```
SUB RK4 (x, y, h, ynew)
  CALL Derivs(x, y, k1)
  ym = y + k1 · h/2
  CALL Derivs(x + h/2, ym, k2)
  ym = y + k2 · h/2
  CALL Derivs(x + h/2, ym, k3)
  ye = y + k3 · h
  CALL Derivs(x + h, ye, k4)
  slope = (k1 + 2(k2 + k3) + k4)/6
  ynew = y + slope · h
  x = x + h
END SUB
```

FIGURE 25.17
Pseudocode to determine a single step of the fourth-order RK method.

costs and the accuracy requirements of the problem also must be considered when choosing a solution technique. Such trade-offs will be explored in detail in the engineering applications in Chap. 28 and in the epilogue for Part Seven.

25.3.5 Computer Algorithms for Runge-Kutta Methods

As with all the methods covered in this chapter, the RK techniques fit nicely into the general algorithm embodied in Fig. 25.7. Figure 25.17 presents pseudocode to determine the slope of the classic fourth-order RK method [Eq. (25.40)]. Subroutines to compute slopes for all the other versions can be easily programmed in a similar fashion.

25.4 SYSTEMS OF EQUATIONS

Many practical problems in engineering and science require the solution of a system of simultaneous ordinary differential equations rather than a single equation. Such systems may be represented generally as

$$\begin{aligned}
\frac{dy_1}{dx} &= f_1(x, y_1, y_2, \ldots, y_n) \\
\frac{dy_2}{dx} &= f_2(x, y_1, y_2, \ldots, y_n) \\
&\vdots \\
\frac{dy_n}{dx} &= f_n(x, y_1, y_2, \ldots, y_n)
\end{aligned} \tag{25.42}$$

The solution of such a system requires that n initial conditions be known at the starting value of x.

25.4.1 Euler's Method

All the methods discussed in this chapter for single equations can be extended to the system shown above. Engineering applications can involve thousands of simultaneous equations. In each case, the procedure for solving a system of equations simply involves applying the one-step technique for every equation at each step before proceeding to the next step. This is best illustrated by the following example for the simple Euler's method.

EXAMPLE 25.9 Solving Systems of ODEs Using Euler's Method

Problem Statement. Solve the following set of differential equations using Euler's method, assuming that at $x = 0$, $y_1 = 4$, and $y_2 = 6$. Integrate to $x = 2$ with a step size of 0.5.

$$\frac{dy_1}{dx} = -0.5y_1 \qquad \frac{dy_2}{dx} = 4 - 0.3y_2 - 0.1y_1$$

Solution. Euler's method is implemented for each variable as in Eq. (25.2):

$$y_1(0.5) = 4 + [-0.5(4)]0.5 = 3$$
$$y_2(0.5) = 6 + [4 - 0.3(6) - 0.1(4)]0.5 = 6.9$$

Note that $y_1(0) = 4$ is used in the second equation rather than the $y_1(0.5) = 3$ computed with the first equation. Proceeding in a like manner gives

x	y_1	y_2
0	4	6
0.5	3	6.9
1.0	2.25	7.715
1.5	1.6875	8.44525
2.0	1.265625	9.094087

25.4.2 Runge-Kutta Methods

Note that any of the higher-order RK methods in this chapter can be applied to systems of equations. However, care must be taken in determining the slopes. Figure 25.15 is helpful in visualizing the proper way to do this for the fourth-order method. That is, we first develop slopes for all variables at the initial value. These slopes (a set of k_1's) are then used to make predictions of the dependent variable at the midpoint of the interval. These midpoint values are in turn used to compute a set of slopes at the midpoint (the k_2's). These new slopes are then taken back to the starting point to make another set of midpoint predictions that lead to new slope predictions at the midpoint (the k_3's). These are then employed to make predictions at the end of the interval that are used to develop slopes at the end of the interval (the k_4's). Finally, the k's are combined into a set of increment functions [as in Eq. (25.40)] and brought back to the beginning to make the final prediction. The following example illustrates the approach.

EXAMPLE 25.10 Solving Systems of ODEs Using the Fourth-Order RK Method

Problem Statement. Use the fourth-order RK method to solve the ODEs from Example 25.9.

Solution. First, we must solve for all the slopes at the beginning of the interval:

$$k_{1,1} = f_1(0, 4, 6) = -0.5(4) = -2$$
$$k_{1,2} = f_2(0, 4, 6) = 4 - 0.3(6) - 0.1(4) = 1.8$$

where $k_{i,j}$ is the ith value of k for the jth dependent variable. Next, we must calculate the first values of y_1 and y_2 at the midpoint:

$$y_1 + k_{1,1}\frac{h}{2} = 4 + (-2)\frac{0.5}{2} = 3.5$$
$$y_2 + k_{1,2}\frac{h}{2} = 6 + (1.8)\frac{0.5}{2} = 6.45$$

which can be used to compute the first set of midpoint slopes,

$$k_{2,1} = f_1(0.25, 3.5, 6.45) = -1.75$$
$$k_{2,2} = f_2(0.25, 3.5, 6.45) = 1.715$$

