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

$$w_1 = w_0 + h_0 f(a, w_0).$$

Copyright 2010 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

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 $O(h^4)$.

Copyright 2010 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

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.

ALGORITHM 5.6

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.

Copyright 2010 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

```
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 $i = 1, 2, \dots, 7$
            for $j = 1, \dots, i$
                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 $j = 1, \dots, NK_k - 1$
                    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: $y_{k-1} \equiv y_{k-1,1}, y_{k-2} \equiv y_{k-2,2}, \dots, y_1 \equiv y_{k-1,k-1}$ 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

                         set $y_{j-1} = y_j + \dfrac{y_j - y_{j-1}}{Q_{k-1,j-1} - 1}$;

                         (Extrapolation to compute $y_{j-1} \equiv y_{k,k-j+2}$.)

                         $\left(\text{Note:} \quad y_{j-1} = \dfrac{h_{j-1}^2 y_j - h_k^2 y_{j-1}}{h_{j-1}^2 - h_k^2}.\right)$

                         $j = j - 1$.

            Step 13  If $|y_1 - v| \le TOL$ then set $NFLAG = 1$.
                         ($y_1$ is accepted as the new $w$.)

        Step 14  Set $k = k + 1$.
```

Copyright 2010 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

```
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

$$y_{11} = \frac{1}{2}(w_2 + w_1 + h_0 f(t_0 + 2h_0, w_2)) = \frac{1}{2}(0.828 + 0.65 + 0.1f(0.2, 0.828)) = 0.8284.$$

For the second approximation to $y(0.2)$ we compute

$$h_1 = h/4 = 0.05;$$

$$w_1 = w_0 + h_1 f(t_0, w_0) = 0.5 + 0.05(1.5) = 0.575;$$

$$w_2 = w_0 + 2h_1 f(t_0 + h_1, w_1) = 0.5 + 0.1(1.5725) = 0.65725;$$

$$w_3 = w_1 + 2h_1 f(t_0 + 2h_1, w_2) = 0.575 + 0.1(1.64725) = 0.739725;$$

$$w_4 = w_2 + 2h_1 f(t_0 + 3h_1, w_3) = 0.65725 + 0.1(1.717225) = 0.8289725.$$

Copyright 2010 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Then the endpoint correction approximation is

$$\begin{aligned} y_{21} &= \frac{1}{2}(w_4 + w_3 + h_1 f(t_0 + 4h_1, w_4)) \\ &= \frac{1}{2}(0.8289725 + 0.739725 + 0.05 f(0.2, 0.8289725)) = 0.8290730625. \end{aligned}$$

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

$$\begin{aligned} h_2 &= h/6 = 0.0\overline{3}; \\ w_1 &= w_0 + h_2 f(t_0, w_0) = 0.55; \\ w_2 &= w_0 + 2h_2 f(t_0 + h_2, w_1) = 0.6032592593; \\ w_3 &= w_1 + 2h_2 f(t_0 + 2h_2, w_2) = 0.6565876543; \\ w_4 &= w_2 + 2h_2 f(t_0 + 3h_2, w_3) = 0.7130317696; \\ w_5 &= w_3 + 2h_2 f(t_0 + 4h_2, w_4) = 0.7696045871; \\ w_6 &= w_4 + 2h_2 f(t_0 + 5h_2, w_4) = 0.8291535569; \end{aligned}$$

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. ■

Copyright 2010 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

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 $y(t) = t + 1/(1 - t)$.

Copyright 2010 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

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.

Copyright 2010 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Definition 5.16 The function $f(t, y_1, \dots, y_m)$, defined on the set

$$D = \{(t, u_1, \dots, u_m) \mid a \le t \le b \text{ and } -\infty < u_i < \infty, \text{ for each } i = 1, 2, \dots, m\}$$

is said to satisfy a **Lipschitz condition** on D in the variables $u_1, u_2, \dots, u_m$ if a constant $L > 0$ exists with

