kutta method.

ALGORITHM
5.4

Adams Fourth-Order Predictor-Corrector

To approximate the solution of the initial-value problem

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

at $(N + 1)$ equally spaced numbers in the interval $[a, b]$:

INPUT endpoints a, b; integer N; initial condition α.

OUTPUT approximation w to y at the $(N + 1)$ values of t.

Step 1 Set $h = (b - a)/N$;
$t_0 = a$;
$w_0 = \alpha$;
OUTPUT (t_0, w_0).

Step 2 For $i = 1, 2, 3$, do Steps 3–5.
(*Compute starting values using Runge-Kutta method.*)

Step 3 Set $K_1 = hf(t_{i-1}, w_{i-1})$;
$K_2 = hf(t_{i-1} + h/2, w_{i-1} + K_1/2)$;
$K_3 = hf(t_{i-1} + h/2, w_{i-1} + K_2/2)$;
$K_4 = hf(t_{i-1} + h, w_{i-1} + K_3)$.

Step 4 Set $w_i = w_{i-1} + (K_1 + 2K_2 + 2K_3 + K_4)/6$;
$t_i = a + ih$.

Step 5 OUTPUT (t_i, w_i).

Step 6 For $i = 4, \ldots, N$ do Steps 7–10.

Step 7 Set $t = a + ih$;
$w = w_3 + h[55f(t_3, w_3) - 59f(t_2, w_2) + 37f(t_1, w_1)$
$\quad - 9f(t_0, w_0)]/24$; (*Predict* w_i.)
$w = w_3 + h[9f(t, w) + 19f(t_3, w_3) - 5f(t_2, w_2)$
$\quad + f(t_1, w_1)]/24$. (*Correct* w_i.)

Step 8 OUTPUT (t, w).

Step 9 For $j = 0, 1, 2$
set $t_j = t_{j+1}$; (*Prepare for next iteration.*)
$w_j = w_{j+1}$.

Step 10 Set $t_3 = t$;
$w_3 = w$.

Step 11 STOP. ■

Example 5 Apply the Adams fourth-order predictor-corrector method with $h = 0.2$ and starting values from the Runge-Kutta fourth order method to the initial-value problem

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

Solution This is continuation and modification of the problem considered in Example 1 at the beginning of the section. In that example we found that the starting approximations from Runge-Kutta are

$$y(0) = w_0 = 0.5,\ y(0.2) \approx w_1 = 0.8292933,\ y(0.4) \approx w_2 = 1.2140762,\ \text{and}$$
$$y(0.6) \approx w_3 = 1.6489220.$$

and the fourth-order Adams-Bashforth method gave

$$\begin{aligned} y(0.8) \approx w_{4p} &= w_3 + \frac{0.2}{24}\left(55f(0.6, w_3) - 59f(0.4, w_2) + 37f(0.2, w_1) - 9f(0, w_0)\right) \\ &= 1.6489220 + \frac{0.2}{24}(55f(0.6, 1.6489220) - 59f(0.4, 1.2140762) \\ &\quad + 37f(0.2, 0.8292933) - 9f(0, 0.5)) \\ &= 1.6489220 + 0.0083333(55(2.2889220) - 59(2.0540762) \\ &\quad + 37(1.7892933) - 9(1.5)) \\ &= 2.1272892. \end{aligned}$$

We will now use w_{4p} as the predictor of the approximation to $y(0.8)$ and determine the corrected value w_4, from the implicit Adams-Moulton method. This gives

$$\begin{aligned} y(0.8) \approx w_4 &= w_3 + \frac{0.2}{24}\left(9f(0.8, w_{4p}) + 19f(0.6, w_3) - 5f(0.4, w_2) + f(0.2, w_1)\right) \\ &= 1.6489220 + \frac{0.2}{24}(9f(0.8, 2.1272892) + 19f(0.6, 1.6489220) \\ &\quad - 5f(0.4, 1.2140762) + f(0.2, 0.8292933)) \\ &= 1.6489220 + 0.0083333(9(2.4872892) + 19(2.2889220) - 5(2.0540762) \\ &\quad + (1.7892933)) \\ &= 2.1272056. \end{aligned}$$

Now we use this approximation to determine the predictor, w_{5p}, for $y(1.0)$ as

$$\begin{aligned} y(1.0) \approx w_{5p} &= w_4 + \frac{0.2}{24}\left(55f(0.8, w_4) - 59f(0.6, w_3) + 37f(0.4, w_2) - 9f(0.2, w_1)\right) \\ &= 2.1272056 + \frac{0.2}{24}(55f(0.8, 2.1272056) - 59f(0.6, 1.6489220) \\ &\quad + 37f(0.4, 1.2140762) - 9f(0.2, 0.8292933)) \\ &= 2.1272056 + 0.0083333(55(2.4872056) - 59(2.2889220) + 37(2.0540762) \\ &\quad - 9(1.7892933)) \\ &= 2.6409314, \end{aligned}$$

and correct this with

$$\begin{aligned} y(1.0) \approx w_5 &= w_4 + \frac{0.2}{24}\left(9f(1.0, w_{5p}) + 19f(0.8, w_4) - 5f(0.6, w_3) + f(0.4, w_2)\right) \\ &= 2.1272056 + \frac{0.2}{24}(9f(1.0, 2.6409314) + 19f(0.8, 2.1272892) \\ &\quad - 5f(0.6, 1.6489220) + f(0.4, 1.2140762)) \end{aligned}$$

$$= 2.1272056 + 0.0083333(9(2.6409314) + 19(2.4872056) - 5(2.2889220) + (2.0540762))$$
$$= 2.6408286.$$

In Example 1 we found that using the explicit Adams-Bashforth method alone produced results that were inferior to those of Runge-Kutta. However, these approximations to $y(0.8)$ and $y(1.0)$ are accurate to within

$$|2.1272295 - 2.1272056| = 2.39 \times 10^{-5} \quad \text{and} \quad |2.6408286 - 2.6408591| = 3.05 \times 10^{-5}.$$

respectively, compared to those of Runge-Kutta, which were accurate, respectively, to within

$$|2.1272027 - 2.1272892| = 2.69 \times 10^{-5} \quad \text{and} \quad |2.6408227 - 2.6408591| = 3.64 \times 10^{-5}.$$

The remaining predictor-corrector approximations were generated using Algorithm 5.4 and are shown in Table 5.14. ■

Table 5.14

t_i	$y_i = y(t_i)$	w_i	Error $\lvert y_i - w_i \rvert$
0.0	0.5000000	0.5000000	0
0.2	0.8292986	0.8292933	0.0000053
0.4	1.2140877	1.2140762	0.0000114
0.6	1.6489406	1.6489220	0.0000186
0.8	2.1272295	2.1272056	0.0000239
1.0	2.6408591	2.6408286	0.0000305
1.2	3.1799415	3.1799026	0.0000389
1.4	3.7324000	3.7323505	0.0000495
1.6	4.2834838	4.2834208	0.0000630
1.8	4.8151763	4.8150964	0.0000799
2.0	5.3054720	5.3053707	0.0001013

Adams Fourth Order Predictor-Corrector method is implemented in Maple for the example problem with

$C := InitialValueProblem(deq, y(0) = 0.5, t = 2, method = adamsbashforthmoulton, submethod = step4, numsteps = 10, output = information, digits = 8)$

and generates the same values as in Table 5.14.

Edward Arthur Milne (1896–1950) worked in ballistic research during World War I, and then for the Solar Physics Observatory at Cambridge. In 1929 he was appointed the W. W. Rouse Ball chair at Wadham College in Oxford.

Other multistep methods can be derived using integration of interpolating polynomials over intervals of the form $[t_j, t_{i+1}]$, for $j \le i-1$, to obtain an approximation to $y(t_{i+1})$. When an interpolating polynomial is integrated over $[t_{i-3}, t_{i+1}]$, the result is the explicit **Milne's method**:

$$w_{i+1} = w_{i-3} + \frac{4h}{3}[2f(t_i, w_i) - f(t_{i-1}, w_{i-1}) + 2f(t_{i-2}, w_{i-2})],$$

which has local truncation error $\frac{14}{45}h^4y^{(5)}(\xi_i)$, for some $\xi_i \in (t_{i-3}, t_{i+1})$.

Simpson's name is associated with this technique because it is based on Simpson's rule for integration.

Milne's method is occasionally used as a predictor for the implicit **Simpson's method**,

$$w_{i+1} = w_{i-1} + \frac{h}{3}[f(t_{i+1}, w_{i+1}) + 4f(t_i, w_i) + f(t_{i-1}, w_{i-1})],$$

which has local truncation error $-(h^4/90)y^{(5)}(\xi_i)$, for some $\xi_i \in (t_{i-1}, t_{i+1})$, and is obtained by integrating an interpolating polynomial over $[t_{i-1}, t_{i+1}]$.

The local truncation error involved with a predictor-corrector method of the Milne-Simpson type is generally smaller than that of the Adams-Bashforth-Moulton method. But the technique has limited use because of round-off error problems, which do not occur with the Adams procedure. Elaboration on this difficulty is given in Section 5.10.

EXERCISE SET 5.6

1. Use all the Adams-Bashforth methods to approximate the solutions to the following initial-value problems. In each case use exact starting values, and compare the results to the actual values.

a. $y' = te^{3t} - 2y, \quad 0 \le t \le 1, \quad y(0) = 0,$ with $h = 0.2;$ actual solution $y(t) = \frac{1}{5}te^{3t} - \frac{1}{25}e^{3t} + \frac{1}{25}e^{-2t}.$

b. $y' = 1 + (t - y)^2, \quad 2 \le t \le 3, \quad y(2) = 1,$ with $h = 0.2;$ actual solution $y(t) = t + \frac{1}{1-t}.$

c. $y' = 1 + y/t, \quad 1 \le t \le 2, \quad y(1) = 2,$ with $h = 0.2;$ actual solution $y(t) = t \ln t + 2t.$

d. $y' = \cos 2t + \sin 3t, \quad 0 \le t \le 1, \quad y(0) = 1,$ with $h = 0.2;$ actual solution $y(t) = \frac{1}{2}\sin 2t - \frac{1}{3}\cos 3t + \frac{4}{3}.$

2. Use each of the Adams-Bashforth methods to approximate the solutions to the following initial-value problems. In each case use starting values obtained from the Runge-Kutta method of order four. Compare the results to the actual values.

a. $y' = \dfrac{2 - 2ty}{t^2 + 1}, \quad 0 \le t \le 1, \quad y(0) = 1,$ with $h = 0.1$ actual solution $y(t) = \dfrac{2t + 1}{t^2 + 2}.$

b. $y' = \dfrac{y^2}{1 + t}, \quad 1 \le t \le 2, \quad y(1) = -(\ln 2)^{-1},$ with $h = 0.1$ actual solution $y(t) = \dfrac{-1}{\ln(t + 1)}.$

c. $y' = (y^2 + y)/t, \quad 1 \le t \le 3, \quad y(1) = -2,$ with $h = 0.2$ actual solution $y(t) = \dfrac{2t}{1 - t}.$

d. $y' = -ty + 4t/y, \quad 0 \le t \le 1, \quad y(0) = 1,$ with $h = 0.1$ actual solution $y(t) = \sqrt{4 - 3e^{-t^2}}.$

3. Use each of the Adams-Bashforth methods to approximate the solutions to the following initial-value problems. In each case use starting values obtained from the Runge-Kutta method of order four. Compare the results to the actual values.

a. $y' = y/t - (y/t)^2, \quad 1 \le t \le 2, \quad y(1) = 1,$ with $h = 0.1;$ actual solution $y(t) = \dfrac{t}{1 + \ln t}.$

b. $y' = 1 + y/t + (y/t)^2, \quad 1 \le t \le 3, \quad y(1) = 0,$ with $h = 0.2;$ actual solution $y(t) = t\tan(\ln t).$

c. $y' = -(y + 1)(y + 3), \quad 0 \le t \le 2, \quad y(0) = -2,$ with $h = 0.1;$ actual solution $y(t) = -3 + 2/(1 + e^{-2t}).$

d. $y' = -5y + 5t^2 + 2t, \quad 0 \le t \le 1, \quad y(0) = 1/3,$ with $h = 0.1;$ actual solution $y(t) = t^2 + \frac{1}{3}e^{-5t}.$

4. Use all the Adams-Moulton methods to approximate the solutions to the Exercises 1(a), 1(c), and 1(d). In each case use exact starting values, and explicitly solve for w_{i+1}. Compare the results to the actual values.

5. Use Algorithm 5.4 to approximate the solutions to the initial-value problems in Exercise 1.

6. Use Algorithm 5.4 to approximate the solutions to the initial-value problems in Exercise 2.

7. Use Algorithm 5.4 to approximate the solutions to the initial-value problems in Exercise 3.

8. Change Algorithm 5.4 so that the corrector can be iterated for a given number p iterations. Repeat Exercise 7 with $p = 2, 3,$ and 4 iterations. Which choice of p gives the best answer for each initial-value problem?

9. The initial-value problem

$$y' = e^y, \quad 0 \le t \le 0.20, \quad y(0) = 1$$

has solution

$$y(t) = 1 - \ln(1 - et).$$

Applying the three-step Adams-Moulton method to this problem is equivalent to finding the fixed point w_{i+1} of

$$g(w) = w_i + \frac{h}{24}\left(9e^{w} + 19e^{w_i} - 5e^{w_{i-1}} + e^{w_{i-2}}\right).$$

a. With $h = 0.01$, obtain w_{i+1} by functional iteration for $i = 2, \ldots, 19$ using exact starting values w_0, w_1, and w_2. At each step use w_i to initially approximate w_{i+1}.

b. Will Newton's method speed the convergence over functional iteration?

10. Use the Milne-Simpson Predictor-Corrector method to approximate the solutions to the initial-value problems in Exercise 3.

11. **a.** Derive the Adams-Bashforth Two-Step method by using the Lagrange form of the interpolating polynomial.

b. Derive the Adams-Bashforth Four-Step method by using Newton's backward-difference form of the interpolating polynomial.

12. Derive the Adams-Bashforth Three-Step method by the following method. Set

$$y(t_{i+1}) = y(t_i) + ahf(t_i, y(t_i)) + bhf(t_{i-1}, y(t_{i-1})) + chf(t_{i-2}, y(t_{i-2})).$$

Expand $y(t_{i+1})$, $f(t_{i-2}, y(t_{i-2}))$, and $f(t_{i-1}, y(t_{i-1}))$ in Taylor series about $(t_i, y(t_i))$, and equate the coefficients of h, h^2 and h^3 to obtain a, b, and c.

13. Derive the Adams-Moulton Two-Step method and its local truncation error by using an appropriate form of an interpolating polynomial.

14. Derive Simpson's method by applying Simpson's rule to the integral

$$y(t_{i+1}) - y(t_{i-1}) = \int_{t_{i-1}}^{t_{i+1}} f(t, y(t))\, dt.$$

15. Derive Milne's method by applying the open Newton-Cotes formula (4.29) to the integral

$$y(t_{i+1}) - y(t_{i-3}) = \int_{t_{i-3}}^{t_{i+1}} f(t, y(t))\, dt.$$

16. Verify the entries in Table 5.12 on page 305.

5.7 Variable Step-Size Multistep Methods

The Runge-Kutta-Fehlberg method is used for error control because at each step it provides, at little additional cost, *two* approximations that can be compared and related to the local truncation error. Predictor-corrector techniques always generate two approximations at each step, so they are natural candidates for error-control adaptation.

To demonstrate the error-control procedure, we construct a variable step-size predictor-corrector method using the four-step explicit Adams-Bashforth method as predictor and the three-step implicit Adams-Moulton method as corrector.

The Adams-Bashforth four-step method comes from the relation

$$y(t_{i+1}) = y(t_i) + \frac{h}{24}[55f(t_i, y(t_i)) - 59f(t_{i-1}, y(t_{i-1})) + 37f(t_{i-2}, y(t_{i-2})) - 9f(t_{i-3}, y(t_{i-3}))] + \frac{251}{720}y^{(5)}(\hat{\mu}_i)h^5,$$

for some $\hat{\mu}_i \in (t_{i-3}, t_{i+1})$. The assumption that the approximations $w_0, w_1, \ldots, w_i$ are all exact implies that the Adams-Bashforth local truncation error is

$$\frac{y(t_{i+1}) - w_{i+1,p}}{h} = \frac{251}{720}y^{(5)}(\hat{\mu}_i)h^4. \tag{5.40}$$

A similar analysis of the Adams-Moulton three-step method, which comes from

$$y(t_{i+1}) = y(t_i) + \frac{h}{24}[9f(t_{i+1}, y(t_{i+1})) + 19f(t_i, y(t_i)) - 5f(t_{i-1}, y(t_{i-1})) + f(t_{i-2}, y(t_{i-2}))] - \frac{19}{720}y^{(5)}(\tilde{\mu}_i)h^5,$$

for some $\tilde{\mu}_i \in (t_{i-2}, t_{i+1})$, leads to the local truncation error

$$\frac{y(t_{i+1}) - w_{i+1}}{h} = -\frac{19}{720}y^{(5)}(\tilde{\mu}_i)h^4. \tag{5.41}$$

To proceed further, we must make the assumption that for small values of h, we have

$$y^{(5)}(\hat{\mu}_i) \approx y^{(5)}(\tilde{\mu}_i).$$

The effectiveness of the error-control technique depends directly on this assumption.