These are used to determine the second set of midpoint predictions,

$$y_1 + k_{2,1}\frac{h}{2} = 4 + (-1.75)\frac{0.5}{2} = 3.5625$$
$$y_2 + k_{2,2}\frac{h}{2} = 6 + (1.715)\frac{0.5}{2} = 6.42875$$

which can be used to compute the second set of midpoint slopes,

$$k_{3,1} = f_1(0.25, 3.5625, 6.42875) = -1.78125$$
$$k_{3,2} = f_2(0.25, 3.5625, 6.42875) = 1.715125$$

These are used to determine the predictions at the end of the interval

$$y_1 + k_{3,1}h = 4 + (-1.78125)(0.5) = 3.109375$$
$$y_2 + k_{3,2}h = 6 + (1.715125)(0.5) = 6.857563$$

which can be used to compute the endpoint slopes,

$$k_{4,1} = f_1(0.5, 3.109375, 6.857563) = -1.554688$$
$$k_{4,2} = f_2(0.5, 3.109375, 6.857563) = 1.631794$$

The values of k can then be used to compute [Eq. (25.40)]:

$$y_1(0.5) = 4 + \frac{1}{6}[-2 + 2(-1.75 - 1.78125) - 1.554688]0.5 = 3.115234$$

$$y_2(0.5) = 6 + \frac{1}{6}[1.8 + 2(1.715 + 1.715125) + 1.631794]0.5 = 6.857670$$

Proceeding in a like manner for the remaining steps yields

x	y_1	y_2
0	4	6
0.5	3.115234	6.857670
1.0	2.426171	7.632106
1.5	1.889523	8.326886
2.0	1.471577	8.946865

25.4.3 Computer Algorithm for Solving Systems of ODEs

The computer code for solving a single ODE with Euler's method (Fig. 25.7) can be easily extended to systems of equations. The modifications include:

1. Inputting the number of equations, n.
2. Inputting the initial values for each of the n dependent variables.
3. Modifying the algorithm so that it computes slopes for each of the dependent variables.
4. Including additional equations to compute derivative values for each of the ODEs.
5. Including loops to compute a new value for each dependent variable.

Such an algorithm is outlined in Fig. 25.18 for the fourth-order RK method. Notice how similar it is in structure and organization to Fig. 25.7. Most of the differences relate to the fact that

1. There are n equations.
2. The added detail of the fourth-order RK method.

EXAMPLE 25.11 Solving Systems of ODEs with the Computer

Problem Statement. A computer program to implement the fourth-order RK method for systems can be easily developed based on Fig. 25.18. Such software makes it convenient to compare different models of a physical system. For example, a linear model for a swinging pendulum is given by [recall Eq. (PT7.11)]

$$\frac{dy_1}{dx} = y_2 \qquad \frac{dy_2}{dx} = -16.1y_1$$

where y_1 and y_2 = angular displacement and velocity. A nonlinear model of the same system is [recall Eq. (PT7.9)]

$$\frac{dy_3}{dx} = y_4 \qquad \frac{dy_4}{dx} = -16.1\sin(y_3)$$

where y_3 and y_4 = angular displacement and velocity for the nonlinear case. Solve these systems for two cases: (a) a small initial displacement ($y_1 = y_3 = 0.1$ radians; $y_2 = y_4 = 0$) and (b) a large displacement ($y_1 = y_3 = \pi/4 = 0.785398$ radians; $y_2 = y_4 = 0$).

(a) Main or "Driver" Program

```
Assign values for
n = number of equations
yi = initial values of n dependent
     variables
xi = initial value independent
     variable
xf = final value independent variable
dx = calculation step size
xout = output interval

x = xi
m = 0
xp_m = x
DOFOR i = 1, n
  yp_i,m = yi_i
  y_i = yi_i
END DO
DO
  xend = x + xout
  IF (xend > xf) THEN xend = xf
  h = dx
  CALL Integrator (x, y, n, h, xend)
  m = m + 1
  xp_m = x
  DOFOR i = 1, n
    yp_i,m = y_i
  END DO
  IF (x ≥ xf) EXIT
END DO
DISPLAY RESULTS
END
```

(b) Routine to Take One Output Step

```
SUB Integrator (x, y, n, h, xend)
  DO
    IF (xend − x < h) THEN h = xend − x
    CALL RK4 (x, y, n, h)
    IF (x ≥ xend) EXIT
  END DO
END SUB
```