$$|f(t, u_1, \dots, u_m) - f(t, z_1, \dots, z_m)| \le L \sum_{j=1}^{m} |u_j - z_j|, \tag{5.47}$$

for all $(t, u_1, \dots, u_m)$ and $(t, z_1, \dots, z_m)$ 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

$$\left| \frac{\partial f(t, u_1, \dots, u_m)}{\partial u_i} \right| \le L,$$

for each $i = 1, 2, \dots, m$ and all $(t, u_1, \dots, u_m)$ 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

$$D = \{(t, u_1, u_2, \dots, u_m) \mid a \le t \le b \text{ and } -\infty < u_i < \infty, \text{ for each } i = 1, 2, \dots, m\},$$

and let $f_i(t, u_1, \dots, u_m)$, for each $i = 1, 2, \dots, m$, 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 $u_1(t), \dots, u_m(t)$, 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

$$\begin{aligned}
w_0 &= \alpha, \\
k_1 &= hf(t_i, w_i), \\
k_2 &= hf\left(t_i + \frac{h}{2}, w_i + \frac{1}{2}k_1\right), \\
k_3 &= hf\left(t_i + \frac{h}{2}, w_i + \frac{1}{2}k_2\right), \\
k_4 &= hf(t_{i+1}, w_i + k_3), \\
w_{i+1} &= w_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4), \quad \text{for each } i = 0, 1, \dots, N-1,
\end{aligned}$$

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.

Copyright 2010 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

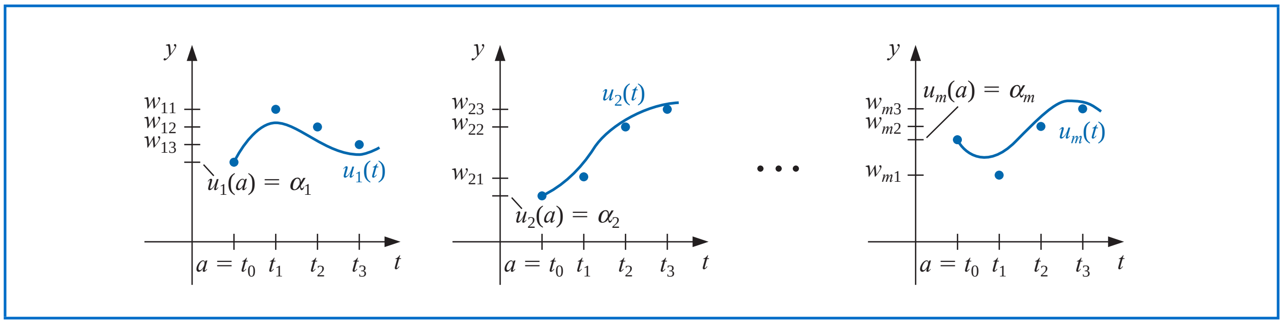
Let an integer $N > 0$ be chosen and set $h = (b - a)/N$. Partition the interval $[a, b]$ into N subintervals with the mesh points

$$t_j = a + jh, \quad \text{for each } j = 0, 1, \ldots, N.$$

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.

Copyright 2010 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

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

$$u_j' = f_j(t, u_1, u_2, \dots, u_m), \quad a \le t \le b, \quad \text{with} \quad u_j(a) = \alpha_j,$$

for $j = 1, 2, \dots, m$ at $(N+1)$ equally spaced numbers in the interval $[a, b]$:

INPUT endpoints a, b; number of equations m; integer N; initial conditions $\alpha_1, \dots, \alpha_m$.

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 $j = 1, 2, \dots, m$ set $w_j = \alpha_j$.
Step 3  OUTPUT $(t, w_1, w_2, \dots, w_m)$.
Step 4  For $i = 1, 2, \dots, N$ do steps 5–11.
    Step 5  For $j = 1, 2, \dots, m$ set
            $k_{1,j} = hf_j(t, w_1, w_2, \dots, w_m)$.
    Step 6  For $j = 1, 2, \dots, m$ set
            $k_{2,j} = hf_j\left(t + \frac{h}{2}, w_1 + \frac{1}{2}k_{1,1}, w_2 + \frac{1}{2}k_{1,2}, \dots, w_m + \frac{1}{2}k_{1,m}\right)$.
    Step 7  For $j = 1, 2, \dots, m$ set
            $k_{3,j} = hf_j\left(t + \frac{h}{2}, w_1 + \frac{1}{2}k_{2,1}, w_2 + \frac{1}{2}k_{2,2}, \dots, w_m + \frac{1}{2}k_{2,m}\right)$.
    Step 8  For $j = 1, 2, \dots, m$ set
            $k_{4,j} = hf_j(t + h, w_1 + k_{3,1}, w_2 + k_{3,2}, \dots, w_m + k_{3,m})$.
    Step 9  For $j = 1, 2, \dots, m$ 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 $(t, w_1, w_2, \dots, w_m)$.
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

$$LI'(t) + RI(t) + \frac{1}{C}\int I(t)\,dt = E(t).$$

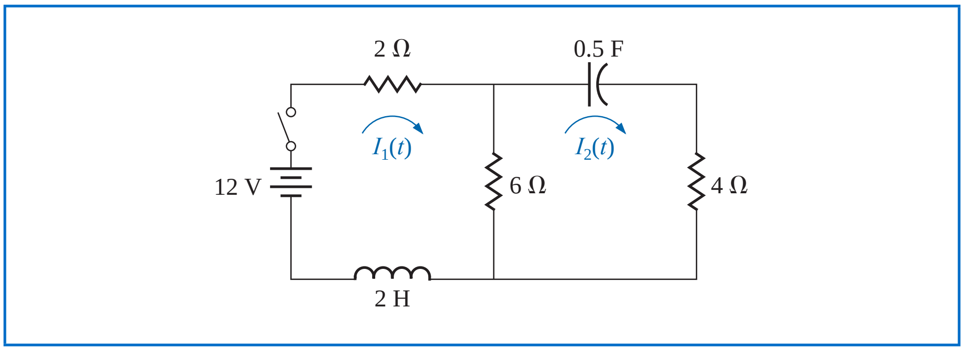
The currents $I_1(t)$ and $I_2(t)$ in the left and right loops, respectively, of the circuit shown in Figure 5.7 are the solutions to the system of equations

$$2I_1(t) + 6[I_1(t) - I_2(t)] + 2I_1'(t) = 12,$$

$$\frac{1}{0.5}\int I_2(t)\,dt + 4I_2(t) + 6[I_2(t) - I_1(t)] = 0.$$

Copyright 2010 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

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

$$k_{3,1} = (0.1)f_1(0.05, 0.267, 0.1584) = 0.54072,$$