If we subtract Eq. (5.40) from Eq. (5.39), we have

$$\frac{w_{i+1} - w_{i+1,p}}{h} = \frac{h^4}{720}[251y^{(5)}(\hat{\mu}_i) + 19y^{(5)}(\tilde{\mu}_i)] \approx \frac{3}{8}h^4y^{(5)}(\tilde{\mu}_i),$$

so

$$y^{(5)}(\tilde{\mu}_i) \approx \frac{8}{3h^5}(w_{i+1} - w_{i+1,p}). \tag{5.42}$$

Using this result to eliminate the term involving $y^{(5)}(\tilde{\mu}_i)h^4$ from Eq. (5.41) gives the approximation to the Adams-Moulton local truncation error

$$|\tau_{i+1}(h)| = \frac{|y(t_{i+1}) - w_{i+1}|}{h} \approx \frac{19h^4}{720} \cdot \frac{8}{3h^5}|w_{i+1} - w_{i+1,p}| = \frac{19|w_{i+1} - w_{i+1,p}|}{270h}.$$

Suppose we now reconsider (Eq. 5.41) with a new step size qh generating new approximations $\hat{w}_{i+1,p}$ and $\hat{w}_{i+1}$. The object is to choose q so that the local truncation error given in Eq. (5.41) is bounded by a prescribed tolerance ε. If we assume that the value $y^{(5)}(\mu)$ in Eq. (5.41) associated with qh is also approximated using Eq. (5.42), then

$$\frac{|y(t_i + qh) - \hat{w}_{i+1}|}{qh} = \frac{19q^4h^4}{720}|y^{(5)}(\mu)| \approx \frac{19q^4h^4}{720}\left[\frac{8}{3h^5}|w_{i+1} - w_{i+1,p}|\right] = \frac{19q^4}{270}\frac{|w_{i+1} - w_{i+1,p}|}{h},$$

and we need to choose q so that

$$\frac{|y(t_i + qh) - \hat{w}_{i+1}|}{qh} \approx \frac{19q^4}{270}\frac{|w_{i+1} - w_{i+1,p}|}{h} < \varepsilon.$$

That is, choose q so that

$$q < \left(\frac{270}{19}\frac{h\varepsilon}{|w_{i+1} - w_{i+1,p}|}\right)^{1/4} \approx 2\left(\frac{h\varepsilon}{|w_{i+1} - w_{i+1,p}|}\right)^{1/4}.$$

A number of approximation assumptions have been made in this development, so in practice q is chosen conservatively, often as

$$q = 1.5\left(\frac{h\varepsilon}{|w_{i+1} - w_{i+1,p}|}\right)^{1/4}.$$

A change in step size for a multistep method is more costly in terms of function evaluations than for a one-step method, because new equally-spaced starting values must be computed. As a consequence, it is common practice to ignore the step-size change whenever the local truncation error is between $\varepsilon/10$ and ε, that is, when

$$\frac{\varepsilon}{10} < |\tau_{i+1}(h)| = \frac{|y(t_{i+1}) - w_{i+1}|}{h} \approx \frac{19|w_{i+1} - w_{i+1,p}|}{270h} < \varepsilon.$$

In addition, q is given an upper bound to ensure that a single unusually accurate approximation does not result in too large a step size. Algorithm 5.5 incorporates this safeguard with an upper bound of 4.

Remember that the multistep methods require equal step sizes for the starting values. So any change in step size necessitates recalculating new starting values at that point. In Steps 3, 16, and 19 of Algorithm 5.5 this is done by calling a Runge-Kutta subalgorithm (Algorithm 5.2), which has been set up in Step 1.

ALGORITHM
5.5

Adams Variable Step-Size Predictor-Corrector

To approximate the solution of the initial-value problem

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

with local truncation error within a given tolerance:

INPUT endpoints a, b; initial condition α; tolerance TOL; maximum step size $hmax$; minimum step size $hmin$.

OUTPUT i, t_i, w_i, h where at the ith step w_i approximates $y(t_i)$ and the step size h was used, or a message that the minimum step size was exceeded.

Step 1 Set up a subalgorithm for the Runge-Kutta fourth-order method to be called $RK4(h, v_0, x_0, v_1, x_1, v_2, x_2, v_3, x_3)$ that accepts as input a step size h and starting values $v_0 \approx y(x_0)$ and returns $\{(x_j, v_j) \mid j = 1, 2, 3\}$ defined by the following:

for $j = 1, 2, 3$
set $K_1 = hf(x_{j-1}, v_{j-1})$;
$K_2 = hf(x_{j-1} + h/2, v_{j-1} + K_1/2)$
$K_3 = hf(x_{j-1} + h/2, v_{j-1} + K_2/2)$
$K_4 = hf(x_{j-1} + h, v_{j-1} + K_3)$
$v_j = v_{j-1} + (K_1 + 2K_2 + 2K_3 + K_4)/6$;
$x_j = x_0 + jh$.

Step 2 Set $t_0 = a$;
$w_0 = \alpha$;
$h = hmax$;
$FLAG = 1$; (*FLAG will be used to exit the loop in Step 4.*)
$LAST = 0$; (*LAST will indicate when the last value is calculated.*)
OUTPUT (t_0, w_0).

Step 3 Call $RK4(h, w_0, t_0, w_1, t_1, w_2, t_2, w_3, t_3)$;
Set $NFLAG = 1$; (*Indicates computation from RK4.*)
$i = 4$;
$t = t_3 + h$.

Step 4 While ($FLAG = 1$) do Steps 5–20.

Step 5 Set $WP = w_{i-1} + \frac{h}{24}[55f(t_{i-1}, w_{i-1}) - 59f(t_{i-2}, w_{i-2})$
$+ 37f(t_{i-3}, w_{i-3}) - 9f(t_{i-4}, w_{i-4})]$; (*Predict* w_i.)

$WC = w_{i-1} + \frac{h}{24}[9f(t, WP) + 19f(t_{i-1}, w_{i-1})$
$- 5f(t_{i-2}, w_{i-2}) + f(t_{i-3}, w_{i-3})]$; (*Correct* w_i.)

$\sigma = 19|WC - WP|/(270h)$.

Step 6 If $\sigma \le TOL$ then do Steps 7–16 (*Result accepted.*)
else do Steps 17–19. (*Result rejected.*)

Step 7 Set $w_i = WC$; (*Result accepted.*)
$t_i = t$.

Step 8 If $NFLAG = 1$ then for $j = i - 3, i - 2, i - 1, i$
OUTPUT (j, t_j, w_j, h);
(*Previous results also accepted.*)
else OUTPUT (i, t_i, w_i, h).
(*Previous results already accepted.*)

Step 9 If $LAST = 1$ then set $FLAG = 0$ (*Next step is* 20.)
else do Steps 10–16.

Step 10 Set $i = i + 1$;
$NFLAG = 0$.

Step 11 If $\sigma \le 0.1\,TOL$ or $t_{i-1} + h > b$ then do Steps 12–16.
(*Increase h if it is more accurate than required or decrease h to include b as a mesh point.*)

Step 12 Set $q = (TOL/(2\sigma))^{1/4}$.

Step 13 If $q > 4$ then set $h = 4h$
else set $h = qh$.

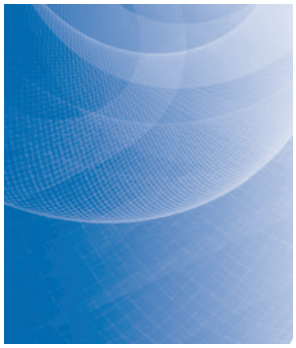
Step 14 If $h > hmax$ then set $h = hmax$.

Step 15 If $t_{i-1} + 4h > b$ then
set $h = (b - t_{i-1})/4$;
$LAST = 1$.

Step 16 Call $RK4(h, w_{i-1}, t_{i-1}, w_i, t_i, w_{i+1}, t_{i+1}, w_{i+2}, t_{i+2})$;
Set $NFLAG = 1$;
$i = i + 3$. (*True branch completed. Next step is 20.*)

Step 17 Set $q = (TOL/(2\sigma))^{1/4}$. (*False branch from Step 6: Result rejected.*)

Step 18 If $q < 0.1$ then set $h = 0.1h$
else set $h = qh$.



Step 19 If $h < hmin$ then set $FLAG = 0$;
OUTPUT ('*hmin* exceeded')
else
if $NFLAG = 1$ then set $i = i - 3$;
(*Previous results also rejected.*)
Call $RK4(h, w_{i-1}, t_{i-1}, w_i, t_i, w_{i+1}, t_{i+1}, w_{i+2}, t_{i+2})$;
set $i = i + 3$;
$NFLAG = 1$.

Step 20 Set $t = t_{i-1} + h$.

Step 21 STOP. ■

Example 1 Use Adams variable step-size predictor-corrector method with maximum step size $hmax = 0.2$, minimum step size $hmin = 0.01$, and tolerance $TOL = 10^{-5}$ to approximate the solution of the initial-value problem

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

Solution We begin with $h = hmax = 0.2$, and obtain w_0, w_1, w_2 and w_3 using Runge-Kutta, then find wp_4 and wc_4 by applying the predictor-corrector method. These calculations were done in Example 5 of Section 5.6 where it was determined that the Runge-Kutta approximations are

$$y(0) = w_0 = 0.5,\ y(0.2) \approx w_1 = 0.8292933,\ y(0.4) \approx w_2 = 1.2140762,\ \text{and}$$
$$y(0.6) \approx w_3 = 1.6489220.$$

The predictor and corrector gave

$$y(0) = w_0 = 0.5,\ y(0.2) \approx w_1 = 0.8292933,\ y(0.4) \approx w_2 = 1.2140762,\ \text{and}$$
$$y(0.6) \approx w_3 = 1.6489220.$$

$$y(0.8) \approx w_{4p} = w_3 + \frac{0.2}{24}\left(55f(0.6, w_3) - 59f(0.4, w_2) + 37f(0.2, w_1) - 9f(0, w_0)\right)$$
$$= 2.1272892,$$

and

$$y(0.8) \approx w_4 = w_3 + \frac{0.2}{24}\left(9f(0.8, w_{4p}) + 19f(0.6, w_3) - 5f(0.42, w_2) + f(0.2, w_1)\right)$$
$$= 2.1272056.$$

We now need to determine if these approximations are sufficiently accurate or if there needs to be a change in the step size. First we find

$$\delta = \frac{19}{270h}|w_4 - w_{4p}| = \frac{19}{270(0.2)}|2.1272056 - 2.1272892| = 2.941 \times 10^{-5}.$$

Because this exceeds the tolerance of 10^{-5} a new step size is needed and the new step size is

$$qh = \left(\frac{10^{-5}}{2\delta}\right)^{1/4} = \left(\frac{10^{-5}}{2(2.941 \times 10^{-5})}\right)^{1/4}(0.2) = 0.642(0.2) \approx 0.128.$$

As a consequence, we need to begin the procedure again computing the Runge-Kutta values with this step size, and then use the predictor-corrector method with this same step size to compute the new values of w_{4p} and w_4. We then need to run the accuracy check on these approximations to see that we have been successful. Table 5.15 shows that this second run is successful and lists the all results obtained using Algorithm 5.5. ■

Table 5.15

t_i	$y(t_i)$	w_i	h_i	σ_i	$\lvert y(t_i) - w_i \rvert$
0	0.5	0.5			
0.1257017	0.7002323	0.7002318	0.1257017	4.051×10^{-6}	0.0000005
0.2514033	0.9230960	0.9230949	0.1257017	4.051×10^{-6}	0.0000011
0.3771050	1.1673894	1.1673877	0.1257017	4.051×10^{-6}	0.0000017
0.5028066	1.4317502	1.4317480	0.1257017	4.051×10^{-6}	0.0000022
0.6285083	1.7146334	1.7146306	0.1257017	4.610×10^{-6}	0.0000028
0.7542100	2.0142869	2.0142834	0.1257017	5.210×10^{-6}	0.0000035
0.8799116	2.3287244	2.3287200	0.1257017	5.913×10^{-6}	0.0000043
1.0056133	2.6556930	2.6556877	0.1257017	6.706×10^{-6}	0.0000054
1.1313149	2.9926385	2.9926319	0.1257017	7.604×10^{-6}	0.0000066
1.2570166	3.3366642	3.3366562	0.1257017	8.622×10^{-6}	0.0000080
1.3827183	3.6844857	3.6844761	0.1257017	9.777×10^{-6}	0.0000097
1.4857283	3.9697541	3.9697433	0.1030100	7.029×10^{-6}	0.0000108
1.5887383	4.2527830	4.2527711	0.1030100	7.029×10^{-6}	0.0000120
1.6917483	4.5310269	4.5310137	0.1030100	7.029×10^{-6}	0.0000133
1.7947583	4.8016639	4.8016488	0.1030100	7.029×10^{-6}	0.0000151
1.8977683	5.0615660	5.0615488	0.1030100	7.760×10^{-6}	0.0000172
1.9233262	5.1239941	5.1239764	0.0255579	3.918×10^{-8}	0.0000177
1.9488841	5.1854932	5.1854751	0.0255579	3.918×10^{-8}	0.0000181
1.9744421	5.2460056	5.2459870	0.0255579	3.918×10^{-8}	0.0000186
2.0000000	5.3054720	5.3054529	0.0255579	3.918×10^{-8}	0.0000191

EXERCISE SET 5.7

1. Use the Adams Variable Step-Size Predictor-Corrector Algorithm with tolerance $TOL = 10^{-4}$, $hmax = 0.25$, and $hmin = 0.025$ to approximate the solutions to the given initial-value problems. Compare the results to the actual values.

a. $y' = te^{3t} - 2y, \quad 0 \le t \le 1, \quad y(0) = 0;$ actual solution $y(t) = \frac{1}{5}te^{3t} - \frac{1}{25}e^{3t} + \frac{1}{25}e^{-2t}$.

b. $y' = 1 + (t - y)^2, \quad 2 \le t \le 3, \quad y(2) = 1;$ actual solution $y(t) = t + 1/(1 - t)$.

c. $y' = 1 + y/t, \quad 1 \le t \le 2, \quad y(1) = 2;$ actual solution $y(t) = t \ln t + 2t$.

d. $y' = \cos 2t + \sin 3t, \quad 0 \le t \le 1, \quad y(0) = 1;$ actual solution $y(t) = \frac{1}{2}\sin 2t - \frac{1}{3}\cos 3t + \frac{4}{3}$.

2. Use the Adams Variable Step-Size Predictor-Corrector Algorithm with $TOL = 10^{-4}$ to approximate the solutions to the following initial-value problems:

a. $y' = (y/t)^2 + y/t, \quad 1 \le t \le 1.2, \quad y(1) = 1,$ with $hmax = 0.05$ and $hmin = 0.01$.

b. $y' = \sin t + e^{-t}, \quad 0 \le t \le 1, \quad y(0) = 0,$ with $hmax = 0.2$ and $hmin = 0.01$.

c. $y' = (y^2 + y)/t, \quad 1 \le t \le 3, \quad y(1) = -2,$ with $hmax = 0.4$ and $hmin = 0.01$.

d. $y' = -ty + 4t/y, \quad 0 \le t \le 1, \quad y(0) = 1,$ with $hmax = 0.2$ and $hmin = 0.01$.

3. Use the Adams Variable Step-Size Predictor-Corrector Algorithm with tolerance $TOL = 10^{-6}$, $hmax = 0.5$, and $hmin = 0.02$ to approximate the solutions to the given initial-value problems. Compare the results to the actual values.

a. $y' = y/t - (y/t)^2, \quad 1 \le t \le 4, \quad y(1) = 1;$ actual solution $y(t) = t/(1 + \ln t)$.

b. $y' = 1 + y/t + (y/t)^2, \quad 1 \le t \le 3, \quad y(1) = 0;$ actual solution $y(t) = t \tan(\ln t)$.

c. $y' = -(y+1)(y+3), \quad 0 \le t \le 3, \quad y(0) = -2;$ actual solution $y(t) = -3 + 2(1 + e^{-2t})^{-1}$.

d. $y' = (t + 2t^3)y^3 - ty, \quad 0 \le t \le 2, \quad y(0) = \frac{1}{3};$ actual solution $y(t) = (3 + 2t^2 + 6e^{t^2})^{-1/2}$.

4. Construct an Adams Variable Step-Size Predictor-Corrector Algorithm based on the Adams-Bashforth five-step method and the Adams-Moulton four-step method. Repeat Exercise 3 using this new method.

5. An electrical circuit consists of a capacitor of constant capacitance $C = 1.1$ farads in series with a resistor of constant resistance $R_0 = 2.1$ ohms. A voltage $\mathcal{E}(t) = 110 \sin t$ is applied at time $t = 0$. When the resistor heats up, the resistance becomes a function of the current i,

$$R(t) = R_0 + ki, \quad \text{where } k = 0.9,$$

and the differential equation for $i(t)$ becomes

$$\left(1 + \frac{2k}{R_0} i\right) \frac{di}{dt} + \frac{1}{R_0 C} i = \frac{1}{R_0 C} \frac{d\mathcal{E}}{dt}.$$

Find $i(2)$, assuming that $i(0) = 0$.

5.8 Extrapolation Methods

Extrapolation was used in Section 4.5 for the approximation of definite integrals, where we found that by correctly averaging relatively inaccurate trapezoidal approximations exceedingly accurate new approximations were produced. In this section we will apply extrapolation to increase the accuracy of approximations to the solution of initial-value problems. As we have previously seen, the original approximations must have an error expansion of a specific form for the procedure to be successful.

To apply extrapolation to solve initial-value problems, we use a technique based on the Midpoint method:

$$w_{i+1} = w_{i-1} + 2hf(t_i, w_i), \quad \text{for } i \ge 1. \tag{5.43}$$

This technique requires two starting values since both w_0 and w_1 are needed before the first midpoint approximation, w_2, can be determined. One starting value is the initial condition for $w_0 = y(a) = \alpha$. To determine the second starting value, w_1, we apply Euler's method. Subsequent approximations are obtained from (5.43). After a series of approximations of this type are generated ending at a value t, an endpoint correction is performed that involves the final two midpoint approximations. This produces an approximation $w(t, h)$ to $y(t)$ that has the form

$$y(t) = w(t, h) + \sum_{k=1}^{\infty} \delta_k h^{2k}, \tag{5.44}$$

where the δ_k are constants related to the derivatives of the solution $y(t)$. The important point is that the δ_k do not depend on the step size h. The details of this procedure can be found in the paper by Gragg [Gr].

To illustrate the extrapolation technique for solving

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

assume that we have a fixed step size h. We wish to approximate $y(t_1) = y(a + h)$.

For the first extrapolation step we let $h_0 = h/2$ and use Euler's method with $w_0 = \alpha$ to approximate $y(a + h_0) = y(a + h/2)$ as

$$w_1 = w_0 + h_0 f(a, w_0).$$

We then apply the Midpoint method with $t_{i-1} = a$ and $t_i = a + h_0 = a + h/2$ to produce a first approximation to $y(a + h) = y(a + 2h_0)$,

$$w_2 = w_0 + 2h_0 f(a + h_0, w_1).$$

The endpoint correction is applied to obtain the final approximation to $y(a+h)$ for the step size h_0. This results in the $O(h_0^2)$ approximation to $y(t_1)$

$$y_{1,1} = \frac{1}{2}[w_2 + w_1 + h_0 f(a + 2h_0, w_2)].$$

We save the approximation $y_{1,1}$ and discard the intermediate results w_1 and w_2.