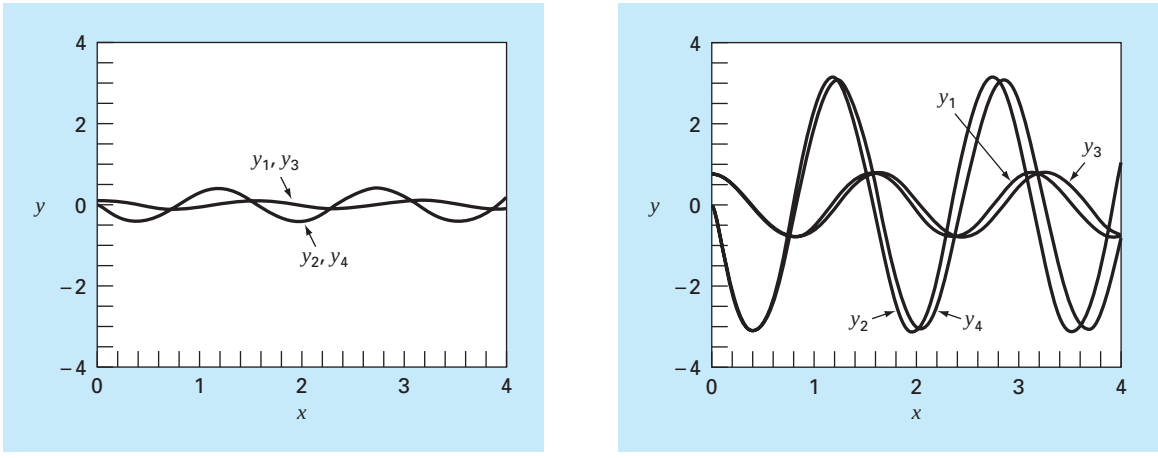
(c) Fourth-Order RK Method for a System of ODEs

```
SUB RK4 (x, y, n, h)
  CALL Derivs (x, y, k1)
  DOFOR i = 1, n
    ym_i = y_i + k1_i * h / 2
  END DO
  CALL Derivs (x + h / 2, ym, k2)
  DOFOR i = 1, n
    ym_i = y_i + k2_i * h / 2
  END DO
  CALL Derivs (x + h / 2, ym, k3)
  DOFOR i = 1, n
    ye_i = y_i + k3_i * h
  END DO
  CALL Derivs (x + h, ye, k4)
  DOFOR i = 1, n
    slope_i = (k1_i + 2*(k2_i+k3_i)+k4_i)/6
    y_i = y_i + slope_i * h
  END DO
  x = x + h
END SUB
```

(d) Routine to Determine Derivatives

```
SUB Derivs (x, y, dy)
  dy_1 = . . .
  dy_2 = . . .
END SUB
```

FIGURE 25.18
Pseudocode for the fourth-order RK method for systems.



(*a*) (*b*)

FIGURE 25.19
Solutions obtained with a computer program for the fourth-order RK method. The plots represent solutions for both linear and nonlinear pendulums with (*a*) small and (*b*) large initial displacements.

Solution.

(a) The calculated results for the linear and nonlinear models are almost identical (Fig. 25.19*a*). This is as expected because when the initial displacement is small, $\sin(\theta) \cong \theta$.

(b) When the initial displacement is $\pi/4 = 0.785398$, the solutions are much different and the difference is magnified as time becomes larger and larger (Fig. 25.19*b*). This is expected because the assumption that $\sin(\theta) = \theta$ is poor when theta is large.

25.5 ADAPTIVE RUNGE-KUTTA METHODS

To this point, we have presented methods for solving ODEs that employ a constant step size. For a significant number of problems, this can represent a serious limitation. For example, suppose that we are integrating an ODE with a solution of the type depicted in Fig. 25.20. For most of the range, the solution changes gradually. Such behavior suggests that a fairly large step size could be employed to obtain adequate results. However, for a localized region from $x = 1.75$ to $x = 2.25$, the solution undergoes an abrupt change. The practical consequence of dealing with such functions is that a very small step size would be required to accurately capture the impulsive behavior. If a constant step-size algorithm were employed, the smaller step size required for the region of abrupt change would have to be applied to the entire computation. As a consequence, a much smaller step size than necessary—and, therefore, many more calculations—would be wasted on the regions of gradual change.

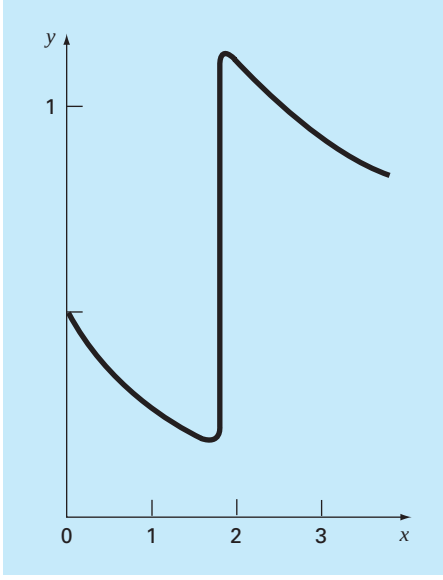


FIGURE 25.20
An example of a solution of an ODE that exhibits an abrupt change. Automatic step-size adjustment has great advantages for such cases.

Algorithms that automatically adjust the step size can avoid such overkill and hence be of great advantage. Because they "adapt" to the solution's trajectory, they are said to have *adaptive step-size control*. Implementation of such approaches requires that an estimate of the local truncation error be obtained at each step. This error estimate can then serve as a basis for either lengthening or decreasing the step size.