$$k_{3,2} = (0.1)f_2(0.05, 0.267, 0.1584) = 0.321264,$$
$$k_{4,1} = (0.1)f_1(0.1, 0.54072, 0.321264) = 0.4800912,$$
$$k_{4,2} = (0.1)f_2(0.1, 0.54072, 0.321264) = 0.28162944.$$

Copyright 2010 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

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

$evalf\,(subs(t = 0.5, r1));\ evalf\,(subs(t = 0.5, r2))$

Copyright 2010 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

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,

$g := dsolve(\{sys2, init2\}, \{u1(t), u2(t)\}, numeric)$

returns

$$\mathbf{proc}(x\_rkf45)\ldots\mathbf{end\ proc}$$

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.

Copyright 2010 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Solution Let $u_1(t) = y(t)$ and $u_2(t) = y'(t)$. This transforms the second-order equation into the system

$$u_1'(t) = u_2(t),$$
$$u_2'(t) = e^{2t}\sin t - 2u_1(t) + 2u_2(t),$$

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

$$k_{1,1} = hf_1(t_0, w_{1,0}, w_{2,0}) = hw_{2,0} = -0.06,$$
$$k_{1,2} = hf_2(t_0, w_{1,0}, w_{2,0}) = h\left[e^{2t_0}\sin t_0 - 2w_{1,0} + 2w_{2,0}\right] = -0.04,$$
$$k_{2,1} = hf_1\left(t_0 + \frac{h}{2}, w_{1,0} + \frac{1}{2}k_{1,1}, w_{2,0} + \frac{1}{2}k_{1,2}\right) = h\left[w_{2,0} + \frac{1}{2}k_{1,2}\right] = -0.062,$$
$$k_{2,2} = hf_2\left(t_0 + \frac{h}{2}, w_{1,0} + \frac{1}{2}k_{1,1}, w_{2,0} + \frac{1}{2}k_{1,2}\right)$$
$$= h\left[e^{2(t_0+0.05)}\sin(t_0 + 0.05) - 2\left(w_{1,0} + \frac{1}{2}k_{1,1}\right) + 2\left(w_{2,0} + \frac{1}{2}k_{1,2}\right)\right]$$
$$= -0.03247644757,$$
$$k_{3,1} = h\left[w_{2,0} + \frac{1}{2}k_{2,2}\right] = -0.06162832238,$$
$$k_{3,2} = h\left[e^{2(t_0+0.05)}\sin(t_0 + 0.05) - 2\left(w_{1,0} + \frac{1}{2}k_{2,1}\right) + 2\left(w_{2,0} + \frac{1}{2}k_{2,2}\right)\right]$$
$$= -0.03152409237,$$
$$k_{4,1} = h\left[w_{2,0} + k_{3,2}\right] = -0.06315240924,$$

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 $u_2(t) = u_1'(t) = 0.2e^{2t}(4\sin t - 3\cos t)$. ■

Copyright 2010 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

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

$def2 := (D@@2)(y)(t) - 2D(y)(t) + 2y(t) = e^{2t}\sin(t)$

and to specify the initial conditions use

$init2 := y(0) = -0.4, D(y)(0) = -0.6$

The solution is obtained with the command

$sol2 := dsolve(\{def2, init2\}, y(t))$

to obtain

$$y(t) = \frac{1}{5}e^{2t}(\sin(t) - 2\cos(t))$$

We isolate the solution in function form using

$g := rhs(sol2)$

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.)

Copyright 2010 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

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 $u_1(t) = -\dfrac{1}{2}e^{2t} + t^2 + 2t - \dfrac{1}{2}$ and $u_2(t) = \dfrac{1}{2}e^{2t} + t^2 - \dfrac{1}{2}$.

b. $u_1' = \dfrac{1}{9}u_1 - \dfrac{2}{3}u_2 - \dfrac{1}{9}t^2 + \dfrac{2}{3}, \quad u_1(0) = -3;$
$u_2' = u_2 + 3t - 4, \quad u_2(0) = 5; \quad 0 \le t \le 2; \quad h = 0.2;$
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 $u_1(t) = -3e^{-t} - 3\sin t + 6\cos t$, $u_2(t) = \dfrac{3}{2}e^{-t} + \dfrac{3}{10}\sin t - \dfrac{21}{10}\cos t - \dfrac{2}{5}e^{2t}$, and $u_3(t) = -e^{-t} + \dfrac{12}{5}\cos t + \dfrac{9}{5}\sin t - \dfrac{2}{5}e^{2t}$.

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 $y(t) = -t^2 + t\cos(\ln t) + t\sin(\ln t) + t^3 \ln t$.

Copyright 2010 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