To obtain the next approximation, $y_{2,1}$, to $y(t_1)$, we let $h_1 = h/4$ and use Euler's method with $w_0 = \alpha$ to obtain an approximation to $y(a + h_1) = y(a + h/4)$ which we will call w_1:

$$w_1 = w_0 + h_1 f(a, w_0).$$

Next we approximate $y(a + 2h_1) = y(a + h/2)$ with w_2, $y(a + 3h_1) = y(a + 3h/4)$ with w_3, and w_4 to $y(a + 4h_1) = y(t_1)$ using the Midpoint method.

$$w_2 = w_0 + 2h_1 f(a + h_1, w_1),$$
$$w_3 = w_1 + 2h_1 f(a + 2h_1, w_2),$$
$$w_4 = w_2 + 2h_1 f(a + 3h_1, w_3).$$

The endpoint correction is now applied to w_3 and w_4 to produce the improved $O(h_1^2)$ approximation to $y(t_1)$,

$$y_{2,1} = \frac{1}{2}[w_4 + w_3 + h_1 f(a + 4h_1, w_4)].$$

Because of the form of the error given in (5.44), the two approximations to $y(a + h)$ have the property that

$$y(a+h) = y_{1,1} + \delta_1 \left(\frac{h}{2}\right)^2 + \delta_2 \left(\frac{h}{2}\right)^4 + \cdots = y_{1,1} + \delta_1 \frac{h^2}{4} + \delta_2 \frac{h^4}{16} + \cdots,$$

and

$$y(a+h) = y_{2,1} + \delta_1 \left(\frac{h}{4}\right)^2 + \delta_2 \left(\frac{h}{4}\right)^4 + \cdots = y_{2,1} + \delta_1 \frac{h^2}{16} + \delta_2 \frac{h^4}{256} + \cdots.$$

We can eliminate the $O(h^2)$ portion of this truncation error by averaging the two formulas appropriately. Specifically, if we subtract the first formula from 4 times the second and divide the result by 3, we have

$$y(a+h) = y_{2,1} + \frac{1}{3}(y_{2,1} - y_{1,1}) - \delta_2 \frac{h^4}{64} + \cdots.$$

So the approximation to $y(t_1)$ given by

$$y_{2,2} = y_{2,1} + \frac{1}{3}(y_{2,1} - y_{1,1})$$

has error of order $O(h^4)$.

We next let $h_2 = h/6$ and apply Euler's method once followed by the Midpoint method five times. Then we use the endpoint correction to determine the h^2 approximation, $y_{3,1}$, to $y(a+h) = y(t_1)$. This approximation can be averaged with $y_{2,1}$ to produce a second $O(h^4)$ approximation that we denote $y_{3,2}$. Then $y_{3,2}$ and $y_{2,2}$ are averaged to eliminate the $O(h^4)$ error terms and produce an approximation with error of order $O(h^6)$. Higher-order formulas are generated by continuing the process.

The only significant difference between the extrapolation performed here and that used for Romberg integration in Section 4.5 results from the way the subdivisions are chosen. In Romberg integration there is a convenient formula for representing the Composite Trapezoidal rule approximations that uses consecutive divisions of the step size by the integers $1, 2, 4, 8, 16, 32, 64, \ldots$ This procedure permits the averaging process to proceed in an easily followed manner.

We do not have a means for easily producing refined approximations for initial-value problems, so the divisions for the extrapolation technique are chosen to minimize the number of required function evaluations. The averaging procedure arising from this choice of subdivision, shown in Table 5.16, is not as elementary, but, other than that, the process is the same as that used for Romberg integration.

Table 5.16

$y_{1,1} = w(t, h_0)$		
$y_{2,1} = w(t, h_1)$	$y_{2,2} = y_{2,1} + \dfrac{h_1^2}{h_0^2 - h_1^2}(y_{2,1} - y_{1,1})$	
$y_{3,1} = w(t, h_2)$	$y_{3,2} = y_{3,1} + \dfrac{h_2^2}{h_1^2 - h_2^2}(y_{3,1} - y_{2,1})$	$y_{3,3} = y_{3,2} + \dfrac{h_2^2}{h_0^2 - h_2^2}(y_{3,2} - y_{2,2})$

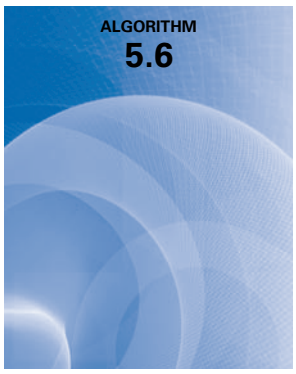
Algorithm 5.6 uses nodes of the form 2^n and $2^n \cdot 3$. Other choices can be used.

Algorithm 5.6 uses the extrapolation technique with the sequence of integers

$$q_0 = 2,\ q_1 = 4,\ q_2 = 6,\ q_3 = 8,\ q_4 = 12,\ q_5 = 16,\ q_6 = 24, \quad \text{and} \quad q_7 = 32.$$

A basic step size h is selected, and the method progresses by using $h_i = h/q_i$, for each $i = 0, \ldots, 7$, to approximate $y(t+h)$. The error is controlled by requiring that the approximations $y_{1,1}, y_{2,2}, \ldots$ be computed until $|y_{i,i} - y_{i-1,i-1}|$ is less than a given tolerance. If the tolerance is not achieved by $i = 8$, then h is reduced, and the process is reapplied.

Minimum and maximum values of h, *hmin*, and *hmax*, respectively, are specified to ensure control of the method. If $y_{i,i}$ is found to be acceptable, then w_1 is set to $y_{i,i}$ and computations begin again to determine w_2, which will approximate $y(t_2) = y(a+2h)$. The process is repeated until the approximation w_N to $y(b)$ is determined.



Extrapolation

To approximate the solution of the initial-value problem

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

with local truncation error within a given tolerance:

INPUT endpoints a, b; initial condition α; tolerance *TOL*; maximum step size *hmax*; minimum step size *hmin*.

OUTPUT T, W, h where W approximates $y(t)$ and step size h was used, or a message that minimum step size was exceeded.

Step 1 Initialize the array $NK = (2, 4, 6, 8, 12, 16, 24, 32)$.

Step 2 Set $TO = a$;
$WO = \alpha$;
$h = hmax$;
$FLAG = 1$. (*FLAG is used to exit the loop in Step* 4.)

Step 3 For $i = 1, 2, \ldots, 7$
for $j = 1, \ldots, i$
set $Q_{i,j} = (NK_{i+1}/NK_j)^2$. (*Note:* $Q_{i,j} = h_j^2/h_{i+1}^2$.)

Step 4 While ($FLAG = 1$) do Steps 5–20.

Step 5 Set $k = 1$;
$NFLAG = 0$. (*When desired accuracy is achieved, NFLAG is set to* 1.)

Step 6 While ($k \le 8$ and $NFLAG = 0$) do Steps 7–14.

Step 7 Set $HK = h/NK_k$;
$T = TO$;
$W2 = WO$;
$W3 = W2 + HK \cdot f(T, W2)$; (*Euler's first step.*)
$T = TO + HK$.

Step 8 For $j = 1, \ldots, NK_k - 1$
set $W1 = W2$;
$W2 = W3$;
$W3 = W1 + 2HK \cdot f(T, W2)$; (*Midpoint method.*)
$T = TO + (j + 1) \cdot HK$.

Step 9 Set $y_k = [W3 + W2 + HK \cdot f(T, W3)]/2$.
(*Endpoint correction to compute* $y_{k,1}$.)

Step 10 If $k \ge 2$ then do Steps 11–13.
(*Note:* $y_{k-1} \equiv y_{k-1,1}, y_{k-2} \equiv y_{k-2,2}, \ldots, y_1 \equiv y_{k-1,k-1}$ *since only the previous row of the table is saved.*)

Step 11 Set $j = k$;
$v = y_1$. (*Save* $y_{k-1,k-1}$.)

Step 12 While ($j \ge 2$) do

$$\text{set } y_{j-1} = y_j + \frac{y_j - y_{j-1}}{Q_{k-1,j-1} - 1};$$

(*Extrapolation to compute* $y_{j-1} \equiv y_{k,k-j+2}$.)

$$\left(\textit{Note:}\quad y_{j-1} = \frac{h_{j-1}^2 y_j - h_k^2 y_{j-1}}{h_{j-1}^2 - h_k^2}.\right)$$

$j = j - 1$.

Step 13 If $|y_1 - v| \le TOL$ then set $NFLAG = 1$.
(y_1 *is accepted as the new* w.)

Step 14 Set $k = k + 1$.

Step 15 Set $k = k - 1$.

Step 16 If $NFLAG = 0$ then do Steps 17 and 18 (*Result rejected.*)
else do Steps 19 and 20. (*Result accepted.*)

Step 17 Set $h = h/2$. (*New value for w rejected, decrease h.*)

Step 18 If $h < hmin$ then
OUTPUT ('*hmin* exceeded');
Set $FLAG = 0$.
(*True branch completed, next step is back to Step* 4.)

Step 19 Set $WO = y_1$; (*New value for w accepted.*)
$TO = TO + h$;
OUTPUT (TO, WO, h).

Step 20 If $TO \geq b$ then set $FLAG = 0$
(*Procedure completed successfully.*)
else if $TO + h > b$ then set $h = b - TO$
(*Terminate at* $t = b$.)
else if ($k \leq 3$ and $h < 0.5(hmax)$ then set $h = 2h$.
(*Increase step size if possible.*)

Step 21 STOP. ■

Example 1 Use the extrapolation method with maximum step size $hmax = 0.2$, minimum step size $hmin = 0.01$, and tolerance $TOL = 10^{-9}$ to approximate the solution of the initial-value problem

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

Solution For the first step of the extrapolation method we let $w_0 = 0.5$, $t_0 = 0$ and $h = 0.2$. Then we compute

$$h_0 = h/2 = 0.1;$$
$$w_1 = w_0 + h_0 f(t_0, w_0) = 0.5 + 0.1(1.5) = 0.65;$$
$$w_2 = w_0 + 2h_0 f(t_0 + h_0, w_1) = 0.5 + 0.2(1.64) = 0.828;$$

and the first approximation to $y(0.2)$ is

$$y_{11} = \frac{1}{2}(w_2 + w_1 + h_0 f(t_0 + 2h_0, w_2)) = \frac{1}{2}(0.828 + 0.65 + 0.1 f(0.2, 0.828)) = 0.8284.$$

For the second approximation to $y(0.2)$ we compute

$$h_1 = h/4 = 0.05;$$
$$w_1 = w_0 + h_1 f(t_0, w_0) = 0.5 + 0.05(1.5) = 0.575;$$
$$w_2 = w_0 + 2h_1 f(t_0 + h_1, w_1) = 0.5 + 0.1(1.5725) = 0.65725;$$
$$w_3 = w_1 + 2h_1 f(t_0 + 2h_1, w_2) = 0.575 + 0.1(1.64725) = 0.739725;$$
$$w_4 = w_2 + 2h_1 f(t_0 + 3h_1, w_3) = 0.65725 + 0.1(1.717225) = 0.8289725.$$

Then the endpoint correction approximation is

$$y_{21} = \frac{1}{2}(w_4 + w_3 + h_1 f(t_0 + 4h_1, w_4))$$
$$= \frac{1}{2}(0.8289725 + 0.739725 + 0.05 f(0.2, 0.8289725)) = 0.8290730625.$$

This gives the first extrapolation approximation

$$y_{22} = y_{21} + \left(\frac{(1/4)^2}{(1/2)^2 - (1/4)^2}\right)(y_{21} - y_{11}) = 0.8292974167.$$

The third approximation is found by computing

$$h_2 = h/6 = 0.0\overline{3};$$
$$w_1 = w_0 + h_2 f(t_0, w_0) = 0.55;$$
$$w_2 = w_0 + 2h_2 f(t_0 + h_2, w_1) = 0.6032592593;$$
$$w_3 = w_1 + 2h_2 f(t_0 + 2h_2, w_2) = 0.6565876543;$$
$$w_4 = w_2 + 2h_2 f(t_0 + 3h_2, w_3) = 0.7130317696;$$
$$w_5 = w_3 + 2h_2 f(t_0 + 4h_2, w_4) = 0.7696045871;$$
$$w_6 = w_4 + 2h_2 f(t_0 + 5h_2, w_4) = 0.8291535569;$$

then the end-point correction approximation

$$y_{31} = \frac{1}{2}(w_6 + w_5 + h_2 f(t_0 + 6h_2, w_6) = 0.8291982979.$$

We can now find two extrapolated approximations,

$$y_{32} = y_{31} + \left(\frac{(1/6)^2}{(1/4)^2 - (1/6)^2}\right)(y_{31} - y_{21}) = 0.8292984862,$$

and

$$y_{33} = y_{32} + \left(\frac{(1/6)^2}{(1/2)^2 - (1/6)^2}\right)(y_{32} - y_{22}) = 0.8292986199.$$

Because

$$|y_{33} - y_{22}| = 1.2 \times 10^{-6}$$

does not satisfy the tolerance, we need to compute at least one more row of the extrapolation table. We use $h_3 = h/8 = 0.025$ and calculate w_1 by Euler's method, $w_2, \cdots, w_8$ by the moidpoint method and apply the endpoint correction. This will give us the new approximation y_{41} which permits us to compute the new extrapolation row

$$y_{41} = 0.8292421745 \quad y_{42} = 0.8292985873 \quad y_{43} = 0.8292986210 \quad y_{44} = 0.8292986211$$

Comparing $|y_{44} - y_{33}| = 1.2 \times 10^{-9}$ we find that the accuracy tolerance has not been reached. To obtain the entries in the next row, we use $h_4 = h/12 = 0.0\overline{6}$. First calculate w_1 by Euler's method, then w_2 through w_{12} by the Midpoint method. Finally use the endpoint correction to obtain y_{51}. The remaining entries in the fifth row are obtained using extrapolation, and are shown in Table 5.17. Because $y_{55} = 0.8292986213$ is within 10^{-9} of y_{44} it is accepted as the approximation to $y(0.2)$. The procedure begins anew to approximate $y(0.4)$. The complete set of approximations accurate to the places listed is given in Table 5.18. ■

Table 5.17

$y_{1,1} = 0.8284000000$				
$y_{2,1} = 0.8290730625$	$y_{2,2} = 0.8292974167$			
$y_{3,1} = 0.8291982979$	$y_{3,2} = 0.8292984862$	$y_{3,3} = 0.8292986199$		
$y_{4,1} = 0.8292421745$	$y_{4,2} = 0.8292985873$	$y_{4,3} = 0.8292986210$	$y_{4,4} = 0.8292986211$	
$y_{5,1} = 0.8292735291$	$y_{5,2} = 0.8292986128$	$y_{5,3} = 0.8292986213$	$y_{5,4} = 0.8292986213$	$y_{5,5} = 0.8292986213$

Table 5.18

t_i	$y_i = y(t_i)$	w_i	h_i	k
0.200	0.8292986210	0.8292986213	0.200	5
0.400	1.2140876512	1.2140876510	0.200	4
0.600	1.6489405998	1.6489406000	0.200	4
0.700	1.8831236462	1.8831236460	0.100	5
0.800	2.1272295358	2.1272295360	0.100	4
0.900	2.3801984444	2.3801984450	0.100	7
0.925	2.4446908698	2.4446908710	0.025	8
0.950	2.5096451704	2.5096451700	0.025	3
1.000	2.6408590858	2.6408590860	0.050	3
1.100	2.9079169880	2.9079169880	0.100	7
1.200	3.1799415386	3.1799415380	0.100	6
1.300	3.4553516662	3.4553516610	0.100	8
1.400	3.7324000166	3.7324000100	0.100	5
1.450	3.8709427424	3.8709427340	0.050	7
1.475	3.9401071136	3.9401071050	0.025	3
1.525	4.0780532154	4.0780532060	0.050	4
1.575	4.2152541820	4.2152541820	0.050	3
1.675	4.4862274254	4.4862274160	0.100	4
1.775	4.7504844318	4.7504844210	0.100	4
1.825	4.8792274904	4.8792274790	0.050	3
1.875	5.0052154398	5.0052154290	0.050	3
1.925	5.1280506670	5.1280506570	0.050	4
1.975	5.2473151731	5.2473151660	0.050	8
2.000	5.3054719506	5.3054719440	0.025	3

The proof that the method presented in Algorithm 5.6 converges involves results from summability theory; it can be found in the original paper of Gragg [Gr]. A number of other extrapolation procedures are available, some of which use the variable step-size techniques. For additional procedures based on the extrapolation process, see the Bulirsch and Stoer papers [BS1], [BS2], [BS3] or the text by Stetter [Stet]. The methods used by Bulirsch and Stoer involve interpolation with rational functions instead of the polynomial interpolation used in the Gragg procedure.

EXERCISE SET 5.8

1. Use the Extrapolation Algorithm with tolerance $TOL = 10^{-4}, hmax = 0.25$, and $hmin = 0.05$ to approximate the solutions to the following initial-value problems. Compare the results to the actual values.

a. $y' = te^{3t} - 2y, \quad 0 \le t \le 1, \quad y(0) = 0;$ actual solution $y(t) = \frac{1}{5}te^{3t} - \frac{1}{25}e^{3t} + \frac{1}{25}e^{-2t}.$

b. $y' = 1 + (t - y)^2, \quad 2 \le t \le 3, \quad y(2) = 1;$ actual solution $y(t) = t + 1/(1 - t).$

c. $y' = 1 + y/t, \quad 1 \le t \le 2, \quad y(1) = 2;$ actual solution $y(t) = t \ln t + 2t.$

d. $y' = \cos 2t + \sin 3t, \quad 0 \le t \le 1, \quad y(0) = 1;$ actual solution $y(t) = \frac{1}{2}\sin 2t - \frac{1}{3}\cos 3t + \frac{4}{3}.$

2. Use the Extrapolation Algorithm with $TOL = 10^{-4}$ to approximate the solutions to the following initial-value problems:

a. $y' = (y/t)^2 + y/t, \quad 1 \le t \le 1.2, \quad y(1) = 1,$ with $hmax = 0.05$ and $hmin = 0.02.$

b. $y' = \sin t + e^{-t}, \quad 0 \le t \le 1, \quad y(0) = 0,$ with $hmax = 0.25$ and $hmin = 0.02.$

c. $y' = (y^2 + y)/t, \quad 1 \le t \le 3, \quad y(1) = -2,$ with $hmax = 0.5$ and $hmin = 0.02.$

d. $y' = -ty + 4t/y, \quad 0 \le t \le 1, \quad y(0) = 1,$ with $hmax = 0.25$ and $hmin = 0.02.$

3. Use the Extrapolation Algorithm with tolerance $TOL = 10^{-6}$, $hmax = 0.5$, and $hmin = 0.05$ to approximate the solutions to the following initial-value problems. Compare the results to the actual values.

a. $y' = y/t - (y/t)^2, \quad 1 \le t \le 4, \quad y(1) = 1;$ actual solution $y(t) = t/(1 + \ln t).$

b. $y' = 1 + y/t + (y/t)^2, \quad 1 \le t \le 3, \quad y(1) = 0;$ actual solution $y(t) = t\tan(\ln t).$

c. $y' = -(y + 1)(y + 3), \quad 0 \le t \le 3, \quad y(0) = -2;$ actual solution $y(t) = -3 + 2(1 + e^{-2t})^{-1}.$

d. $y' = (t + 2t^3)y^3 - ty, \quad 0 \le t \le 2, \quad y(0) = \frac{1}{3};$ actual solution $y(t) = (3 + 2t^2 + 6e^{t^2})^{-1/2}.$

4. Let $P(t)$ be the number of individuals in a population at time t, measured in years. If the average birth rate b is constant and the average death rate d is proportional to the size of the population (due to overcrowding), then the growth rate of the population is given by the **logistic equation**

$$\frac{dP(t)}{dt} = bP(t) - k[P(t)]^2,$$

where $d = kP(t)$. Suppose $P(0) = 50,976$, $b = 2.9 \times 10^{-2}$, and $k = 1.4 \times 10^{-7}$. Find the population after 5 years.