Before proceeding, we should mention that aside from solving ODEs, the methods described in this chapter can also be used to evaluate definite integrals. As mentioned previously in the introduction to Part Six, the evaluation of the integral

$$I = \int_a^b f(x)\,dx$$

is equivalent to solving the differential equation

$$\frac{dy}{dx} = f(x)$$

for $y(b)$ given the initial condition $y(a) = 0$. Thus, the following techniques can be employed to efficiently evaluate definite integrals involving functions that are generally smooth but exhibit regions of abrupt change.

There are two primary approaches to incorporate adaptive step-size control into one-step methods. In the first, the error is estimated as the difference between two predictions using the same-order RK method but with different step sizes. In the second, the local

truncation error is estimated as the difference between two predictions using different-order RK methods.

25.5.1 Adaptive RK or Step-Halving Method

Step halving (also called *adaptive RK*) involves taking each step twice, once as a full step and independently as two half steps. The difference in the two results represents an estimate of the local truncation error. If y_1 designates the single-step prediction and y_2 designates the prediction using the two half steps, the error Δ can be represented as

$$\Delta = y_2 - y_1 \tag{25.43}$$

In addition to providing a criterion for step-size control, Eq. (25.43) can also be used to correct the y_2 prediction. For the fourth-order RK version, the correction is

$$y_2 \leftarrow y_2 + \frac{\Delta}{15} \tag{25.44}$$

This estimate is fifth-order accurate.

EXAMPLE 25.12 Adaptive Fourth-Order RK Method

Problem Statement. Use the adaptive fourth-order RK method to integrate $y' = 4e^{0.8x} - 0.5y$ from $x = 0$ to 2 using $h = 2$ and an initial condition of $y(0) = 2$. This is the same differential equation that was solved previously in Example 25.5. Recall that the true solutions is $y(2) = 14.84392$.

Solution. The single prediction with a step of h is computed as

$$y(2) = 2 + \frac{1}{6}[3 + 2(6.40216 + 4.70108) + 14.11105]2 = 15.10584$$

The two half-step predictions are

$$y(1) = 2 + \frac{1}{6}[3 + 2(4.21730 + 3.91297) + 5.945681]1 = 6.20104$$

and

$$y(2) = 6.20104 + \frac{1}{6}[5.80164 + 2(8.72954 + 7.99756) + 12.71283]1 = 14.86249$$

Therefore, the approximate error is

$$E_a = \frac{14.86249 - 15.10584}{15} = -0.01622$$

which compares favorably with the true error of

$$E_t = 14.84392 - 14.86249 = -0.01857$$

The error estimate can also be used to correct the prediction

$$y(2) = 14.86249 - 0.01622 = 14.84627$$

which has an $E_t = -0.00235$.

25.5.2 Runge-Kutta Fehlberg

Aside from step halving as a strategy to adjust step size, an alternative approach for obtaining an error estimate involves computing two RK predictions of different order. The results can then be subtracted to obtain an estimate of the local truncation error. One shortcoming of this approach is that it greatly increases the computational overhead. For example, a fourth- and fifth-order prediction amount to a total of 10 function evaluations per step. The *Runge-Kutta Fehlberg* or *embedded RK* method cleverly circumvents this problem by using a fifth-order RK method that employs the function evaluations from the accompanying fourth-order RK method. Thus, the approach yields the error estimate on the basis of only six function evaluations!

For the present case, we use the following fourth-order estimate

$$y_{i+1} = y_i + \left(\frac{37}{378}k_1 + \frac{250}{621}k_3 + \frac{125}{594}k_4 + \frac{512}{1771}k_6\right)h \tag{25.45}$$

along with the fifth-order formula:

$$y_{i+1} = y_i + \left(\frac{2825}{27{,}648}k_1 + \frac{18{,}575}{48{,}384}k_3 + \frac{13{,}525}{55{,}296}k_4 + \frac{277}{14{,}336}k_5 + \frac{1}{4}k_6\right)h \tag{25.46}$$

where

$$k_1 = f(x_i, y_i)$$

$$k_2 = f\left(x_i + \frac{1}{5}h, y_i + \frac{1}{5}k_1h\right)$$

$$k_3 = f\left(x_i + \frac{3}{10}h, y_i + \frac{3}{40}k_1h + \frac{9}{40}k_2h\right)$$

$$k_4 = f\left(x_i + \frac{3}{5}h, y_i + \frac{3}{10}k_1h - \frac{9}{10}k_2h + \frac{6}{5}k_3h\right)$$

$$k_5 = f\left(x_i + h, y_i - \frac{11}{54}k_1h + \frac{5}{2}k_2h - \frac{70}{27}k_3h + \frac{35}{27}k_4h\right)$$

$$k_6 = f\left(x_i + \frac{7}{8}h, y_i + \frac{1631}{55{,}296}k_1h + \frac{175}{512}k_2h + \frac{575}{13{,}824}k_3h + \frac{44{,}275}{110{,}592}k_4h + \frac{253}{4096}k_5h\right)$$

Thus, the ODE can be solved with Eq. (25.46) and the error estimated as the difference of the fifth- and fourth-order estimates. It should be noted that the particular coefficients used above were developed by Cash and Karp (1990). Therefore, it is sometimes called the *Cash-Karp* RK method.