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. $y'' - 3y' + 2y = 6e^{-t}$, $0 \le t \le 1$, $y(0) = y'(0) = 2$, with $h = 0.1$; actual solution $y(t) = 2e^{2t} - e^t + e^{-t}$.

   b. $t^2y'' + ty' - 4y = -3t$, $1 \le t \le 3$, $y(1) = 4$, $y'(1) = 3$, with $h = 0.2$; actual solution $y(t) = 2t^2 + t + t^{-2}$.

   c. $y''' + y'' - 4y' - 4y = 0$, $0 \le t \le 2$, $y(0) = 3$, $y'(0) = -1$, $y''(0) = 9$, with $h = 0.2$; actual solution $y(t) = e^{-t} + e^{2t} + e^{-2t}$.

   d. $t^3y''' + t^2y'' - 2ty' + 2y = 8t^3 - 2$, $1 \le t \le 2$, $y(1) = 2$, $y'(1) = 8$, $y''(1) = 6$, 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. $\dfrac{d^2\theta}{dt^2} + \dfrac{g}{L}\sin\theta = 0$, $\theta(0) = \dfrac{\pi}{6}$, $\theta'(0) = 0$,

   b. $\dfrac{d^2\theta}{dt^2} + \dfrac{g}{L}\theta = 0$, $\theta(0) = \dfrac{\pi}{6}$, $\theta'(0) = 0$,

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

$$\frac{dx_1(t)}{dt} = x_1(t)[4 - 0.0003x_1(t) - 0.0004x_2(t)] \quad \text{and} \quad \frac{dx_2(t)}{dt} = x_2(t)[2 - 0.0002x_1(t) - 0.0001x_2(t)].$$

Copyright 2010 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

If it is known that the initial population of each species is 10,000, find the solution to this system for $0 \leq t \leq 4$. 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

$$\lim_{h\to 0} \max_{1\leq i\leq N} |\tau_i(h)| = 0. \qquad ■$$

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

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.

$$\lim_{h\to 0} \max_{1\leq i\leq N} |w_i - y(t_i)| = 0,$$

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. ■

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,

$$\max_{1\leq i\leq N} |w_i - y(t_i)| \leq \frac{Mh}{2L}|e^{L(b-a)} - 1|.$$

Copyright 2010 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

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

$$\phi(t, y, 0) = f(t, y), \quad \text{for all } a \le t \le b;$$

Copyright 2010 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

(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

$$|y(t_i) - w_i| \le \frac{\tau(h)}{L} e^{L(t_i - a)}. \qquad \blacksquare$$

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

$$\phi(t, w, h) - \phi(t, \overline{w}, h) = \frac{1}{2} f(t, w) + \frac{1}{2} f(t + h, w + hf(t, w))$$
$$- \frac{1}{2} f(t, \overline{w}) - \frac{1}{2} f(t + h, \overline{w} + hf(t, \overline{w})),$$

the Lipschitz condition on f leads to

$$\begin{aligned}
|\phi(t, w, h) - \phi(t, \overline{w}, h)| &\le \frac{1}{2} L|w - \overline{w}| + \frac{1}{2} L\left|w + hf(t, w) - \overline{w} - hf(t, \overline{w})\right| \\
&\le L|w - \overline{w}| + \frac{1}{2} L\left|hf(t, w) - hf(t, \overline{w})\right| \\
&\le L|w - \overline{w}| + \frac{1}{2} hL^2|w - \overline{w}| \\
&= \left(L + \frac{1}{2} hL^2\right)|w - \overline{w}|.
\end{aligned}$$

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 $O(h^2)$. $\blacksquare$

Copyright 2010 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

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

$$\begin{aligned} \tau_{i+1}(h) = {} & \frac{y(t_{i+1}) - a_{m-1}y(t_i) - \cdots - a_0 y(t_{i+1-m})}{h} \\ & - F(t_i, h, y(t_{i+1}), y(t_i), \ldots, y(t_{i+1-m})), \end{aligned}$$

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.

Copyright 2010 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

- 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 $\sigma(h) = \max_{m\le j\le N} |\sigma_j(h)|$. ■

Copyright 2010 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