5.9 Higher-Order Equations and Systems of Differential Equations

This section contains an introduction to the numerical solution of higher-order initial-value problems. The techniques discussed are limited to those that transform a higher-order equation into a system of first-order differential equations. Before discussing the transformation procedure, some remarks are needed concerning systems that involve first-order differential equations.

An ***m*th-order system** of first-order initial-value problems has the form

$$\begin{aligned}
\frac{du_1}{dt} &= f_1(t, u_1, u_2, \ldots, u_m),\\
\frac{du_2}{dt} &= f_2(t, u_1, u_2, \ldots, u_m),\\
&\vdots\\
\frac{du_m}{dt} &= f_m(t, u_1, u_2, \ldots, u_m),
\end{aligned} \tag{5.45}$$

for $a \le t \le b$, with the initial conditions

$$u_1(a) = \alpha_1, \; u_2(a) = \alpha_2, \; \ldots, \; u_m(a) = \alpha_m. \tag{5.46}$$

The object is to find m functions $u_1(t), u_2(t), \ldots, u_m(t)$ that satisfy each of the differential equations together with all the initial conditions.

To discuss existence and uniqueness of solutions to systems of equations, we need to extend the definition of the Lipschitz condition to functions of several variables.

Definition 5.16 The function $f(t, y_1, \ldots, y_m)$, defined on the set

$$D = \{(t, u_1, \ldots, u_m) \mid a \le t \le b \text{ and } -\infty < u_i < \infty, \text{ for each } i = 1, 2, \ldots, m\}$$

is said to satisfy a **Lipschitz condition** on D in the variables $u_1, u_2, \ldots, u_m$ if a constant $L > 0$ exists with

$$|f(t, u_1, \ldots, u_m) - f(t, z_1, \ldots, z_m)| \le L \sum_{j=1}^{m} |u_j - z_j|, \tag{5.47}$$

for all $(t, u_1, \ldots, u_m)$ and $(t, z_1, \ldots, z_m)$ in D. ■

By using the Mean Value Theorem, it can be shown that if f and its first partial derivatives are continuous on D and if

$$\left| \frac{\partial f(t, u_1, \ldots, u_m)}{\partial u_i} \right| \le L,$$

for each $i = 1, 2, \ldots, m$ and all $(t, u_1, \ldots, u_m)$ in D, then f satisfies a Lipschitz condition on D with Lipschitz constant L (see [BiR], p. 141). A basic existence and uniqueness theorem follows. Its proof can be found in [BiR], pp. 152–154.

Theorem 5.17 Suppose that

$$D = \{(t, u_1, u_2, \ldots, u_m) \mid a \le t \le b \text{ and } -\infty < u_i < \infty, \text{ for each } i = 1, 2, \ldots, m\},$$

and let $f_i(t, u_1, \ldots, u_m)$, for each $i = 1, 2, \ldots, m$, be continuous and satisfy a Lipschitz condition on D. The system of first-order differential equations (5.45), subject to the initial conditions (5.46), has a unique solution $u_1(t), \ldots, u_m(t)$, for $a \le t \le b$. ■

Methods to solve systems of first-order differential equations are generalizations of the methods for a single first-order equation presented earlier in this chapter. For example, the classical Runge-Kutta method of order four given by

$$\begin{aligned}
w_0 &= \alpha, \\
k_1 &= h f(t_i, w_i), \\
k_2 &= h f\left(t_i + \frac{h}{2}, w_i + \frac{1}{2}k_1\right), \\
k_3 &= h f\left(t_i + \frac{h}{2}, w_i + \frac{1}{2}k_2\right), \\
k_4 &= h f(t_{i+1}, w_i + k_3), \\
w_{i+1} &= w_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4), \quad \text{for each } i = 0, 1, \ldots, N-1,
\end{aligned}$$

used to solve the first-order initial-value problem

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

is generalized as follows.

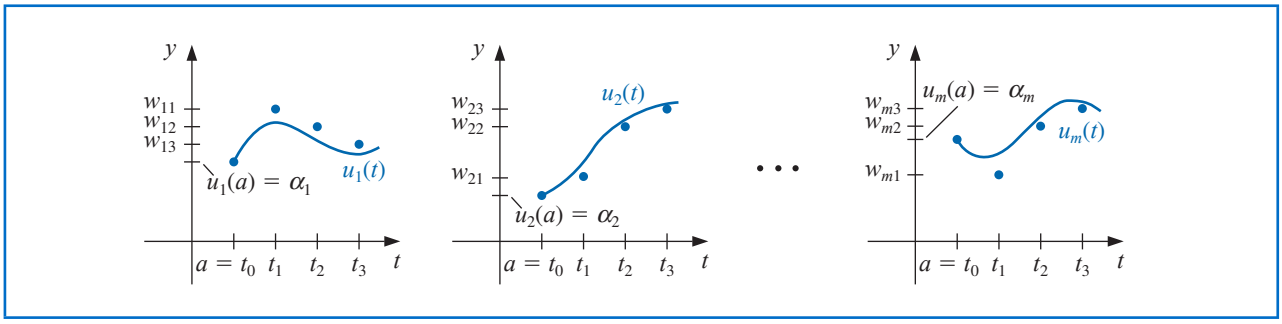
Let an integer $N > 0$ be chosen and set $h = (b-a)/N$. Partition the interval $[a, b]$ into N subintervals with the mesh points

$$t_j = a + jh, \quad \text{for each } j = 0, 1, \ldots, N.$$

Use the notation w_{ij}, for each $j = 0, 1, \ldots, N$ and $i = 1, 2, \ldots, m$, to denote an approximation to $u_i(t_j)$. That is, w_{ij} approximates the ith solution $u_i(t)$ of (5.45) at the jth mesh point t_j. For the initial conditions, set (see Figure 5.6)

$$w_{1,0} = \alpha_1, \; w_{2,0} = \alpha_2, \; \ldots, \; w_{m,0} = \alpha_m. \tag{5.48}$$

Figure 5.6



Suppose that the values $w_{1,j}, w_{2,j}, \ldots, w_{m,j}$ have been computed. We obtain $w_{1,j+1}$, $w_{2,j+1}, \ldots, w_{m,j+1}$ by first calculating

$$k_{1,i} = hf_i(t_j, w_{1,j}, w_{2,j}, \ldots, w_{m,j}), \quad \text{for each } i = 1, 2, \ldots, m; \tag{5.49}$$

$$k_{2,i} = hf_i\left(t_j + \frac{h}{2}, w_{1,j} + \frac{1}{2}k_{1,1}, w_{2,j} + \frac{1}{2}k_{1,2}, \ldots, w_{m,j} + \frac{1}{2}k_{1,m}\right), \tag{5.50}$$

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

$$k_{3,i} = hf_i\left(t_j + \frac{h}{2}, w_{1,j} + \frac{1}{2}k_{2,1}, w_{2,j} + \frac{1}{2}k_{2,2}, \ldots, w_{m,j} + \frac{1}{2}k_{2,m}\right), \tag{5.51}$$

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

$$k_{4,i} = hf_i(t_j + h, w_{1,j} + k_{3,1}, w_{2,j} + k_{3,2}, \ldots, w_{m,j} + k_{3,m}), \tag{5.52}$$

for each $i = 1, 2, \ldots, m$; and then

$$w_{i,j+1} = w_{i,j} + \frac{1}{6}(k_{1,i} + 2k_{2,i} + 2k_{3,i} + k_{4,i}), \tag{5.53}$$

for each $i = 1, 2, \ldots, m$. Note that all the values $k_{1,1}, k_{1,2}, \ldots, k_{1,m}$ must be computed before any of the terms of the form $k_{2,i}$ can be determined. In general, each $k_{l,1}, k_{l,2}, \ldots, k_{l,m}$ must be computed before any of the expressions $k_{l+1,i}$. Algorithm 5.7 implements the Runge-Kutta fourth-order method for systems of initial-value problems.

ALGORITHM
5.7

Runge-Kutta Method for Systems of Differential Equations

To approximate the solution of the mth-order system of first-order initial-value problems

$$u_j' = f_j(t, u_1, u_2, \ldots, u_m), \quad a \le t \le b, \quad \text{with} \quad u_j(a) = \alpha_j,$$

for $j = 1, 2, \ldots, m$ at $(N + 1)$ equally spaced numbers in the interval $[a, b]$:

INPUT endpoints a, b; number of equations m; integer N; initial conditions $\alpha_1, \ldots, \alpha_m$.

OUTPUT approximations w_j to $u_j(t)$ at the $(N + 1)$ values of t.

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

Step 2 For $j = 1, 2, \ldots, m$ set $w_j = \alpha_j$.

Step 3 OUTPUT $(t, w_1, w_2, \ldots, w_m)$.

Step 4 For $i = 1, 2, \ldots, N$ do steps 5–11.

Step 5 For $j = 1, 2, \ldots, m$ set
$k_{1,j} = hf_j(t, w_1, w_2, \ldots, w_m)$.

Step 6 For $j = 1, 2, \ldots, m$ set

$$k_{2,j} = hf_j\left(t + \tfrac{h}{2}, w_1 + \tfrac{1}{2}k_{1,1}, w_2 + \tfrac{1}{2}k_{1,2}, \ldots, w_m + \tfrac{1}{2}k_{1,m}\right).$$

Step 7 For $j = 1, 2, \ldots, m$ set

$$k_{3,j} = hf_j\left(t + \tfrac{h}{2}, w_1 + \tfrac{1}{2}k_{2,1}, w_2 + \tfrac{1}{2}k_{2,2}, \ldots, w_m + \tfrac{1}{2}k_{2,m}\right).$$

Step 8 For $j = 1, 2, \ldots, m$ set
$k_{4,j} = hf_j(t + h, w_1 + k_{3,1}, w_2 + k_{3,2}, \ldots, w_m + k_{3,m})$.

Step 9 For $j = 1, 2, \ldots, m$ set
$w_j = w_j + (k_{1,j} + 2k_{2,j} + 2k_{3,j} + k_{4,j})/6$.

Step 10 Set $t = a + ih$.

Step 11 OUTPUT $(t, w_1, w_2, \ldots, w_m)$.

Step 12 STOP. ■

Illustration Kirchhoff's Law states that the sum of all instantaneous voltage changes around a closed circuit is zero. This law implies that the current $I(t)$ in a closed circuit containing a resistance of R ohms, a capacitance of C farads, an inductance of L henries, and a voltage source of $E(t)$ volts satisfies the equation

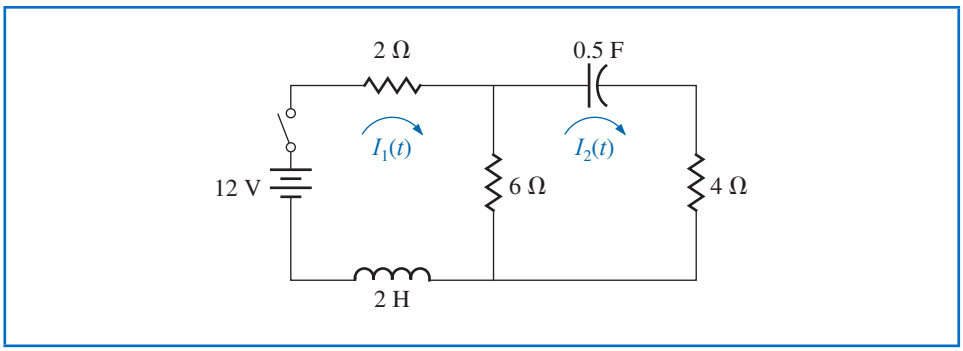
$$LI'(t) + RI(t) + \frac{1}{C}\int I(t)\,dt = E(t).$$

The currents $I_1(t)$ and $I_2(t)$ in the left and right loops, respectively, of the circuit shown in Figure 5.7 are the solutions to the system of equations

$$2I_1(t) + 6[I_1(t) - I_2(t)] + 2I_1'(t) = 12,$$

$$\frac{1}{0.5}\int I_2(t)\,dt + 4I_2(t) + 6[I_2(t) - I_1(t)] = 0.$$

Figure 5.7



If the switch in the circuit is closed at time $t = 0$, we have the initial conditions $I_1(0) = 0$ and $I_2(0) = 0$. Solve for $I_1'(t)$ in the first equation, differentiate the second equation, and substitute for $I_1'(t)$ to get

$$I_1' = f_1(t, I_1, I_2) = -4I_1 + 3I_2 + 6, \quad I_1(0) = 0,$$
$$I_2' = f_2(t, I_1, I_2) = 0.6I_1' - 0.2I_2 = -2.4I_1 + 1.6I_2 + 3.6, \quad I_2(0) = 0.$$

The exact solution to this system is

$$I_1(t) = -3.375e^{-2t} + 1.875e^{-0.4t} + 1.5,$$
$$I_2(t) = -2.25e^{-2t} + 2.25e^{-0.4t}.$$

We will apply the Runge-Kutta method of order four to this system with $h = 0.1$. Since $w_{1,0} = I_1(0) = 0$ and $w_{2,0} = I_2(0) = 0$,

$$k_{1,1} = hf_1(t_0, w_{1,0}, w_{2,0}) = 0.1\ f_1(0, 0, 0) = 0.1\ (-4(0) + 3(0) + 6) = 0.6,$$
$$k_{1,2} = hf_2(t_0, w_{1,0}, w_{2,0}) = 0.1\ f_2(0, 0, 0) = 0.1\ (-2.4(0) + 1.6(0) + 3.6) = 0.36,$$
$$k_{2,1} = hf_1\left(t_0 + \frac{1}{2}h, w_{1,0} + \frac{1}{2}k_{1,1}, w_{2,0} + \frac{1}{2}k_{1,2}\right) = 0.1\ f_1(0.05, 0.3, 0.18)$$
$$= 0.1\ (-4(0.3) + 3(0.18) + 6) = 0.534,$$
$$k_{2,2} = hf_2\left(t_0 + \frac{1}{2}h, w_{1,0} + \frac{1}{2}k_{1,1}, w_{2,0} + \frac{1}{2}k_{1,2}\right) = 0.1\ f_2(0.05, 0.3, 0.18)$$
$$= 0.1\ (-2.4(0.3) + 1.6(0.18) + 3.6) = 0.3168.$$

Generating the remaining entries in a similar manner produces

$$k_{3,1} = (0.1)f_1(0.05, 0.267, 0.1584) = 0.54072,$$
$$k_{3,2} = (0.1)f_2(0.05, 0.267, 0.1584) = 0.321264,$$
$$k_{4,1} = (0.1)f_1(0.1, 0.54072, 0.321264) = 0.4800912,$$
$$k_{4,2} = (0.1)f_2(0.1, 0.54072, 0.321264) = 0.28162944.$$

As a consequence,

$$I_1(0.1) \approx w_{1,1} = w_{1,0} + \frac{1}{6}(k_{1,1} + 2k_{2,1} + 2k_{3,1} + k_{4,1})$$
$$= 0 + \frac{1}{6}(0.6 + 2(0.534) + 2(0.54072) + 0.4800912) = 0.5382552$$

and

$$I_2(0.1) \approx w_{2,1} = w_{2,0} + \frac{1}{6}(k_{1,2} + 2k_{2,2} + 2k_{3,2} + k_{4,2}) = 0.3196263.$$

The remaining entries in Table 5.19 are generated in a similar manner. □

Table 5.19

t_j	$w_{1,j}$	$w_{2,j}$	$\lvert I_1(t_j) - w_{1,j}\rvert$	$\lvert I_2(t_j) - w_{2,j}\rvert$
0.0	0	0	0	0
0.1	0.5382550	0.3196263	0.8285×10^{-5}	0.5803×10^{-5}
0.2	0.9684983	0.5687817	0.1514×10^{-4}	0.9596×10^{-5}
0.3	1.310717	0.7607328	0.1907×10^{-4}	0.1216×10^{-4}
0.4	1.581263	0.9063208	0.2098×10^{-4}	0.1311×10^{-4}
0.5	1.793505	1.014402	0.2193×10^{-4}	0.1240×10^{-4}

Recall that Maple reserves the letter *D* to represent differentiation.

Maple's *NumericalAnalysis* package does not currently approximate the solution to systems of initial value problems, but systems of first-order differential equations can by solved using *dsolve*. The system in the Illustration is defined with

$sys\ 2 := D(u1)(t) = -4u1(t) + 3u2(t) + 6,\ D(u2)(t) = -2.4u1(t) + 1.6u2(t) + 3.6$

and the initial conditions with

$init\ 2 := u1(0) = 0,\ u2(0) = 0$

The system is solved with the command

$sol\ 2 := dsolve(\{sys\ 2, init\ 2\}, \{u1(t), u2(t)\})$

and Maple responds with

$$\left\{u1(t) = -\frac{27}{8}e^{-2t} + \frac{15}{8}e^{-\frac{5}{2}t} + \frac{3}{2},\ u2(t) = -\frac{9}{4}e^{-2t} + \frac{9}{4}e^{-\frac{5}{2}t}\right\}$$

To isolate the individual functions we use

$r1 := rhs(sol\ 2[1]);\ r2 := rhs(sol\ 2[2])$

producing

$$-\frac{27}{8}e^{-2t} + \frac{15}{8}e^{-\frac{5}{2}t} + \frac{3}{2}$$
$$-\frac{9}{4}e^{-2t} + \frac{9}{4}e^{-\frac{5}{2}t}$$

and to determine the value of the functions at $t = 0.5$ we use

$evalf\ (subs(t = 0.5, r1));\ evalf\ (subs(t = 0.5, r2))$

giving, in agreement with Table 5.19,

$$1.793527048$$

$$1.014415451$$

The command *dsolve* will fail if an explicit solution cannot be found. In that case we can use the numeric option in *dsolve*, which applies the Runge-Kutta-Fehlberg technique. This technique can also be used, of course, when the exact solution can be determined with *dsolve*. For example, with the system defined previously,

$g := dsolve(\{sys\ 2, init\ 2\}, \{u1(t), u2(t)\}, numeric)$

returns

$$\textbf{proc}(x\_\ rk\ f45)\ \ldots\ \textbf{end proc}$$

To approximate the solutions at $t = 0.5$, enter

$g(0.5)$

which gives approximations in the form

$$[t = 0.5,\ u2(t) = 1.014415563,\ u1(t) = 1.793527215]$$

Higher-Order Differential Equations

Many important physical problems—for example, electrical circuits and vibrating systems—involve initial-value problems whose equations have orders higher than one. New techniques are not required for solving these problems. By relabeling the variables, we can reduce a higher-order differential equation into a system of first-order differential equations and then apply one of the methods we have already discussed.