EXAMPLE 25.13 Runge-Kutta Fehlberg Method

Problem Statement. Use the Cash-Karp version of the Runge-Kutta Fehlberg approach to perform the same calculation as in Example 25.12 from $x = 0$ to 2 using $h = 2$.

Solution. The calculation of the k's can be summarized in the following table:

	x	y	$f(x, y)$
k_1	0	2	3
k_2	0.4	3.2	3.908511
k_3	0.6	4.20883	4.359883
k_4	1.2	7.228398	6.832587
k_5	2	15.42765	12.09831
k_6	1.75	12.17686	10.13237

These can then be used to compute the fourth-order prediction

$$y_1 = 2 + \left(\frac{37}{378}3 + \frac{250}{621}4.359883 + \frac{125}{594}6.832587 + \frac{512}{1771}10.13237\right)2 = 14.83192$$

along with a fifth-order formula:

$$y_1 = 2 + \left(\frac{2825}{27{,}648}3 + \frac{18{,}575}{48{,}384}4.359883 + \frac{13{,}525}{55{,}296}6.832587 + \frac{277}{14{,}336}12.09831 + \frac{1}{4}10.13237\right)2 = 14.83677$$

The error estimate is obtained by subtracting these two equations to give

$$E_a = 14.83677 - 14.83192 = 0.004842$$

25.5.3 Step-Size Control

Now that we have developed ways to estimate the local truncation error, it can be used to adjust the step size. In general, the strategy is to increase the step size if the error is too small and decrease it if the error is too large. Press et al. (1992) have suggested the following criterion to accomplish this:

$$h_{\text{new}} = h_{\text{present}} \left| \frac{\Delta_{\text{new}}}{\Delta_{\text{present}}} \right|^{\alpha} \tag{25.47}$$

where h_{present} and h_{new} = the present and the new step sizes, respectively, Δ_{present} = the computed present accuracy, Δ_{new} = the desired accuracy, and α = a constant power that is equal to 0.2 when the step size is increased (that is, when $\Delta_{\text{present}} \leq \Delta_{\text{new}}$) and 0.25 when the step size is decreased ($\Delta_{\text{present}} > \Delta_{\text{new}}$).

The key parameter in Eq. (25.47) is obviously Δ_{new} because it is your vehicle for specifying the desired accuracy. One way to do this would be to relate Δ_{new} to a relative error level. Although this works well when only positive values occur, it can cause problems for solutions that pass through zero. For example, you might be simulating an oscillating function that repeatedly passes through zero but is bounded by maximum absolute values. For such a case, you might want these maximum values to figure in the desired accuracy.

A more general way to handle such cases is to determine Δ_{new} as

$$\Delta_{new} = \varepsilon y_{scale}$$

where ε = an overall tolerance level. Your choice of y_{scale} will then determine how the error is scaled. For example, if $y_{scale} = y$, the accuracy will be couched in terms of fractional relative errors. If you are dealing with a case where you desire constant errors relative to a prescribed maximum bound, set y_{scale} equal to that bound. A trick suggested by Press et al. (1992) to obtain the constant relative errors except very near zero crossings is

$$y_{scale} = |y| + \left| h\frac{dy}{dx} \right|$$

This is the version we will use in our algorithm.

25.5.4 Computer Algorithm

Figures 25.21 and 25.22 outline pseudocode to implement the Cash-Karp version of the Runge-Kutta Fehlberg algorithm. This algorithm is patterned after a more detailed implementation by Press et al. (1992) for systems of ODEs.

Figure 25.21 implements a single step of the Cash-Karp routine (that is Eqs. 25.45 and 25.46). Figure 25.22 outlines a general driver program along with a subroutine that actually adapts the step size.

FIGURE 25.21
Pseudocode for a single step of the Cash-Karp RK method.

```
SUBROUTINE RKkc (y,dy,x,h,yout,yerr)
PARAMETER (a2=0.2,a3=0.3,a4=0.6,a5=1.,a6=0.875,
  b21=0.2,b31=3./40.,b32=9./40.,b41=0.3,b42=-0.9,
  b43=1.2,b51=-11./54.,b52=2.5,b53=-70./27.,
  b54=35./27.,b61=1631./55296.,b62=175./512.,
  b63=575./13824.,b64=44275./110592.,b65=253./4096.,
  c1=37./378.,c3=250./621.,c4=125./594.,
  c6=512./1771.,dc1=c1-2825./27648.,
  dc3=c3-18575./48384.,dc4=c4-13525./55296.,
  dc5=-277./14336.,dc6=c6-0.25)
ytemp=y+b21*h*dy
CALL Derivs (x+a2*h,ytemp,k2)
ytemp=y+h*(b31*dy+b32*k2)
CALL Derivs(x+a3*h,ytemp,k3)
ytemp=y+h*(b41*dy+b42*k2+b43*k3)
CALL Derivs(x+a4*h,ytemp,k4)
ytemp=y+h*(b51*dy+b52*k2+b53*k3+b54*k4)
CALL Derivs(x+a5*h,ytemp,k5)
ytemp=y+h*(b61*dy+b62*k2+b63*k3+b64*k4+b65*k5)
CALL Derivs(x+a6*h,ytemp,k6)
yout=y+h*(c1*dy+c3*k3+c4*k4+c6*k6)
yerr=h*(dc1*dy+dc3*k3+dc4*k4+dc5*k5+dc6*k6)
END RKkc
```