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

Copyright 2010 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

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

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

$$w_{i+1} = a_{m-1}w_i + a_{m-2}w_{i-1} + \cdots + a_0 w_{i+1-m} + hF(t_i, h, w_{i+1}, w_i, \ldots, w_{i+1-m}).$$

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

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

$$w_{i+1} = a_{m-1}w_i + a_{m-2}w_{i-1} + \cdots + a_0 w_{i+1-m} + hF(t_i, h, w_{i+1}, w_i, \ldots, w_{i+1-m})$$

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. ■

Copyright 2010 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Example 3 The fourth-order Adams-Bashforth method can be expressed as

$$w_{i+1} = w_i + hF(t_i, h, w_{i+1}, w_i, \dots, w_{i-3}),$$

where

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

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}

Copyright 2010 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

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

$$|u_i - v_i| \leq K|u_0 - v_0|, \quad \text{for each } 1 \leq i \leq N,$$

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

$$|F(t_i, h, w_{i+1}, \ldots, w_{i+1-m}) - F(t_i, h, v_{i+1}, \ldots, v_{i+1-m})| \leq C \sum_{j=0}^{m} |w_{i+1-j} - v_{i+1-j}|.$$

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

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

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

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

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.

Copyright 2010 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

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

$$u_1' = 9u_1 + 24u_2 + 5\cos t - \frac{1}{3}\sin t, \quad u_1(0) = \frac{4}{3}$$
$$u_2' = -24u_1 - 51u_2 - 9\cos t + \frac{1}{3}\sin t, \quad u_2(0) = \frac{2}{3}$$

Copyright 2010 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

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

Copyright 2010 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

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

$$y' = \lambda y, \quad y(0) = \alpha, \qquad \text{where } \lambda < 0,$$

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 $|\beta_k| < 1$;

Copyright 2010 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

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 $R = \{h\lambda \in \mathcal{C} \mid |\beta_k| < 1$, for all zeros β_k 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.

Copyright 2010 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

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 $k_1 = w + \frac{h}{2} f(t, w)$;
                $w_0 = k_1$;
                $j = 1$;
                $FLAG = 0$.
```

Copyright 2010 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

```
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].

Copyright 2010 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

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

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

Copyright 2010 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

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

$$w_{i+1} = w_i + hf(t_{i+1}, w_{i+1}), \quad \text{for } i = 0, \dots, N-1.$$

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

Copyright 2010 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.