A general mth-order initial-value problem

$$y^{(m)}(t) = f(t, y, y', \ldots, y^{(m-1)}), \quad a \le t \le b,$$

with initial conditions $y(a) = \alpha_1, y'(a) = \alpha_2, \ldots, y^{(m-1)}(a) = \alpha_m$ can be converted into a system of equations in the form (5.45) and (5.46).

Let $u_1(t) = y(t), u_2(t) = y'(t), \ldots,$ and $u_m(t) = y^{(m-1)}(t)$. This produces the first-order system

$$\frac{du_1}{dt} = \frac{dy}{dt} = u_2, \quad \frac{du_2}{dt} = \frac{dy'}{dt} = u_3, \quad \cdots, \quad \frac{du_{m-1}}{dt} = \frac{dy^{(m-2)}}{dt} = u_m,$$

and

$$\frac{du_m}{dt} = \frac{dy^{(m-1)}}{dt} = y^{(m)} = f(t, y, y', \ldots, y^{(m-1)}) = f(t, u_1, u_2, \ldots, u_m),$$

with initial conditions

$$u_1(a) = y(a) = \alpha_1, \quad u_2(a) = y'(a) = \alpha_2, \quad \ldots, \quad u_m(a) = y^{(m-1)}(a) = \alpha_m.$$

Example 1 Transform the the second-order initial-value problem

$$y'' - 2y' + 2y = e^{2t} \sin t, \qquad \text{for } 0 \le t \le 1, \quad \text{with } y(0) = -0.4,\ y'(0) = -0.6$$

into a system of first order initial-value problems, and use the Runge-Kutta method with $h = 0.1$ to approximate the solution.

Solution Let $u_1(t) = y(t)$ and $u_2(t) = y'(t)$. This transforms the second-order equation into the system

$$\begin{aligned} u_1'(t) &= u_2(t), \\ u_2'(t) &= e^{2t}\sin t - 2u_1(t) + 2u_2(t), \end{aligned}$$

with initial conditions $u_1(0) = -0.4,\ u_2(0) = -0.6$.

The initial conditions give $w_{1,0} = -0.4$ and $w_{2,0} = -0.6$. The Runge-Kutta Eqs. (5.49) through (5.52) on page 330 with $j = 0$ give

$$\begin{aligned} k_{1,1} &= hf_1(t_0, w_{1,0}, w_{2,0}) = hw_{2,0} = -0.06, \\ k_{1,2} &= hf_2(t_0, w_{1,0}, w_{2,0}) = h\left[e^{2t_0}\sin t_0 - 2w_{1,0} + 2w_{2,0}\right] = -0.04, \\ k_{2,1} &= hf_1\left(t_0 + \frac{h}{2}, w_{1,0} + \frac{1}{2}k_{1,1}, w_{2,0} + \frac{1}{2}k_{1,2}\right) = h\left[w_{2,0} + \frac{1}{2}k_{1,2}\right] = -0.062, \\ k_{2,2} &= hf_2\left(t_0 + \frac{h}{2}, w_{1,0} + \frac{1}{2}k_{1,1}, w_{2,0} + \frac{1}{2}k_{1,2}\right) \\ &= h\left[e^{2(t_0+0.05)}\sin(t_0 + 0.05) - 2\left(w_{1,0} + \frac{1}{2}k_{1,1}\right) + 2\left(w_{2,0} + \frac{1}{2}k_{1,2}\right)\right] \\ &= -0.03247644757, \\ k_{3,1} &= h\left[w_{2,0} + \frac{1}{2}k_{2,2}\right] = -0.06162832238, \\ k_{3,2} &= h\left[e^{2(t_0+0.05)}\sin(t_0 + 0.05) - 2\left(w_{1,0} + \frac{1}{2}k_{2,1}\right) + 2\left(w_{2,0} + \frac{1}{2}k_{2,2}\right)\right] \\ &= -0.03152409237, \\ k_{4,1} &= h\left[w_{2,0} + k_{3,2}\right] = -0.06315240924, \end{aligned}$$

and

$$k_{4,2} = h\left[e^{2(t_0+0.1)}\sin(t_0 + 0.1) - 2(w_{1,0} + k_{3,1}) + 2(w_{2,0} + k_{3,2})\right] = -0.02178637298.$$

So

$$w_{1,1} = w_{1,0} + \frac{1}{6}(k_{1,1} + 2k_{2,1} + 2k_{3,1} + k_{4,1}) = -0.4617333423$$

and

$$w_{2,1} = w_{2,0} + \frac{1}{6}(k_{1,2} + 2k_{2,2} + 2k_{3,2} + k_{4,2}) = -0.6316312421.$$

The value $w_{1,1}$ approximates $u_1(0.1) = y(0.1) = 0.2e^{2(0.1)}(\sin 0.1 - 2\cos 0.1)$, and $w_{2,1}$ approximates $u_2(0.1) = y'(0.1) = 0.2e^{2(0.1)}(4\sin 0.1 - 3\cos 0.1)$.

The set of values $w_{1,j}$ and $w_{2,j}$, for $j = 0, 1, \ldots, 10$, are presented in Table 5.20 and are compared to the actual values of $u_1(t) = 0.2e^{2t}(\sin t - 2\cos t)$ and $u_2(t) = u_1'(t) = 0.2e^{2t}(4\sin t - 3\cos t)$. ■

Table 5.20

t_j	$y(t_j) = u_1(t_j)$	$w_{1,j}$	$y'(t_j) = u_2(t_j)$	$w_{2,j}$	$\lvert y(t_j) - w_{1,j}\rvert$	$\lvert y'(t_j) - w_{2,j}\rvert$
0.0	−0.40000000	−0.40000000	−0.6000000	−0.60000000	0	0
0.1	−0.46173297	−0.46173334	−0.6316304	−0.63163124	3.7×10^{-7}	7.75×10^{-7}
0.2	−0.52555905	−0.52555988	−0.6401478	−0.64014895	8.3×10^{-7}	1.01×10^{-6}
0.3	−0.58860005	−0.58860144	−0.6136630	−0.61366381	1.39×10^{-6}	8.34×10^{-7}
0.4	−0.64661028	−0.64661231	−0.5365821	−0.53658203	2.03×10^{-6}	1.79×10^{-7}
0.5	−0.69356395	−0.69356666	−0.3887395	−0.38873810	2.71×10^{-6}	5.96×10^{-7}
0.6	−0.72114849	−0.72115190	−0.1443834	−0.14438087	3.41×10^{-6}	7.75×10^{-7}
0.7	−0.71814890	−0.71815295	0.2289917	0.22899702	4.05×10^{-6}	2.03×10^{-6}
0.8	−0.66970677	−0.66971133	0.7719815	0.77199180	4.56×10^{-6}	5.30×10^{-6}
0.9	−0.55643814	−0.55644290	1.534764	1.5347815	4.76×10^{-6}	9.54×10^{-6}
1.0	−0.35339436	−0.35339886	2.578741	2.5787663	4.50×10^{-6}	1.34×10^{-5}

In Maple the nth derivative $y^{(n)}(t)$ is specified by $(D@@n)(y)(t)$.

We can also use *dsolve* from Maple on higher-order equations. To define the differential equation in Example 1, use

$def\,2 := (D@@2)(y)(t) - 2D(y)(t) + 2y(t) = e^{2t}\sin(t)$

and to specify the initial conditions use

$init\,2 := y(0) = -0.4, D(y)(0) = -0.6$

The solution is obtained with the command

$sol\,2 := dsolve(\{def\,2, init\,2\}, y(t))$

to obtain

$$y(t) = \frac{1}{5}e^{2t}(\sin(t) - 2\cos(t))$$

We isolate the solution in function form using

$g := rhs(sol\,2)$

To obtain $y(1.0) = g(1.0)$, enter

$evalf\,(subs(t = 1.0, g))$

which gives −0.3533943574.

Runge-Kutta-Fehlberg is also available for higher-order equations via the *dsolve* command with the numeric option. It is employed in the same manner as illustrated for systems of equations.

The other one-step methods can be extended to systems in a similar way. When error control methods like the Runge-Kutta-Fehlberg method are extended, each component of the numerical solution $(w_{1j}, w_{2j}, \ldots, w_{mj})$ must be examined for accuracy. If any of the components fail to be sufficiently accurate, the entire numerical solution $(w_{1j}, w_{2j}, \ldots, w_{mj})$ must be recomputed.

The multistep methods and predictor-corrector techniques can also be extended to systems. Again, if error control is used, each component must be accurate. The extension of the extrapolation technique to systems can also be done, but the notation becomes quite involved. If this topic is of interest, see [HNW1].

Convergence theorems and error estimates for systems are similar to those considered in Section 5.10 for the single equations, except that the bounds are given in terms of vector norms, a topic considered in Chapter 7. (A good reference for these theorems is [Ge1], pp. 45–72.)

EXERCISE SET 5.9

1. Use the Runge-Kutta method for systems to approximate the solutions of the following systems of first-order differential equations, and compare the results to the actual solutions.

a. $u_1' = 3u_1 + 2u_2 - (2t^2 + 1)e^{2t}, \quad u_1(0) = 1;$
$u_2' = 4u_1 + u_2 + (t^2 + 2t - 4)e^{2t}, \quad u_2(0) = 1; \quad 0 \le t \le 1; \quad h = 0.2;$
actual solutions $u_1(t) = \frac{1}{3}e^{5t} - \frac{1}{3}e^{-t} + e^{2t}$ and $u_2(t) = \frac{1}{3}e^{5t} + \frac{2}{3}e^{-t} + t^2e^{2t}$.

b. $u_1' = -4u_1 - 2u_2 + \cos t + 4\sin t, \quad u_1(0) = 0;$
$u_2' = 3u_1 + u_2 - 3\sin t, \quad u_2(0) = -1; \quad 0 \le t \le 2; \quad h = 0.1;$
actual solutions $u_1(t) = 2e^{-t} - 2e^{-2t} + \sin t$ and $u_2(t) = -3e^{-t} + 2e^{-2t}$.

c. $u_1' = u_2, \quad u_1(0) = 1;$
$u_2' = -u_1 - 2e^t + 1, \quad u_2(0) = 0;$
$u_3' = -u_1 - e^t + 1, \quad u_3(0) = 1; \quad 0 \le t \le 2; \quad h = 0.5;$
actual solutions $u_1(t) = \cos t + \sin t - e^t + 1$, $u_2(t) = -\sin t + \cos t - e^t$, and $u_3(t) = -\sin t + \cos t$.

d. $u_1' = u_2 - u_3 + t, \quad u_1(0) = 1;$
$u_2' = 3t^2, \quad u_2(0) = 1;$
$u_3' = u_2 + e^{-t}, \quad u_3(0) = -1; \quad 0 \le t \le 1; \quad h = 0.1;$
actual solutions $u_1(t) = -0.05t^5 + 0.25t^4 + t + 2 - e^{-t}$, $u_2(t) = t^3 + 1$, and $u_3(t) = 0.25t^4 + t - e^{-t}$.

2. Use the Runge-Kutta method for systems to approximate the solutions of the following systems of first-order differential equations, and compare the results to the actual solutions.

a. $u_1' = u_1 - u_2 + 2, \quad u_1(0) = -1;$
$u_2' = -u_1 + u_2 + 4t, \quad u_2(0) = 0; \quad 0 \le t \le 1; \quad h = 0.1;$
actual solutions $u_1(t) = -\frac{1}{2}e^{2t} + t^2 + 2t - \frac{1}{2}$ and $u_2(t) = \frac{1}{2}e^{2t} + t^2 - \frac{1}{2}$.

b. $u_1' = \frac{1}{9}u_1 - \frac{2}{3}u_2 - \frac{1}{9}t^2 + \frac{2}{3}, \quad u_1(0) = -3;$
$u_2' = u_2 + 3t - 4, \quad u_2(0) = 5; \quad 0 \le t \le 2; \quad h = 0.2;$
actual solutions $u_1(t) = -3e^t + t^2$ and $u_2(t) = 4e^t - 3t + 1$.

c. $u_1' = u_1 + 2u_2 - 2u_3 + e^{-t}, \quad u_1(0) = 3;$
$u_2' = u_2 + u_3 - 2e^{-t}, \quad u_2(0) = -1;$
$u_3' = u_1 + 2u_2 + e^{-t}, \quad u_3(0) = 1; \quad 0 \le t \le 1; \quad h = 0.1;$
actual solutions $u_1(t) = -3e^{-t} - 3\sin t + 6\cos t$, $u_2(t) = \frac{3}{2}e^{-t} + \frac{3}{10}\sin t - \frac{21}{10}\cos t - \frac{2}{5}e^{2t}$, and $u_3(t) = -e^{-t} + \frac{12}{5}\cos t + \frac{9}{5}\sin t - \frac{2}{5}e^{2t}$.

d. $u_1' = 3u_1 + 2u_2 - u_3 - 1 - 3t - 2\sin t, \quad u_1(0) = 5;$
$u_2' = u_1 - 2u_2 + 3u_3 + 6 - t + 2\sin t + \cos t, \quad u_2(0) = -9;$
$u_3' = 2u_1 + 4u_3 + 8 - 2t, \quad u_3(0) = -5; \quad 0 \le t \le 2; \quad h = 0.2;$
actual solutions $u_1(t) = 2e^{3t} + 3e^{-2t} + 1$, $u_2(t) = -8e^{-2t} + e^{4t} - 2e^{3t} + \sin t$, and $u_3(t) = 2e^{4t} - 4e^{3t} - e^{-2t} - 2$.

3. Use the Runge-Kutta for Systems Algorithm to approximate the solutions of the following higher-order differential equations, and compare the results to the actual solutions.

a. $y'' - 2y' + y = te^t - t, \quad 0 \le t \le 1, \quad y(0) = y'(0) = 0,$ with $h = 0.1;$
actual solution $y(t) = \frac{1}{6}t^3e^t - te^t + 2e^t - t - 2$.

b. $t^2y'' - 2ty' + 2y = t^3\ln t, \quad 1 \le t \le 2, \quad y(1) = 1, \quad y'(1) = 0,$ with $h = 0.1;$
actual solution $y(t) = \frac{7}{4}t + \frac{1}{2}t^3\ln t - \frac{3}{4}t^3$.

c. $y''' + 2y'' - y' - 2y = e^t, \quad 0 \le t \le 3, \quad y(0) = 1, \quad y'(0) = 2, \quad y''(0) = 0,$ with $h = 0.2;$
actual solution $y(t) = \frac{43}{36}e^t + \frac{1}{4}e^{-t} - \frac{4}{9}e^{-2t} + \frac{1}{6}te^t$.

d. $t^3y''' - t^2y'' + 3ty' - 4y = 5t^3\ln t + 9t^3, \quad 1 \le t \le 2, \quad y(1) = 0, \quad y'(1) = 1, \quad y''(1) = 3,$
with $h = 0.1;$ actual solution $y(t) = -t^2 + t\cos(\ln t) + t\sin(\ln t) + t^3\ln t$.

4. Use the Runge-Kutta for Systems Algorithm to approximate the solutions of the following higher-order differential equations, and compare the results to the actual solutions.

a. $y'' - 3y' + 2y = 6e^{-t}, \quad 0 \le t \le 1, \quad y(0) = y'(0) = 2,$ with $h = 0.1$;
actual solution $y(t) = 2e^{2t} - e^t + e^{-t}$.

b. $t^2y'' + ty' - 4y = -3t, \quad 1 \le t \le 3, \quad y(1) = 4, \quad y'(1) = 3,$ with $h = 0.2$;
actual solution $y(t) = 2t^2 + t + t^{-2}$.

c. $y''' + y'' - 4y' - 4y = 0, \quad 0 \le t \le 2, \quad y(0) = 3, \quad y'(0) = -1, \quad y''(0) = 9,$ with $h = 0.2$;
actual solution $y(t) = e^{-t} + e^{2t} + e^{-2t}$.

d. $t^3y''' + t^2y'' - 2ty' + 2y = 8t^3 - 2, \quad 1 \le t \le 2, \quad y(1) = 2, \quad y'(1) = 8, \quad y''(1) = 6,$ with $h = 0.1$; actual solution $y(t) = 2t - t^{-1} + t^2 + t^3 - 1$.

5. Change the Adams Fourth-Order Predictor-Corrector Algorithm to obtain approximate solutions to systems of first-order equations.

6. Repeat Exercise 2 using the algorithm developed in Exercise 5.

7. Repeat Exercise 1 using the algorithm developed in Exercise 5.

8. Suppose the swinging pendulum described in the lead example of this chapter is 2 ft long and that $g = 32.17$ ft/s^2. With $h = 0.1$ s, compare the angle θ obtained for the following two initial-value problems at $t = 0$, 1, and 2 s.

a. $$\frac{d^2\theta}{dt^2} + \frac{g}{L}\sin\theta = 0, \quad \theta(0) = \frac{\pi}{6}, \quad \theta'(0) = 0,$$

b. $$\frac{d^2\theta}{dt^2} + \frac{g}{L}\theta = 0, \quad \theta(0) = \frac{\pi}{6}, \quad \theta'(0) = 0,$$

9. The study of mathematical models for predicting the population dynamics of competing species has its origin in independent works published in the early part of the 20th century by A. J. Lotka and V. Volterra (see, for example, [Lo1], [Lo2], and [Vo]).

Consider the problem of predicting the population of two species, one of which is a predator, whose population at time t is $x_2(t)$, feeding on the other, which is the prey, whose population is $x_1(t)$. We will assume that the prey always has an adequate food supply and that its birth rate at any time is proportional to the number of prey alive at that time; that is, birth rate (prey) is $k_1x_1(t)$. The death rate of the prey depends on both the number of prey and predators alive at that time. For simplicity, we assume death rate (prey) = $k_2x_1(t)x_2(t)$. The birth rate of the predator, on the other hand, depends on its food supply, $x_1(t)$, as well as on the number of predators available for reproduction purposes. For this reason, we assume that the birth rate (predator) is $k_3x_1(t)x_2(t)$. The death rate of the predator will be taken as simply proportional to the number of predators alive at the time; that is, death rate (predator) = $k_4x_2(t)$.