(a) Driver Program

```
INPUT xi, xf, yi
maxstep=100
hi=.5; tiny = 1. × 10^-30
eps=0.00005
print *, xi,yi
x=xi
y=yi
h=hi
istep=0
DO
  IF (istep > maxstep AND x ≤ xf) EXIT
  istep=istep+1
  CALL Derivs(x,y,dy)
  yscal=ABS(y)+ABS(h*dy)+tiny
  IF (x+h>xf) THEN h=xf-x
  CALL Adapt (x,y,dy,h,yscal,eps,hnxt)
  PRINT x,y
  h=hnxt
END DO
END
```

(b) Adaptive Step Routine

```
SUB Adapt (x,y,dy,htry,yscal,eps,hnxt)
PARAMETER (safety=0.9,econ=1.89e-4)
h=htry
DO
  CALL RKkc(y,dy,x,h,ytemp,yerr)
  emax=abs(yerr/yscal/eps)
  IF emax ≤ 1 EXIT
  htemp=safety*h*emax^-0.25
  h=max(abs(htemp),0.25*abs(h))
  xnew=x+h
  IF xnew = x THEN pause
END DO
IF emax > econ THEN
  hnxt=safety*emax^-.2*h
ELSE
  hnxt=4.*h
END IF
x=x+h
y=ytemp
END Adapt
```

FIGURE 25.22
Pseudocode for a (*a*) driver program and an (*b*) adaptive step routine to solve a single ODE.

EXAMPLE 25.14 Computer Application of an Adaptive Fourth-Order RK Scheme

Problem Statement. The adaptive RK method is well-suited for the following ordinary differential equation

$$\frac{dy}{dx} + 0.6y = 10e^{-(x-2)^2/[2(0.075)^2]} \tag{E25.14.1}$$

Notice for the initial condition, $y(0) = 0.5$, the general solution is

$$y = 0.5e^{-0.6x} \tag{E25.14.2}$$

which is a smooth curve that gradually approaches zero as x increases. In contrast, the particular solution undergoes an abrupt transition in the vicinity of $x = 2$ due to the nature of the forcing function (Fig. 25.23*a*). Use a standard fourth-order RK scheme to solve Eq. (E25.14.1) from $x = 0$ to 4. Then employ the adaptive scheme described in this section to perform the same computation.

Solution. First, the classical fourth-order scheme is used to compute the solid curve in Fig. 25.23*b*. For this computation, a step size of 0.1 is used so that $4/(0.1) = 40$ applications of the technique are made. Then, the calculation is repeated with a step size of 0.05 for a total of 80 applications. The major discrepancy between the two results occurs in the region from 1.8 to 2.0. The magnitude of the discrepancy is about 0.1 to 0.2 percent.

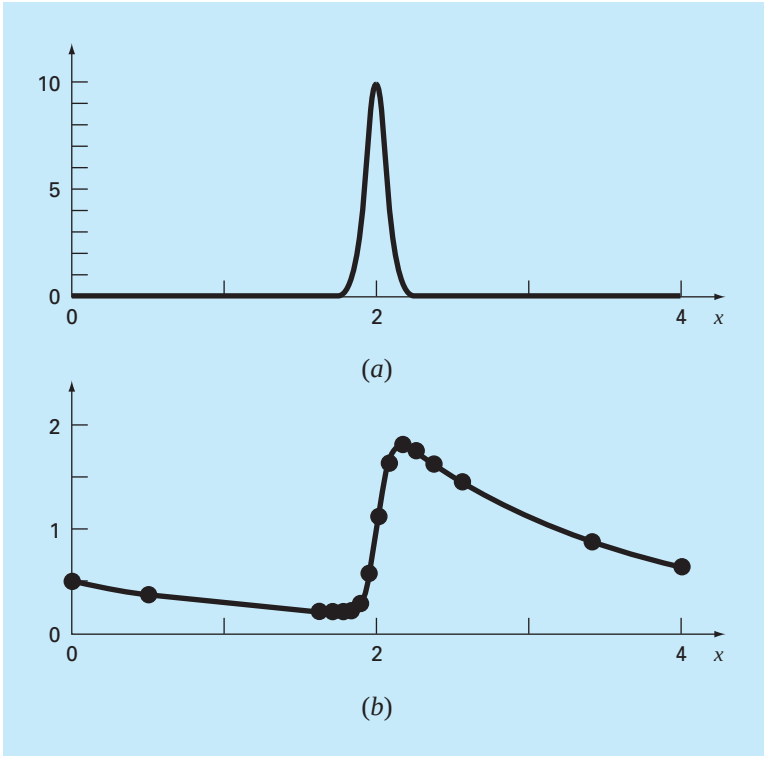


FIGURE 25.23
(*a*) A bell-shaped forcing function that induces an abrupt change in the solution of an ODE [Eq. (E25.14.1)]. (*b*) The solution. The points indicate the predictions of an adaptive step-size routine.