Since $x_1'(t)$ and $x_2'(t)$ represent the change in the prey and predator populations, respectively, with respect to time, the problem is expressed by the system of nonlinear differential equations

$$x_1'(t) = k_1x_1(t) - k_2x_1(t)x_2(t) \quad \text{and} \quad x_2'(t) = k_3x_1(t)x_2(t) - k_4x_2(t).$$

Solve this system for $0 \le t \le 4$, assuming that the initial population of the prey is 1000 and of the predators is 500 and that the constants are $k_1 = 3$, $k_2 = 0.002$, $k_3 = 0.0006$, and $k_4 = 0.5$. Sketch a graph of the solutions to this problem, plotting both populations with time, and describe the physical phenomena represented. Is there a stable solution to this population model? If so, for what values x_1 and x_2 is the solution stable?

10. In Exercise 9 we considered the problem of predicting the population in a predator-prey model. Another problem of this type is concerned with two species competing for the same food supply. If the numbers of species alive at time t are denoted by $x_1(t)$ and $x_2(t)$, it is often assumed that, although the birth rate of each of the species is simply proportional to the number of species alive at that time, the death rate of each species depends on the population of both species. We will assume that the population of a particular pair of species is described by the equations

$$\frac{dx_1(t)}{dt} = x_1(t)[4 - 0.0003x_1(t) - 0.0004x_2(t)] \quad \text{and} \quad \frac{dx_2(t)}{dt} = x_2(t)[2 - 0.0002x_1(t) - 0.0001x_2(t)].$$

If it is known that the initial population of each species is 10,000, find the solution to this system for $0 \le t \le 4$. Is there a stable solution to this population model? If so, for what values of x_1 and x_2 is the solution stable?

5.10 Stability

A number of methods have been presented in this chapter for approximating the solution to an initial-value problem. Although numerous other techniques are available, we have chosen the methods described here because they generally satisfied three criteria:

- Their development is clear enough so that you can understand how and why they work.
- One or more of the methods will give satisfactory results for most of the problems that are encountered by students in science and engineering.
- Most of the more advanced and complex techniques are based on one or a combination of the procedures described here.

One-Step Methods

In this section, we discuss why these methods are expected to give satisfactory results when some similar methods do not. Before we begin this discussion, we need to present two definitions concerned with the convergence of one-step difference-equation methods to the solution of the differential equation as the step size decreases.

Definition 5.18 A one-step difference-equation method with local truncation error $\tau_i(h)$ at the ith step is said to be **consistent** with the differential equation it approximates if

$$\lim_{h \to 0} \max_{1 \le i \le N} |\tau_i(h)| = 0.$$ ■

A one-step method is consistent if the difference equation for the method approaches the differential equation as the step size goes to zero.

Note that this definition is a *local* definition since, for each of the values $\tau_i(h)$, we are assuming that the approximation w_{i-1} and the exact solution $y(t_{i-1})$ are the same. A more realistic means of analyzing the effects of making h small is to determine the *global* effect of the method. This is the maximum error of the method over the entire range of the approximation, assuming only that the method gives the exact result at the initial value.

Definition 5.19 A one-step difference-equation method is said to be **convergent** with respect to the differential equation it approximates if

$$\lim_{h \to 0} \max_{1 \le i \le N} |w_i - y(t_i)| = 0,$$

where $y(t_i)$ denotes the exact value of the solution of the differential equation and w_i is the approximation obtained from the difference method at the ith step. ■

A method is convergent if the solution to the difference equation approaches the solution to the differential equation as the step size goes to zero.

Example 1 Show that Euler's method is convergent.

Solution Examining Inequality (5.10) on page 271, in the error-bound formula for Euler's method, we see that under the hypotheses of Theorem 5.9,

$$\max_{1 \le i \le N} |w_i - y(t_i)| \le \frac{Mh}{2L} |e^{L(b-a)} - 1|.$$

However, M, L, a, and b are all constants and

$$\lim_{h\to 0} \max_{1\le i\le N} |w_i - y(t_i)| \le \lim_{h\to 0} \frac{Mh}{2L} \left|e^{L(b-a)} - 1\right| = 0.$$

So Euler's method is convergent with respect to a differential equation satisfying the conditions of this definition. The rate of convergence is $O(h)$. ■

A consistent one-step method has the property that the difference equation for the method approaches the differential equation when the step size goes to zero. So the local truncation error of a consistent method approaches zero as the step size approaches zero.

The other error-bound type of problem that exists when using difference methods to approximate solutions to differential equations is a consequence of not using exact results. In practice, neither the initial conditions nor the arithmetic that is subsequently performed is represented exactly because of the round-off error associated with finite-digit arithmetic. In Section 5.2 we saw that this consideration can lead to difficulties even for the convergent Euler's method.

A method is stable when the results depend continuously on the initial data.

To analyze this situation, at least partially, we will try to determine which methods are **stable**, in the sense that small changes or perturbations in the initial conditions produce correspondingly small changes in the subsequent approximations.

The concept of stability of a one-step difference equation is somewhat analogous to the condition of a differential equation being well-posed, so it is not surprising that the Lipschitz condition appears here, as it did in the corresponding theorem for differential equations, Theorem 5.6 in Section 5.1.

Part (i) of the following theorem concerns the stability of a one-step method. The proof of this result is not difficult and is considered in Exercise 1. Part (ii) of Theorem 5.20 concerns sufficient conditions for a consistent method to be convergent. Part (iii) justifies the remark made in Section 5.5 about controlling the global error of a method by controlling its local truncation error and implies that when the local truncation error has the rate of convergence $O(h^n)$, the global error will have the same rate of convergence. The proofs of parts (ii) and (iii) are more difficult than that of part (i), and can be found within the material presented in [Ge1], pp. 57–58.

Theorem 5.20 Suppose the initial-value problem

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

is approximated by a one-step difference method in the form

$$w_0 = \alpha,$$
$$w_{i+1} = w_i + h\phi(t_i, w_i, h).$$

Suppose also that a number $h_0 > 0$ exists and that $\phi(t, w, h)$ is continuous and satisfies a Lipschitz condition in the variable w with Lipschitz constant L on the set

$$D = \{(t, w, h) \mid a \le t \le b \text{ and } -\infty < w < \infty, 0 \le h \le h_0\}.$$

Then

(i) The method is stable;

(ii) The difference method is convergent if and only if it is consistent, which is equivalent to

$$\phi(t, y, 0) = f(t, y), \quad \text{for all } a \le t \le b;$$

(iii) If a function τ exists and, for each $i = 1, 2, \ldots, N$, the local truncation error $\tau_i(h)$ satisfies $|\tau_i(h)| \le \tau(h)$ whenever $0 \le h \le h_0$, then

$$|y(t_i) - w_i| \le \frac{\tau(h)}{L} e^{L(t_i - a)}.$$

■

Example 2 The Modified Euler method is given by $w_0 = \alpha$,

$$w_{i+1} = w_i + \frac{h}{2}\left[f(t_i, w_i) + f(t_{i+1}, w_i + hf(t_i, w_i))\right], \quad \text{for } i = 0, 1, \ldots, N-1.$$

Verify that this method is stable by showing that it satisfies the hypothesis of Theorem 5.20.

Solution For this method,

$$\phi(t, w, h) = \frac{1}{2} f(t, w) + \frac{1}{2} f(t + h, w + hf(t, w)).$$

If f satisfies a Lipschitz condition on $\{(t, w) \mid a \le t \le b \text{ and } -\infty < w < \infty\}$ in the variable w with constant L, then, since

$$\begin{aligned}\phi(t, w, h) - \phi(t, \overline{w}, h) = &\frac{1}{2} f(t, w) + \frac{1}{2} f(t + h, w + hf(t, w)) \\ &- \frac{1}{2} f(t, \overline{w}) - \frac{1}{2} f(t + h, \overline{w} + hf(t, \overline{w})),\end{aligned}$$

the Lipschitz condition on f leads to

$$\begin{aligned}|\phi(t, w, h) - \phi(t, \overline{w}, h)| &\le \frac{1}{2} L|w - \overline{w}| + \frac{1}{2} L\,|w + hf(t, w) - \overline{w} - hf(t, \overline{w})| \\ &\le L|w - \overline{w}| + \frac{1}{2} L\,|hf(t, w) - hf(t, \overline{w})| \\ &\le L|w - \overline{w}| + \frac{1}{2} hL^2 |w - \overline{w}| \\ &= \left(L + \frac{1}{2} hL^2\right) |w - \overline{w}|.\end{aligned}$$

Therefore, ϕ satisfies a Lipschitz condition in w on the set

$$\{(t, w, h) \mid a \le t \le b, -\infty < w < \infty, \text{ and } 0 \le h \le h_0\},$$

for any $h_0 > 0$ with constant

$$L' = L + \frac{1}{2} h_0 L^2.$$

Finally, if f is continuous on $\{(t, w) \mid a \le t \le b, -\infty < w < \infty\}$, then ϕ is continuous on

$$\{(t, w, h) \mid a \le t \le b, -\infty < w < \infty, \text{ and } 0 \le h \le h_0\};$$

so Theorem 5.20 implies that the Modified Euler method is stable. Letting $h = 0$, we have

$$\phi(t, w, 0) = \frac{1}{2} f(t, w) + \frac{1}{2} f(t + 0, w + 0 \cdot f(t, w)) = f(t, w),$$

so the consistency condition expressed in Theorem 5.20, part (ii), holds. Thus, the method is convergent. Moreover, we have seen that for this method the local truncation error is $O(h^2)$, so the convergence of the Modified Euler method is also $O(h^2)$. ■

Multistep Methods

For multistep methods, the problems involved with consistency, convergence, and stability are compounded because of the number of approximations involved at each step. In the one-step methods, the approximation w_{i+1} depends directly only on the previous approximation w_i, whereas the multistep methods use at least two of the previous approximations, and the usual methods that are employed involve more.

The general multistep method for approximating the solution to the initial-value problem

$$y' = f(t, y), \quad a \le t \le b, \quad y(a) = \alpha, \tag{5.54}$$

has the form

$$\begin{aligned} &w_0 = \alpha, \quad w_1 = \alpha_1, \quad \ldots, \ w_{m-1} = \alpha_{m-1}, \\ &w_{i+1} = a_{m-1}w_i + a_{m-2}w_{i-1} + \cdots + a_0 w_{i+1-m} + hF(t_i, h, w_{i+1}, w_i, \ldots, w_{i+1-m}), \end{aligned} \tag{5.55}$$

for each $i = m-1, m, \ldots, N-1$, where $a_0, a_1, \ldots, a_{m+1}$ are constants and, as usual, $h = (b-a)/N$ and $t_i = a + ih$.

The local truncation error for a multistep method expressed in this form is

$$\tau_{i+1}(h) = \frac{y(t_{i+1}) - a_{m-1}y(t_i) - \cdots - a_0 y(t_{i+1-m})}{h} - F(t_i, h, y(t_{i+1}), y(t_i), \ldots, y(t_{i+1-m})),$$

for each $i = m-1, m, \ldots, N-1$. As in the one-step methods, the local truncation error measures how the solution y to the differential equation fails to satisfy the difference equation.

For the four-step Adams-Bashforth method, we have seen that

$$\tau_{i+1}(h) = \frac{251}{720}y^{(5)}(\mu_i)h^4, \quad \text{for some } \mu_i \in (t_{i-3}, t_{i+1}),$$

whereas the three-step Adams-Moulton method has

$$\tau_{i+1}(h) = -\frac{19}{720}y^{(5)}(\mu_i)h^4, \quad \text{for some } \mu_i \in (t_{i-2}, t_{i+1}),$$

provided, of course, that $y \in C^5[a, b]$.

Throughout the analysis, two assumptions will be made concerning the function F:

- If $f \equiv 0$ (that is, if the differential equation is homogeneous), then $F \equiv 0$ also.
- F satisfies a Lipschitz condition with respect to $\{w_j\}$, in the sense that a constant L exists and, for every pair of sequences $\{v_j\}_{j=0}^N$ and $\{\tilde{v}_j\}_{j=0}^N$ and for $i = m-1, m, \ldots, N-1$, we have

$$|F(t_i, h, v_{i+1}, \ldots, v_{i+1-m}) - F(t_i, h, \tilde{v}_{i+1}, \ldots, \tilde{v}_{i+1-m})| \le L\sum_{j=0}^{m} |v_{i+1-j} - \tilde{v}_{i+1-j}|.$$

The explicit Adams-Bashforth and implicit Adams-Moulton methods satisfy both of these conditions, provided f satisfies a Lipschitz condition. (See Exercise 2.)

The concept of convergence for multistep methods is the same as that for one-step methods.

- A multistep method is **convergent** if the solution to the difference equation approaches the solution to the differential equation as the step size approaches zero. This means that $\lim_{h\to 0} \max_{0\le i\le N} \mid w_i - y(t_i)| = 0$.

For consistency, however, a slightly different situation occurs. Again, we want a multistep method to be consistent provided that the difference equation approaches the differential equation as the step size approaches zero; that is, the local truncation error approaches zero at each step as the step size approaches zero. The additional condition occurs because of the number of starting values required for multistep methods. Since usually only the first starting value, $w_0 = \alpha$, is exact, we need to require that the errors in all the starting values $\{\alpha_i\}$ approach zero as the step size approaches zero. So

$$\lim_{h\to 0} |\tau_i(h)| = 0, \quad \text{for all } i = m, m+1, \ldots, N \quad \text{and} \tag{5.56}$$

$$\lim_{h\to 0} |\alpha_i - y(t_i)| = 0, \quad \text{for all } i = 1, 2, \ldots, m-1, \tag{5.57}$$

must be true for a multistep method in the form (5.55) to be **consistent**. Note that (5.57) implies that a multistep method will not be consistent unless the one-step method generating the starting values is also consistent.

The following theorem for multistep methods is similar to Theorem 5.20, part (iii), and gives a relationship between the local truncation error and global error of a multistep method. It provides the theoretical justification for attempting to control global error by controlling local truncation error. The proof of a slightly more general form of this theorem can be found in [IK], pp. 387–388.

Theorem 5.21 Suppose the initial-value problem

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

is approximated by an explicit Adams predictor-corrector method with an m-step Adams-Bashforth predictor equation

$$w_{i+1} = w_i + h[b_{m-1} f(t_i, w_i) + \cdots + b_0 f(t_{i+1-m}, w_{i+1-m})],$$

with local truncation error $\tau_{i+1}(h)$, and an $(m-1)$-step implicit Adams-Moulton corrector equation

$$w_{i+1} = w_i + h\left[\tilde{b}_{m-1} f(t_i, w_{i+1}) + \tilde{b}_{m-2} f(t_i, w_i) + \cdots + \tilde{b}_0 f(t_{i+2-m}, w_{i+2-m})\right],$$

with local truncation error $\tilde{\tau}_{i+1}(h)$. In addition, suppose that $f(t, y)$ and $f_y(t, y)$ are continuous on $D = \{(t, y) \mid a \le t \le b \text{ and } -\infty < y < \infty\}$ and that f_y is bounded. Then the local truncation error $\sigma_{i+1}(h)$ of the predictor-corrector method is

$$\sigma_{i+1}(h) = \tilde{\tau}_{i+1}(h) + \tau_{i+1}(h)\tilde{b}_{m-1}\frac{\partial f}{\partial y}(t_{i+1}, \theta_{i+1}),$$

where θ_{i+1} is a number between zero and $h\tau_{i+1}(h)$.

Moreover, there exist constants k_1 and k_2 such that

$$|w_i - y(t_i)| \le \left[\max_{0\le j\le m-1} \left|w_j - y(t_j)\right| + k_1\sigma(h)\right] e^{k_2(t_i - a)},$$

where $\sigma(h) = \max_{m\le j\le N} |\sigma_j(h)|$. ■

Before discussing connections between consistency, convergence, and stability for multistep methods, we need to consider in more detail the difference equation for a multistep method. In doing so, we will discover the reason for choosing the Adams methods as our standard multistep methods.

Associated with the difference equation (5.55) given at the beginning of this discussion,

$$w_0 = \alpha,\ w_1 = \alpha_1,\ \ldots,\ w_{m-1} = \alpha_{m-1},$$
$$w_{i+1} = a_{m-1}w_i + a_{m-2}w_{i-1} + \cdots + a_0 w_{i+1-m} + hF(t_i, h, w_{i+1}, w_i, \ldots, w_{i+1-m}),$$

is a polynomial, called the **characteristic polynomial** of the method, given by

$$P(\lambda) = \lambda^m - a_{m-1}\lambda^{m-1} - a_{m-2}\lambda^{m-2} - \cdots - a_1\lambda - a_0. \tag{5.58}$$

The stability of a multistep method with respect to round-off error is dictated the by magnitudes of the zeros of the characteristic polynomial. To see this, consider applying the standard multistep method (5.55) to the trivial initial-value problem

$$y' \equiv 0, \quad y(a) = \alpha, \quad \text{where } \alpha \neq 0. \tag{5.59}$$

This problem has exact solution $y(t) \equiv \alpha$. By examining Eqs. (5.27) and (5.28) in Section 5.6 (see page 304), we can see that any multistep method will, in theory, produce the exact solution $w_n = \alpha$ for all n. The only deviation from the exact solution is due to the round-off error of the method.

The right side of the differential equation in (5.59) has $f(t, y) \equiv 0$, so by assumption (1), we have $F(t_i, h, w_{i+1}, w_{i+2}, \ldots, w_{i+1-m}) = 0$ in the difference equation (5.55). As a consequence, the standard form of the difference equation becomes

$$w_{i+1} = a_{m-1}w_i + a_{m-2}w_{i-1} + \cdots + a_0 w_{i+1-m}. \tag{5.60}$$

Suppose λ is one of the zeros of the characteristic polynomial associated with (5.55). Then $w_n = \lambda^n$ for each n is a solution to (5.59) since

$$\lambda^{i+1} - a_{m-1}\lambda^i - a_{m-2}\lambda^{i-1} - \cdots - a_0\lambda^{i+1-m} = \lambda^{i+1-m}[\lambda^m - a_{m-1}\lambda^{m-1} - \cdots - a_0] = 0.$$

In fact, if $\lambda_1, \lambda_2, \ldots, \lambda_m$ are distinct zeros of the characteristic polynomial for (5.55), it can be shown that *every* solution to (5.60) can be expressed in the form

$$w_n = \sum_{i=1}^{m} c_i \lambda_i^n, \tag{5.61}$$

for some unique collection of constants $c_1, c_2, \ldots, c_m$.

Since the exact solution to (5.59) is $y(t) = \alpha$, the choice $w_n = \alpha$, for all n, is a solution to (5.60). Using this fact in (5.60) gives

$$0 = \alpha - \alpha a_{m-1} - \alpha a_{m-2} - \cdots - \alpha a_0 = \alpha[1 - a_{m-1} - a_{m-2} - \cdots - a_0].$$

This implies that $\lambda = 1$ is one of the zeros of the characteristic polynomial (5.58). We will assume that in the representation (5.61) this solution is described by $\lambda_1 = 1$ and $c_1 = \alpha$, so all solutions to (5.59) are expressed as

$$w_n = \alpha + \sum_{i=2}^{m} c_i \lambda_i^n. \tag{5.62}$$

If all the calculations were exact, all the constants $c_2, c_3, \ldots, c_m$ would be zero. In practice, the constants $c_2, c_3, \ldots, c_m$ are not zero due to round-off error. In fact, the round-off error

grows exponentially unless $|\lambda_i| \leq 1$ for each of the roots $\lambda_2, \lambda_3, \ldots, \lambda_m$. The smaller the magnitude of these roots, the more stable the method with respect to the growth of round-off error.

In deriving (5.62), we made the simplifying assumption that the zeros of the characteristic polynomial are distinct. The situation is similar when multiple zeros occur. For example, if $\lambda_k = \lambda_{k+1} = \cdots = \lambda_{k+p}$ for some k and p, it simply requires replacing the sum

$$c_k\lambda_k^n + c_{k+1}\lambda_{k+1}^n + \cdots + c_{k+p}\lambda_{k+p}^n$$

in (5.62) with

$$c_k\lambda_k^n + c_{k+1}n\lambda_k^{n-1} + c_{k+2}n(n-1)\lambda_k^{n-2} + \cdots + c_{k+p}[n(n-1)\cdots(n-p+1)]\lambda_k^{n-p}. \tag{5.63}$$

(See [He2], pp. 119–145.) Although the form of the solution is modified, the round-off error if $|\lambda_k| > 1$ still grows exponentially.

Although we have considered only the special case of approximating initial-value problems of the form (5.59), the stability characteristics for this equation determine the stability for the situation when $f(t, y)$ is not identically zero. This is because the solution to the homogeneous equation (5.59) is embedded in the solution to any equation. The following definitions are motivated by this discussion.

Definition 5.22 Let $\lambda_1, \lambda_2, \ldots, \lambda_m$ denote the (not necessarily distinct) roots of the characteristic equation

$$P(\lambda) = \lambda^m - a_{m-1}\lambda^{m-1} - \cdots - a_1\lambda - a_0 = 0$$

associated with the multistep difference method

$$w_0 = \alpha, \quad w_1 = \alpha_1, \quad \ldots, \quad w_{m-1} = \alpha_{m-1}$$

$$w_{i+1} = a_{m-1}w_i + a_{m-2}w_{i-1} + \cdots + a_0w_{i+1-m} + hF(t_i, h, w_{i+1}, w_i, \ldots, w_{i+1-m}).$$

If $|\lambda_i| \leq 1$, for each $i = 1, 2, \ldots, m$, and all roots with absolute value 1 are simple roots, then the difference method is said to satisfy the **root condition**. ■

Definition 5.23

(i) Methods that satisfy the root condition and have $\lambda = 1$ as the only root of the characteristic equation with magnitude one are called **strongly stable**.

(ii) Methods that satisfy the root condition and have more than one distinct root with magnitude one are called **weakly stable**.

(iii) Methods that do not satisfy the root condition are called **unstable**. ■

Consistency and convergence of a multistep method are closely related to the round-off stability of the method. The next theorem details these connections. For the proof of this result and the theory on which it is based, see [IK], pp. 410–417.

Theorem 5.24 A multistep method of the form

$$w_0 = \alpha, \quad w_1 = \alpha_1, \quad \ldots, \quad w_{m-1} = \alpha_{m-1},$$

$$w_{i+1} = a_{m-1}w_i + a_{m-2}w_{i-1} + \cdots + a_0w_{i+1-m} + hF(t_i, h, w_{i+1}, w_i, \ldots, w_{i+1-m})$$

is stable if and only if it satisfies the root condition. Moreover, if the difference method is consistent with the differential equation, then the method is stable if and only if it is convergent. ■

Example 3 The fourth-order Adams-Bashforth method can be expressed as

$$w_{i+1} = w_i + hF(t_i, h, w_{i+1}, w_i, \ldots, w_{i-3}),$$

where

$$F(t_i, h, w_{i+1}, , \ldots, w_{i-3}) = \frac{h}{24}[55f(t_i, w_i) - 59f(t_{i-1}, w_{i-1}) + 37f(t_{i-2}, w_{i-2}) - 9f(t_{i-3}, w_{i-3})];$$

Show that this method is strongly stable.

Solution In this case we have $m = 4$, $a_0 = 0$, $a_1 = 0$, $a_2 = 0$, and $a_3 = 1$, so the characteristic equation for this Adams-Bashforth method is

$$0 = P(\lambda) = \lambda^4 - \lambda^3 = \lambda^3(\lambda - 1).$$

This polynomial has roots $\lambda_1 = 1$, $\lambda_2 = 0$, $\lambda_3 = 0$, and $\lambda_4 = 0$. Hence it satisfies the root condition and is strongly stable.

The Adams-Moulton method has a similar characteristic polynomial, $P(\lambda) = \lambda^3 - \lambda^2$, with zeros $\lambda_1 = 1$, $\lambda_2 = 0$, and $\lambda_3 = 0$, and is also strongly stable. ■

Example 4 Show that the fourth-order Milne's method, the explicit multistep method given by

$$w_{i+1} = w_{i-3} + \frac{4h}{3}\left[2f(t_i, w_i) - f(t_{i-1}, w_{i-1}) + 2f(t_{i-2}, w_{i-2})\right]$$

satisfies the root condition, but it is only weakly stable.

Solution The characteristic equation for this method, $0 = P(\lambda) = \lambda^4 - 1$, has four roots with magnitude one: $\lambda_1 = 1$, $\lambda_2 = -1$, $\lambda_3 = i$, and $\lambda_4 = -i$. Because all the roots have magnitude 1, the method satisfies the root condition. However, there are multiple roots with magnitude 1, so the method is only weakly stable. ■

Example 5 Apply the strongly stable fourth-order Adams-Bashforth method and the weakly stable Milne's method with $h = 0.1$ to the initial-value problem

$$y' = -6y + 6, \quad 0 \le t \le 1, \quad y(0) = 2,$$

which has the exact solution $y(t) = 1 + e^{-6t}$.

Solution The results in Table 5.21 show the effects of a weakly stable method versus a strongly stable method for this problem. ■

Table 5.21

t_i	Exact $y(t_i)$	Adams-Bashforth Method w_i	Error $\lvert y_i - w_i\rvert$	Milne's Method w_i	Error $\lvert y_i - w_i\rvert$
0.10000000		1.5488116		1.5488116	
0.20000000		1.3011942		1.3011942	
0.30000000		1.1652989		1.1652989	
0.40000000	1.0907180	1.0996236	8.906×10^{-3}	1.0983785	7.661×10^{-3}
0.50000000	1.0497871	1.0513350	1.548×10^{-3}	1.0417344	8.053×10^{-3}
0.60000000	1.0273237	1.0425614	1.524×10^{-2}	1.0486438	2.132×10^{-2}
0.70000000	1.0149956	1.0047990	1.020×10^{-2}	0.9634506	5.154×10^{-2}
0.80000000	1.0082297	1.0359090	2.768×10^{-2}	1.1289977	1.208×10^{-1}
0.90000000	1.0045166	0.9657936	3.872×10^{-2}	0.7282684	2.762×10^{-1}
1.00000000	1.0024788	1.0709304	6.845×10^{-2}	1.6450917	6.426×10^{-1}

The reason for choosing the Adams-Bashforth-Moulton as our standard fourth-order predictor-corrector technique in Section 5.6 over the Milne-Simpson method of the same order is that both the Adams-Bashforth and Adams-Moulton methods are strongly stable. They are more likely to give accurate approximations to a wider class of problems than is the predictor-corrector based on the Milne and Simpson techniques, both of which are weakly stable.

EXERCISE SET 5.10

1. To prove Theorem 5.20, part (i), show that the hypotheses imply that there exists a constant $K > 0$ such that

$$|u_i - v_i| \le K|u_0 - v_0|, \quad \text{for each } 1 \le i \le N,$$

whenever $\{u_i\}_{i=1}^N$ and $\{v_i\}_{i=1}^N$ satisfy the difference equation $w_{i+1} = w_i + h\phi(t_i, w_i, h)$.

2. For the Adams-Bashforth and Adams-Moulton methods of order four,

a. Show that if $f = 0$, then

$$F(t_i, h, w_{i+1}, \ldots, w_{i+1-m}) = 0.$$

b. Show that if f satisfies a Lipschitz condition with constant L, then a constant C exists with

$$|F(t_i, h, w_{i+1}, \ldots, w_{i+1-m}) - F(t_i, h, v_{i+1}, \ldots, v_{i+1-m})| \le C \sum_{j=0}^{m} |w_{i+1-j} - v_{i+1-j}|.$$

3. Use the results of Exercise 32 in Section 5.4 to show that the Runge-Kutta method of order four is consistent.

4. Consider the differential equation

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

a. Show that

$$y'(t_i) = \frac{-3y(t_i) + 4y(t_{i+1}) - y(t_{i+2})}{2h} + \frac{h^2}{3} y'''(\xi_1),$$

for some ξ, where $t_i < \xi_i < t_{i+2}$.

b. Part (a) suggests the difference method

$$w_{i+2} = 4w_{i+1} - 3w_i - 2hf(t_i, w_i), \quad \text{for } i = 0, 1, \ldots, N-2.$$

Use this method to solve

$$y' = 1 - y, \quad 0 \le t \le 1, \quad y(0) = 0,$$

with $h = 0.1$. Use the starting values $w_0 = 0$ and $w_1 = y(t_1) = 1 - e^{-0.1}$.

c. Repeat part (b) with $h = 0.01$ and $w_1 = 1 - e^{-0.01}$.

d. Analyze this method for consistency, stability, and convergence.

5. Given the multistep method

$$w_{i+1} = -\frac{3}{2} w_i + 3w_{i-1} - \frac{1}{2} w_{i-2} + 3hf(t_i, w_i), \quad \text{for } i = 2, \ldots, N-1,$$

with starting values w_0, w_1, w_2:

a. Find the local truncation error.

b. Comment on consistency, stability, and convergence.

6. Obtain an approximate solution to the differential equation

$$y' = -y, \quad 0 \le t \le 10, \quad y(0) = 1$$

using Milne's method with $h = 0.1$ and then $h = 0.01$, with starting values $w_0 = 1$ and $w_1 = e^{-h}$ in both cases. How does decreasing h from $h = 0.1$ to $h = 0.01$ affect the number of correct digits in the approximate solutions at $t = 1$ and $t = 10$?

7. Investigate stability for the difference method

$$w_{i+1} = -4w_i + 5w_{i-1} + 2h[f(t_i, w_i) + 2hf(t_{i-1}, w_{i-1})],$$

for $i = 1, 2, \ldots, N-1$, with starting values w_0, w_1.

8. Consider the problem $y' = 0$, $0 \le t \le 10$, $y(0) = 0$, which has the solution $y \equiv 0$. If the difference method of Exercise 4 is applied to the problem, then

$$w_{i+1} = 4w_i - 3w_{i-1}, \quad \text{for } i = 1, 2, \ldots, N-1,$$
$$w_0 = 0, \quad \text{and} \quad w_1 = \alpha_1.$$

Suppose $w_1 = \alpha_1 = \varepsilon$, where ε is a small rounding error. Compute w_i exactly for $i = 2, 3, \ldots, 6$ to find how the error ε is propagated.

5.11 Stiff Differential Equations

All the methods for approximating the solution to initial-value problems have error terms that involve a higher derivative of the solution of the equation. If the derivative can be reasonably bounded, then the method will have a predictable error bound that can be used to estimate the accuracy of the approximation. Even if the derivative grows as the steps increase, the error can be kept in relative control, provided that the solution also grows in magnitude. Problems frequently arise, however, when the magnitude of the derivative increases but the solution does not. In this situation, the error can grow so large that it dominates the calculations. Initial-value problems for which this is likely to occur are called **stiff equations** and are quite common, particularly in the study of vibrations, chemical reactions, and electrical circuits.

Stiff systems derive their name from the motion of spring and mass systems that have large spring constants.

Stiff differential equations are characterized as those whose exact solution has a term of the form e^{-ct}, where c is a large positive constant. This is usually only a part of the solution, called the *transient* solution. The more important portion of the solution is called the *steady-state* solution. The transient portion of a stiff equation will rapidly decay to zero as t increases, but since the nth derivative of this term has magnitude $c^n e^{-ct}$, the derivative does not decay as quickly. In fact, since the derivative in the error term is evaluated not at t, but at a number between zero and t, the derivative terms can increase as t increases–and very rapidly indeed. Fortunately, stiff equations generally can be predicted from the physical problem from which the equation is derived and, with care, the error can be kept under control. The manner in which this is done is considered in this section.

Illustration The system of initial-value problems

$$u_1' = 9u_1 + 24u_2 + 5\cos t - \frac{1}{3}\sin t, \quad u_1(0) = \frac{4}{3}$$
$$u_2' = -24u_1 - 51u_2 - 9\cos t + \frac{1}{3}\sin t, \quad u_2(0) = \frac{2}{3}$$

has the unique solution

$$u_1(t) = 2e^{-3t} - e^{-39t} + \frac{1}{3}\cos t, \quad u_2(t) = -e^{-3t} + 2e^{-39t} - \frac{1}{3}\cos t.$$

The transient term e^{-39t} in the solution causes this system to be stiff. Applying Algorithm 5.7, the Runge-Kutta Fourth-Order Method for Systems, gives results listed in Table 5.22. When $h = 0.05$, stability results and the approximations are accurate. Increasing the step size to $h = 0.1$, however, leads to the disastrous results shown in the table. □

Table 5.22

t	$u_1(t)$	$w_1(t)$ $h = 0.05$	$w_1(t)$ $h = 0.1$	$u_2(t)$	$w_2(t)$ $h = 0.05$	$w_2(t)$ $h = 0.1$
0.1	1.793061	1.712219	−2.645169	−1.032001	−0.8703152	7.844527
0.2	1.423901	1.414070	−18.45158	−0.8746809	−0.8550148	38.87631
0.3	1.131575	1.130523	−87.47221	−0.7249984	−0.7228910	176.4828
0.4	0.9094086	0.9092763	−934.0722	−0.6082141	−0.6079475	789.3540
0.5	0.7387877	9.7387506	−1760.016	−0.5156575	−0.5155810	3520.00
0.6	0.6057094	0.6056833	−7848.550	−0.4404108	−0.4403558	15697.84
0.7	0.4998603	0.4998361	−34989.63	−0.3774038	−0.3773540	69979.87
0.8	0.4136714	0.4136490	−155979.4	−0.3229535	−0.3229078	311959.5
0.9	0.3416143	0.3415939	−695332.0	−0.2744088	−0.2743673	1390664.
1.0	0.2796748	0.2796568	−3099671.	−0.2298877	−0.2298511	6199352.

Although stiffness is usually associated with systems of differential equations, the approximation characteristics of a particular numerical method applied to a stiff system can be predicted by examining the error produced when the method is applied to a simple *test equation*,

$$y' = \lambda y, \quad y(0) = \alpha, \quad \text{where } \lambda < 0. \tag{5.64}$$

The solution to this equation is $y(t) = \alpha e^{\lambda t}$, which contains the transient solution $e^{\lambda t}$. The steady-state solution is zero, so the approximation characteristics of a method are easy to determine. (A more complete discussion of the round-off error associated with stiff systems requires examining the test equation when λ is a complex number with negative real part; see [Ge1], p. 222.)

First consider Euler's method applied to the test equation. Letting $h = (b - a)/N$ and $t_j = jh$, for $j = 0, 1, 2, \ldots, N$, Eq. (5.8) on page 266 implies that

$$w_0 = \alpha, \quad \text{and} \quad w_{j+1} = w_j + h(\lambda w_j) = (1 + h\lambda)w_j,$$

so

$$w_{j+1} = (1 + h\lambda)^{j+1} w_0 = (1 + h\lambda)^{j+1}\alpha, \quad \text{for } j = 0, 1, \ldots, N - 1. \tag{5.65}$$

Since the exact solution is $y(t) = \alpha e^{\lambda t}$, the absolute error is

$$|y(t_j) - w_j| = \left|e^{jh\lambda} - (1 + h\lambda)^j\right| |\alpha| = \left|(e^{h\lambda})^j - (1 + h\lambda)^j\right| |\alpha|,$$

and the accuracy is determined by how well the term $1 + h\lambda$ approximates $e^{h\lambda}$. When $\lambda < 0$, the exact solution $(e^{h\lambda})^j$ decays to zero as j increases, but by Eq.(5.65), the approximation

will have this property only if $|1 + h\lambda| < 1$, which implies that $-2 < h\lambda < 0$. This effectively restricts the step size h for Euler's method to satisfy $h < 2/|\lambda|$.

Suppose now that a round-off error δ_0 is introduced in the initial condition for Euler's method,

$$w_0 = \alpha + \delta_0.$$

At the jth step the round-off error is

$$\delta_j = (1 + h\lambda)^j \delta_0.$$

Since $\lambda < 0$, the condition for the control of the growth of round-off error is the same as the condition for controlling the absolute error, $|1 + h\lambda| < 1$, which implies that $h < 2/|\lambda|$. So

- Euler's method is expected to be stable for

$$y' = \lambda y, \quad y(0) = \alpha, \quad \text{where } \lambda < 0,$$

only if the step size h is less than $2/|\lambda|$.

The situation is similar for other one-step methods. In general, a function Q exists with the property that the difference method, when applied to the test equation, gives

$$w_{i+1} = Q(h\lambda)w_i. \tag{5.66}$$

The accuracy of the method depends upon how well $Q(h\lambda)$ approximates $e^{h\lambda}$, and the error will grow without bound if $|Q(h\lambda)| > 1$. An nth-order Taylor method, for example, will have stability with regard to both the growth of round-off error and absolute error, provided h is chosen to satisfy

$$\left|1 + h\lambda + \frac{1}{2}h^2\lambda^2 + \cdots + \frac{1}{n!}h^n\lambda^n\right| < 1.$$

Exercise 10 examines the specific case when the method is the classical fourth-order Runge-Kutta method,which is essentially a Taylor method of order four.