Next, the algorithm in Figs. 25.21 and 25.22 is developed into a computer program and used to solve the same problem. An initial step size of 0.5 and an $\varepsilon = 0.00005$ were chosen. The results were superimposed on Fig. 25.23*b*. Notice how large steps are taken in the regions of gradual change. Then, in the vicinity of $x = 2$, the steps are decreased to accommodate the abrupt nature of the forcing function.

The utility of an adaptive integration scheme obviously depends on the nature of the functions being modeled. It is particularly advantageous for those solutions with long smooth stretches and short regions of abrupt change. In addition, it has utility in those situations where the correct step size is not known a priori. For these cases, an adaptive routine will "feel" its way through the solution while keeping the results within the desired tolerance. Thus, it will tiptoe through the regions of abrupt change and step out briskly when the variations become more gradual.

PROBLEMS

25.1 Solve the following initial value problem over the interval from $t = 0$ to 2 where $y(0) = 1$. Display all your results on the same graph.

$$\frac{dy}{dt} = yt^3 - 1.5y$$

(a) Analytically.
(b) Euler's method with $h = 0.5$ and 0.25.
(c) Midpoint method with $h = 0.5$.
(d) Fourth-order RK method with $h = 0.5$.

25.2 Solve the following problem over the interval from $x = 0$ to 1 using a step size of 0.25 where $y(0) = 1$. Display all your results on the same graph.

$$\frac{dy}{dx} = (1 + 2x)\sqrt{y}$$

(a) Analytically.
(b) Euler's method.
(c) Heun's method without the corrector.
(d) Ralston's method.
(e) Fourth-order RK method.

25.3 Use the **(a)** Euler and **(b)** Heun (without iteration) methods to solve

$$\frac{d^2y}{dt^2} - t + y = 0$$

where $y(0) = 2$ and $y'(0) = 0$. Solve from $x = 0$ to 4 using $h = 0.1$. Compare the methods by plotting the solutions.

25.4 Solve the following problem with the fourth-order RK method:

$$\frac{d^2y}{dx^2} + 0.5\frac{dy}{dx} + 7y = 0$$

where $y(0) = 4$ and $y'(0) = 0$. Solve from $x = 0$ to 5 with $h = 0.5$. Plot your results.

25.5 Solve from $t = 0$ to 3 with $h = 0.1$ using **(a)** Heun (without corrector) and **(b)** Ralston's 2nd-order RK method:

$$\frac{dy}{dt} = y\sin^3(t) \qquad y(0) = 1$$

25.6 Solve the following problem numerically from $t = 0$ to 3:

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

Use the third-order RK method with a step size of 0.5.

25.7 Use **(a)** Euler's and **(b)** the fourth-order RK method to solve

$$\frac{dy}{dx} = -2y + 4e^{-x}$$
$$\frac{dz}{dx} = -\frac{yz^2}{3}$$

over the range $x = 0$ to 1 using a step size of 0.2 with $y(0) = 2$ and $z(0) = 4$.

25.8 Compute the first step of Example 25.14 using the adaptive fourth-order RK method with $h = 0.5$. Verify whether step-size adjustment is in order.

25.9 If $\varepsilon = 0.001$, determine whether step size adjustment is required for Example 25.12.

25.10 Use the RK-Fehlberg approach to perform the same calculation as in Example 25.12 from $x = 0$ to 1 with $h = 1$.

25.11 Write a computer program based on Fig. 25.7. Among other things, place documentation statements throughout the program to identify what each section is intended to accomplish.

25.12 Test the program you developed in Prob. 25.11 by duplicating the computations from Examples 25.1 and 25.4.

25.13 Develop a user-friendly program for the Heun method with an iterative corrector. Test the program by duplicating the results in Table 25.2.

25.14 Develop a user-friendly computer program for the classical fourth-order RK method. Test the program by duplicating Example 25.7.

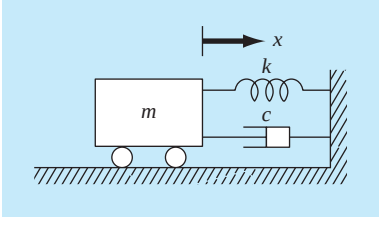
25.15 Develop a user-friendly computer program for systems of equations using the fourth-order RK method. Use this program to duplicate the computation in Example 25.10.

25.16 The motion of a damped spring-mass system (Fig. P25.16) is described by the following ordinary differential equation:

$$m\frac{d^2x}{dt^2} + c\frac{dx}{dt} + kx = 0$$

where x = displacement from equilibrium position (m), t = time (s), m = 20-kg mass, and c = the damping coefficient (N · s/m). The damping coefficient c takes on three values of 5 (underdamped), 40 (critically damped), and 200 (overdamped). The spring constant $k = 20$ N/m. The initial velocity is zero, and the initial displacement $x = 1$ m. Solve this equation using a numerical method over the time period $0 \le t \le 15$ s. Plot the displacement versus time for each of the three values of the damping coefficient on the same curve.

Figure P25.16



25.17 If water is drained from a vertical cylindrical tank by opening a valve at the base, the water will flow fast when the tank is full and slow down as it continues to drain. As it turns out, the rate at which the water level drops is:

$$\frac{dy}{dt} = -k\sqrt{y}$$

where k is a constant depending on the shape of the hole and the cross-sectional area of the tank and drain hole. The depth of the water y is measured in meters and the time t in minutes. If $k = 0.06$, determine how long it takes the tank to drain if the fluid level is initially 3 m. Solve by applying Euler's equation and writing a computer program or using Excel. Use a step of 0.5 minutes.

25.18 The following is an initial value, second-order differential equation:

$$\frac{d^2x}{dt^2} + (5x)\frac{dx}{dt} + (x + 7)\sin(\omega t) = 0$$

where

$$\frac{dx}{dt}(0) = 1.5 \quad \text{and} \quad x(0) = 6$$

Note that $\omega = 1$. Decompose the equation into two first-order differential equations. After the decomposition, solve the system from $t = 0$ to 15 and plot the results.

25.19 Assuming that drag is proportional to the square of velocity, we can model the velocity of a falling object like a parachutist with the following differential equation:

$$\frac{dv}{dt} = g - \frac{c_d}{m}v^2$$

where v is velocity (m/s), t = time (s), g is the acceleration due to gravity (9.81 m/s^2), c_d = a second-order drag coefficient (kg/m), and m = mass (kg). Solve for the velocity and distance fallen by a 90-kg object with a drag coefficient of 0.225 kg/m. If the initial height is 1 km, determine when it hits the ground. Obtain your solution with **(a)** Euler's method and **(b)** the fourth-order RK method.

25.20 A spherical tank has a circular orifice in its bottom through which the liquid flows out (Fig. P25.20). The flow rate through the hole can be estimated as

$$Q_{\text{out}} = CA\sqrt{2gH}$$

where Q_{out} = outflow (m^3/s), C = an empirically-derived coefficient, A = the area of the orifice (m^2), g = the gravitational constant (= 9.81 m/s^2), and H = the depth of liquid in the tank. Use one of the numerical methods described in this chapter to determine how long it will take for the water to flow out of a 3-m diameter tank with an initial height of 2.75 m. Note that the orifice has a diameter of 3 cm and $C = 0.55$.

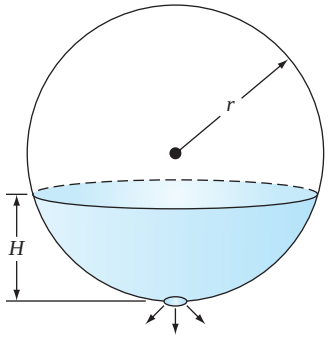


Figure P25.20
A spherical tank.

25.21 The logistic model is used to simulate population as in

$$\frac{dp}{dt} = k_{gm}(1 - p/p_{\max})p$$

where p = population, k_{gm} = the maximum growth rate under unlimited conditions, and $p_{\max}$ = the carrying capacity. Simulate the world's population from 1950 to 2000 using one of the numerical methods described in this chapter. Employ the following initial conditions and parameter values for your simulation: p_0 (in 1950) = 2555 million people, k_{gm} = 0.026/yr, and $p_{\max}$ = 12,000 million people. Have the function generate output corresponding to the dates for the following measured population data. Develop a plot of your simulation along with the data.

t	1950	1960	1970	1980	1990	2000
p	2555	3040	3708	4454	5276	6079

25.22 Suppose that a projectile is launched upward from the earth's surface. Assume that the only force acting on the object is the downward force of gravity. Under these conditions, a force balance can be used to derive,

$$\frac{dv}{dt} = -g(0)\frac{R^2}{(R + x)^2}$$

where v = upward velocity (m/s), t = time (s), x = altitude (m) measured upwards from the earth's surface, $g(0)$ = the gravitational acceleration at the earth's surface ($\cong$ 9.81 m/s^2), and R = the earth's radius ($\cong$ 6.37 $\times$ 10^6 m). Recognizing that $dx/dt = v$, use Euler's method to determine the maximum height that would be obtained if $v(t = 0) = 1400$ m/s.

25.23 The following function exhibits both flat and steep regions over a relatively short x region:

$$f(x) = \frac{1}{(x - 0.3)^2 + 0.01} + \frac{1}{(x - 0.9)^2 + 0.04} - 6$$

Determine the value of the definite integral of this function between $x = 0$ and 1 using an adaptive RK method.