When a multistep method of the form (5.54) is applied to the test equation, the result is

$$w_{j+1} = a_{m-1}w_j + \cdots + a_0 w_{j+1-m} + h\lambda(b_m w_{j+1} + b_{m-1}w_j + \cdots + b_0 w_{j+1-m}),$$

for $j = m - 1, \ldots, N - 1$, or

$$(1 - h\lambda b_m)w_{j+1} - (a_{m-1} + h\lambda b_{m-1})w_j - \cdots - (a_0 + h\lambda b_0)w_{j+1-m} = 0.$$

Associated with this homogeneous difference equation is a **characteristic polynomial**

$$Q(z, h\lambda) = (1 - h\lambda b_m)z^m - (a_{m-1} + h\lambda b_{m-1})z^{m-1} - \cdots - (a_0 + h\lambda b_0).$$

This polynomial is similar to the characteristic polynomial (5.58), but it also incorporates the test equation. The theory here parallels the stability discussion in Section 5.10.

Suppose $w_0, \ldots, w_{m-1}$ are given, and, for fixed $h\lambda$, let $\beta_1, \ldots, \beta_m$ be the zeros of the polynomial $Q(z, h\lambda)$. If $\beta_1, \ldots, \beta_m$ are distinct, then $c_1, \ldots, c_m$ exist with

$$w_j = \sum_{k=1}^{m} c_k(\beta_k)^j, \quad \text{for } j = 0, \ldots, N. \tag{5.67}$$

If $Q(z, h\lambda)$ has multiple zeros, w_j is similarly defined. (See Eq. (5.63) in Section 5.10.) If w_j is to accurately approximate $y(t_j) = e^{jh\lambda} = (e^{h\lambda})^j$, then all zeros β_k must satisfy $|\beta_k| < 1$;

otherwise, certain choices of α will result in $c_k \neq 0$, and the term $c_k(\beta_k)^j$ will not decay to zero.

Illustration The test differential equation

$$y' = -30y, \quad 0 \le t \le 1.5, \quad y(0) = \frac{1}{3}$$

has exact solution $y = \frac{1}{3}e^{-30t}$. Using $h = 0.1$ for Euler's Algorithm 5.1, Runge-Kutta Fourth-Order Algorithm 5.2, and the Adams Predictor-Corrector Algorithm 5.4, gives the results at $t = 1.5$ in Table 5.23. □

Table 5.23

Exact solution	9.54173×10^{-21}
Euler's method	-1.09225×10^{4}
Runge-Kutta method	3.95730×10^{1}
Predictor-corrector method	8.03840×10^{5}

The inaccuracies in the Illustration are due to the fact that $|Q(h\lambda)| > 1$ for Euler's method and the Runge-Kutta method and that $Q(z, h\lambda)$ has zeros with modulus exceeding 1 for the predictor-corrector method. To apply these methods to this problem, the step size must be reduced. The following definition is used to describe the amount of step-size reduction that is required.

Definition 5.25 The **region *R* of absolute stability** for a one-step method is $R = \{h\lambda \in \mathcal{C} \mid |Q(h\lambda)| < 1\}$, and for a multistep method, it is $R = \{h\lambda \in \mathcal{C} \mid |\beta_k| < 1, \text{ for all zeros } \beta_k \text{ of } Q(z, h\lambda)\}$. ■

Equations (5.66) and (5.67) imply that a method can be applied effectively to a stiff equation only if $h\lambda$ is in the region of absolute stability of the method, which for a given problem places a restriction on the size of h. Even though the exponential term in the exact solution decays quickly to zero, λh must remain within the region of absolute stability throughout the interval of t values for the approximation to decay to zero and the growth of error to be under control. This means that, although h could normally be increased because of truncation error considerations, the absolute stability criterion forces h to remain small. Variable step-size methods are especially vulnerable to this problem because an examination of the local truncation error might indicate that the step size could increase. This could inadvertently result in λh being outside the region of absolute stability.

The region of absolute stability of a method is generally the critical factor in producing accurate approximations for stiff systems, so numerical methods are sought with as large a region of absolute stability as possible. A numerical method is said to be ***A*-stable** if its region R of absolute stability contains the entire left half-plane.

This method is implicit because it involves w_{j+1} on both sides of the equation.

The **Implicit Trapezoidal method**, given by

$$w_0 = \alpha, \tag{5.68}$$

$$w_{j+1} = w_j + \frac{h}{2}\left[f(t_{j+1}, w_{j+1}) + f(t_j, w_j)\right], \quad 0 \le j \le N-1,$$

is an A-stable method (see Exercise 15) and is the only A-stable multistep method. Although the Trapezoidal method does not give accurate approximations for large step sizes, its error will not grow exponentially.

The techniques commonly used for stiff systems are implicit multistep methods. Generally w_{i+1} is obtained by solving a nonlinear equation or nonlinear system iteratively, often by Newton's method. Consider, for example, the Implicit Trapezoidal method

$$w_{j+1} = w_j + \frac{h}{2}[f(t_{j+1}, w_{j+1}) + f(t_j, w_j)].$$

Having computed t_j, t_{j+1}, and w_j, we need to determine w_{j+1}, the solution to

$$F(w) = w - w_j - \frac{h}{2}[f(t_{j+1}, w) + f(t_j, w_j)] = 0. \tag{5.69}$$

To approximate this solution, select $w_{j+1}^{(0)}$, usually as w_j, and generate $w_{j+1}^{(k)}$ by applying Newton's method to (5.69),

$$\begin{aligned} w_{j+1}^{(k)} &= w_{j+1}^{(k-1)} - \frac{F(w_{j+1}^{(k-1)})}{F'(w_{j+1}^{(k-1)})} \\ &= w_{j+1}^{(k-1)} - \frac{w_{j+1}^{(k-1)} - w_j - \frac{h}{2}[f(t_j, w_j) + f(t_{j+1}, w_{j+1}^{(k-1)})]}{1 - \frac{h}{2} f_y(t_{j+1}, w_{j+1}^{(k-1)})} \end{aligned}$$

until $|w_{j+1}^{(k)} - w_{j+1}^{(k-1)}|$ is sufficiently small. This is the procedure that is used in Algorithm 5.8. Normally only three or four iterations per step are required, because of the quadratic convergence of Newton's mehod.

The Secant method can be used as an alternative to Newton's method in Eq. (5.69), but then two distinct initial approximations to w_{j+1} are required. To employ the Secant method, the usual practice is to let $w_{j+1}^{(0)} = w_j$ and obtain $w_{j+1}^{(1)}$ from some explicit multistep method. When a system of stiff equations is involved, a generalization is required for either Newton's or the Secant method. These topics are considered in Chapter 10.

ALGORITHM
5.8

Trapezoidal with Newton Iteration

To approximate the solution of the initial-value problem

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

at $(N + 1)$ equally spaced numbers in the interval $[a, b]$:

INPUT endpoints a, b; integer N; initial condition α; tolerance TOL; maximum number of iterations M at any one step.

OUTPUT approximation w to y at the $(N + 1)$ values of t or a message of failure.

Step 1 Set $h = (b - a)/N$;
$t = a$;
$w = \alpha$;
OUTPUT (t, w).

Step 2 For $i = 1, 2, \ldots, N$ do Steps 3–7.

Step 3 Set $k_1 = w + \frac{h}{2} f(t, w)$;
$w_0 = k_1$;
$j = 1$;
$FLAG = 0$.

Step 4 While $FLAG = 0$ do Steps 5–6.

Step 5 Set $w = w_0 - \dfrac{w_0 - \frac{h}{2} f(t+h, w_0) - k_1}{1 - \frac{h}{2} f_y(t+h, w_0)}$.

Step 6 If $|w - w_0| < TOL$ then set $FLAG = 1$
else set $j = j + 1$;
$w_0 = w$;
if $j > M$ then
OUTPUT ('The maximum number of iterations exceeded');
STOP.

Step 7 Set $t = a + ih$;
OUTPUT (t, w).

Step 8 STOP. ■

Illustration The stiff initial-value problem

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

has solution $y(t) = t - e^{-5t}$. To show the effects of stiffness, the Implicit Trapezoidal method and the Runge-Kutta fourth-order method are applied both with $N = 4$, giving $h = 0.25$, and with $N = 5$, giving $h = 0.20$.

The Trapezoidal method performs well in both cases using $M = 10$ and $TOL = 10^{-6}$, as does Runge-Kutta with $h = 0.2$. However, $h = 0.25$ is outside the region of absolute stability of the Runge-Kutta method, which is evident from the results in Table 5.24. □

Table 5.24

	Runge–Kutta Method		Trapezoidal Method	
	$h = 0.2$		$h = 0.2$	
t_i	w_i	$\lvert y(t_i) - w_i \rvert$	w_i	$\lvert y(t_i) - w_i \rvert$
0.0	−1.0000000	0	−1.0000000	0
0.2	−0.1488521	1.9027×10^{-2}	−0.1414969	2.6383×10^{-2}
0.4	0.2684884	3.8237×10^{-3}	0.2748614	1.0197×10^{-2}
0.6	0.5519927	1.7798×10^{-3}	0.5539828	3.7700×10^{-3}
0.8	0.7822857	6.0131×10^{-4}	0.7830720	1.3876×10^{-3}
1.0	0.9934905	2.2845×10^{-4}	0.9937726	5.1050×10^{-4}
	$h = 0.25$		$h = 0.25$	
t_i	w_i	$\lvert y(t_i) - w_i \rvert$	w_i	$\lvert y(t_i) - w_i \rvert$
0.0	−1.0000000	0	−1.0000000	0
0.25	0.4014315	4.37936×10^{-1}	0.0054557	4.1961×10^{-2}
0.5	3.4374753	3.01956×10^{0}	0.4267572	8.8422×10^{-3}
0.75	1.44639×10^{23}	1.44639×10^{23}	0.7291528	2.6706×10^{-3}
1.0	Overflow		0.9940199	7.5790×10^{-4}

We have presented here only brief introduction to what the reader frequently encountering stiff differential equations should know. For further details, consult [Ge2], [Lam], or [SGe].

EXERCISE SET 5.11

1. Solve the following stiff initial-value problems using Euler's method, and compare the results with the actual solution.

a. $y' = -9y, \quad 0 \le t \le 1, \quad y(0) = e,$ with $h = 0.1;$ actual solution $y(t) = e^{1-9t}$.

b. $y' = -20(y-t^2)+2t, \quad 0 \le t \le 1, \quad y(0) = \frac{1}{3},$ with $h = 0.1;$ actual solution $y(t) = t^2+\frac{1}{3}e^{-20t}$.

c. $y' = -20y + 20\sin t + \cos t, \quad 0 \le t \le 2, \quad y(0) = 1,$ with $h = 0.25;$ actual solution $y(t) = \sin t + e^{-20t}$.

d. $y' = 50/y-50y, \quad 0 \le t \le 1, \quad y(0) = \sqrt{2},$ with $h = 0.1;$ actual solution $y(t) = (1+e^{-100t})^{1/2}$.

2. Solve the following stiff initial-value problems using Euler's method, and compare the results with the actual solution.

a. $y' = -5y + 6e^t, \quad 0 \le t \le 1, \quad y(0) = 2,$ with $h = 0.1;$ actual solution $y(t) = e^{-5t} + e^t$.

b. $y' = -10y+10t+1, \quad 0 \le t \le 1, \quad y(0) = e,$ with $h = 0.1;$ actual solution $y(t) = e^{-10t+1}+t$.

c. $y' = -15(y - t^{-3}) - 3/t^4, \quad 1 \le t \le 3, \quad y(1) = 0,$ with $h = 0.25;$ actual solution $y(t) = -e^{-15t} + t^{-3}$.

d. $y' = -20y + 20\cos t - \sin t, \quad 0 \le t \le 2, \quad y(0) = 0,$ with $h = 0.25;$ actual solution $y(t) = -e^{-20t} + \cos t$.

3. Repeat Exercise 1 using the Runge-Kutta fourth-order method.

4. Repeat Exercise 2 using the Runge-Kutta fourth-order method.

5. Repeat Exercise 1 using the Adams fourth-order predictor-corrector method.

6. Repeat Exercise 2 using the Adams fourth-order predictor-corrector method.

7. Repeat Exercise 1 using the Trapezoidal Algorithm with $TOL = 10^{-5}$.

8. Repeat Exercise 2 using the Trapezoidal Algorithm with $TOL = 10^{-5}$.

9. Solve the following stiff initial-value problem using the Runge-Kutta fourth-order method with (a) $h = 0.1$ and (b) $h = 0.025$.

$$u_1' = 32u_1 + 66u_2 + \frac{2}{3}t + \frac{2}{3}, \quad 0 \le t \le 0.5, \quad u_1(0) = \frac{1}{3};$$

$$u_2' = -66u_1 - 133u_2 - \frac{1}{3}t - \frac{1}{3}, \quad 0 \le t \le 0.5, \quad u_2(0) = \frac{1}{3}.$$

Compare the results to the actual solution,

$$u_1(t) = \frac{2}{3}t + \frac{2}{3}e^{-t} - \frac{1}{3}e^{-100t} \quad \text{and} \quad u_2(t) = -\frac{1}{3}t - \frac{1}{3}e^{-t} + \frac{2}{3}e^{-100t}.$$

10. Show that the fourth-order Runge-Kutta method,

$$\begin{aligned} k_1 &= hf(t_i, w_i),\\ k_2 &= hf(t_i + h/2, w_i + k_1/2),\\ k_3 &= hf(t_i + h/2, w_i + k_2/2),\\ k_4 &= hf(t_i + h, w_i + k_3),\\ w_{i+1} &= w_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4), \end{aligned}$$

when applied to the differential equation $y' = \lambda y$, can be written in the form

$$w_{i+1} = \left(1 + h\lambda + \frac{1}{2}(h\lambda)^2 + \frac{1}{6}(h\lambda)^3 + \frac{1}{24}(h\lambda)^4\right) w_i.$$

11. Discuss consistency, stability, and convergence for the Implicit Trapezoidal method

$$w_{i+1} = w_i + \frac{h}{2}\left(f(t_{i+1}, w_{i+1}) + f(t_i, w_i)\right), \quad \text{for } i = 0, 1, \ldots, N-1,$$

with $w_0 = \alpha$ applied to the differential equation

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

12. The Backward Euler one-step method is defined by

$$w_{i+1} = w_i + hf(t_{i+1}, w_{i+1}), \quad \text{for } i = 0, \ldots, N-1.$$

Show that $Q(h\lambda) = 1/(1 - h\lambda)$ for the Backward Euler method.

13. Apply the Backward Euler method to the differential equations given in Exercise 1. Use Newton's method to solve for w_{i+1}.

14. Apply the Backward Euler method to the differential equations given in Exercise 2. Use Newton's method to solve for w_{i+1}.

15. **a.** Show that the Implicit Trapezoidal method is A-stable.

b. Show that the Backward Euler method described in Exercise 12 is A-stable.

5.12 Survey of Methods and Software

In this chapter we have considered methods to approximate the solutions to initial-value problems for ordinary differential equations. We began with a discussion of the most elementary numerical technique, Euler's method. This procedure is not sufficiently accurate to be of use in applications, but it illustrates the general behavior of the more powerful techniques, without the accompanying algebraic difficulties. The Taylor methods were then considered as generalizations of Euler's method. They were found to be accurate but cumbersome because of the need to determine extensive partial derivatives of the defining function of the differential equation. The Runge-Kutta formulas simplified the Taylor methods, without increasing the order of the error. To this point we had considered only one-step methods, techniques that use only data at the most recently computed point.

Multistep methods are discussed in Section 5.6, where explicit methods of Adams-Bashforth type and implicit methods of Adams-Moulton type were considered. These culminate in predictor-corrector methods, which use an explicit method, such as an Adams-Bashforth, to predict the solution and then apply a corresponding implicit method, like an Adams-Moulton, to correct the approximation.

Section 5.9 illustrated how these techniques can be used to solve higher-order initial-value problems and systems of initial-value problems.

The more accurate adaptive methods are based on the relatively uncomplicated one-step and multistep techniques. In particular, we saw in Section 5.5 that the Runge-Kutta-Fehlberg method is a one-step procedure that seeks to select mesh spacing to keep the local error of the approximation under control. The Variable Step-Size Predictor-Corrector method presented in Section 5.7 is based on the four-step Adams-Bashforth method and three-step Adams-Moulton method. It also changes the step size to keep the local error within a given tolerance. The Extrapolation method discussed in Section 5.8 is based on a modification of the Midpoint method and incorporates extrapolation to maintain a desired accuracy of approximation.

The final topic in the chapter concerned the difficulty that is inherent in the approximation of the solution to a stiff equation, a differential equation whose exact solution contains a portion of the form $e^{-\lambda t}$, where λ is a positive constant. Special caution must be taken with problems of this type, or the results can be overwhelmed by round-off error.

Methods of the Runge-Kutta-Fehlberg type are generally sufficient for nonstiff problems when moderate accuracy is required. The extrapolation procedures are recommended

for nonstiff problems where high accuracy is required. Extensions of the Implicit Trapezoidal method to variable-order and variable step-size implicit Adams-type methods are used for stiff initial-value problems.

The IMSL Library includes two subroutines for approximating the solutions of initial-value problems. Each of the methods solves a system of m first-order equations in m variables. The equations are of the form

$$\frac{du_i}{dt} = f_i(t, u_1, u_2, \ldots, u_m), \quad \text{for } i = 1, 2, \ldots, m,$$

where $u_i(t_0)$ is given for each i. A variable step-size subroutine is based on the Runge-Kutta-Verner fifth- and sixth-order methods described in Exercise 4 of Section 5.5. A subroutine of Adams type is also available to be used for stiff equations based on a method of C. William Gear. This method uses implicit multistep methods of order up to 12 and backward differentiation formulas of order up to 5.

Runge-Kutta-type procedures contained in the NAG Library are based on the Merson form of the Runge-Kutta method. A variable-order and variable step-size Adams method is also in the library, as well as a variable-order, variable step-size backward-difference method for stiff systems. Other routines incorporate the same methods but iterate until a component of the solution attains a given value or until a function of the solution is zero.

The netlib library includes several subroutines for approximating the solutions of initial-value problems in the package ODE. One subroutine is based on the Runge-Kutta-Verner fifth- and sixth-order methods, another on the Runge-Kutta-Fehlberg fourth- and fifth-order methods as described on page 297 of Section 5.5. A subroutine for stiff ordinary differential equation initial-value problems, is based on a variable coefficient backward differentiation formula.

Many books specialize in the numerical solution of initial-value problems. Two classics are by Henrici [He1] and Gear [Ge1]. Other books that survey the field are by Botha and Pinder [BP], Ortega and Poole [OP], Golub and Ortega [GO], Shampine [Sh], and Dormand [Do].

Two books by Hairer, Nörsett, and Warner provide comprehensive discussions on nonstiff [HNW1] and stiff [HNW2] problems. The book by Burrage [Bur] describes parallel and sequential methods for solving systems of initial-value problems.