2. The initial-value problem

$$y' = -y + 1 - \frac{y}{t}, \quad \text{for } 1 \le t \le 2, \quad \text{with } y(1) = 1$$

has the exact solution $y(t) = 1 + (e^{1-t} - 1)t^{-1}$.

 a. Use the Runge-Kutta-Fehlberg method with tolerance $TOL = 10^{-3}$ to find w_1 and w_2. Compare the approximate solutions to the actual values.

 b. Use the Adams Variable-Step-Size Predictor-Corrector method with tolerance $TOL = 0.002$ and starting values from the Runge-Kutta method of order 4 to find w_4 and w_5. Compare the approximate solutions to the actual values.

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

 a. $y' = \left(\frac{y}{t}\right)^2 + \frac{y}{t}$, for $1 \le t \le 1.2$, with $y(1) = 1$, $hmax = 0.05$, and $hmin = 0.02$.

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

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

 d. $y' = -ty + 4ty^{-1}$, for $0 \le t \le 1$, with $y(0) = 1$, $hmax = 0.2$, and $hmin = 0.01$.

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

 a. $y' = \frac{y}{t} - \frac{y^2}{t^2}$, for $1 \le t \le 4$, with $y(1) = 1$; actual solution $y(t) = t/(1 + \ln t)$.

 b. $y' = 1 + \frac{y}{t} + \left(\frac{y}{t}\right)^2$, for $1 \le t \le 3$, with $y(1) = 0$; actual solution $y(t) = t\tan(\ln t)$.

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

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

5. Use the Adams Variable-Step-Size Predictor-Corrector method with $TOL = 10^{-4}$ to approximate the solutions to the initial-value problems in Exercise 3.

6. Use the Adams Variable-Step-Size Predictor-Corrector method with tolerance $TOL = 10^{-6}$, $hmax = 0.5$, and $hmin = 0.02$ to approximate the solutions to the initial-value problems in Exercise 4.

7. 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 becomes

$$\left(1 + \frac{2k}{R_0}i\right)\frac{di}{dt} + \frac{1}{R_0C}i = \frac{1}{R_0C}\frac{d\mathcal{E}}{dt}.$$

Find the current i after 2 s, assuming $i(0) = 0$.

5.7 Methods for Systems of Equations

The most common application of numerical methods for approximating the solution of initial-value problems concerns not a single problem, but a linked system of differential equations. Why, then, have we spent the majority of this chapter considering the solution of a single equation? The answer is simple: to approximate the solution of a system of initial-value problems, we successively apply the techniques that we used to solve problems involving a single equation. As is so often the case in mathematics, the key to the methods for systems can be found by examining the easier problem and then logically modifying it to treat the more complicated situation.

An m**th-order system** of first-order initial-value problems has the form

$$\frac{du_1}{dt} = f_1(t, u_1, u_2, \ldots, u_m),$$
$$\frac{du_2}{dt} = f_2(t, u_1, u_2, \ldots, u_m),$$
$$\vdots$$
$$\frac{du_m}{dt} = f_m(t, u_1, u_2, \ldots, u_m),$$

for $a \le t \le b$, with the initial conditions

$$u_1(a) = \alpha_1, \quad u_2(a) = \alpha_2, \ \ldots, \ u_m(a) = \alpha_m.$$

The object is to find m functions $u_1, u_2, \ldots, u_m$ that satisfy the system of differential equations together with all the initial conditions.

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 4 given by

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

and

$$w_{i+1} = w_i + \frac{1}{6}[k_1 + 2k_2 + 2k_3 + k_4],$$

for each $i = 0, 1, \ldots, N-1$, is used to solve the first-order initial-value problem

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

It is generalized for systems as follows.

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

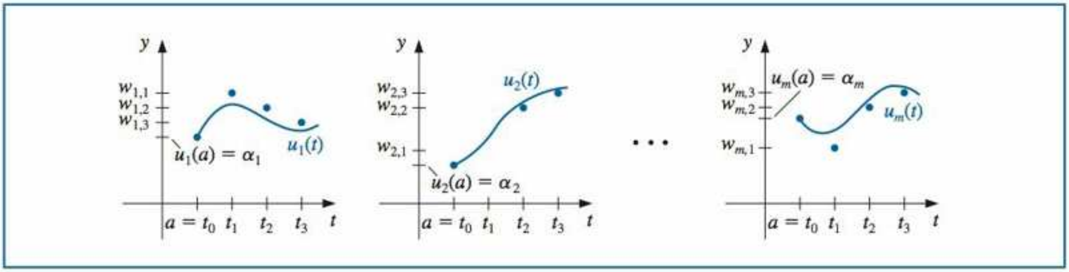
$$t_j = a + jh \quad \text{for each } j = 0, 1, \ldots, N.$$

Use the notation w_{ij} for each $j = 0, 1, \ldots, N$ and $i = 1, 2, \ldots, m$ to denote an approximation to $u_i(t_j)$; that is, w_{ij} approximates the ith solution $u_i(t)$ of the system at the jth mesh point t_j. For the initial conditions, set

$$w_{1,0} = \alpha_1, \quad w_{2,0} = \alpha_2, \quad \ldots, \quad w_{m,0} = \alpha_m.$$

Figure 5.4 gives an illustration of this notation.

Figure 5.4



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, for each $i = 1, 2, \ldots, m$,

$$k_{1,i} = hf_i(t_j, w_{1,j}, w_{2,j}, \ldots, w_{m,j}),$$

and then finding, for each i,

$$k_{2,i} = hf_i\left(t_j + \frac{h}{2}, w_{1,j} + \frac{1}{2}k_{1,1}, w_{2,j} + \frac{1}{2}k_{1,2}, \ldots, w_{m,j} + \frac{1}{2}k_{1,m}\right).$$

We next determine all the terms

$$k_{3,i} = hf_i\left(t_j + \frac{h}{2}, w_{1,j} + \frac{1}{2}k_{2,1}, w_{2,j} + \frac{1}{2}k_{2,2}, \ldots, w_{m,j} + \frac{1}{2}k_{2,m}\right)$$

and, finally, calculate all the terms

$$k_{4,i} = hf_i(t_j + h, w_{1,j} + k_{3,1}, w_{2,j} + k_{3,2}, \ldots, w_{m,j} + k_{3,m}).$$

Combining these values gives

$$w_{i,j+1} = w_{i,j} + \frac{1}{6}[k_{1,i} + 2k_{2,i} + 2k_{3,i} + k_{4,i}]$$

The program RKO4SY57 implements the Runge-Kutta method of order 4 for systems.

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 $k_{l+1,i}$.

Example 1 Kirchhoff's Law states that the sum of all instantaneous voltage changes around a closed electrical circuit is zero. This 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 henrys, and a voltage source of $E(t)$ volts must satisfy the equation

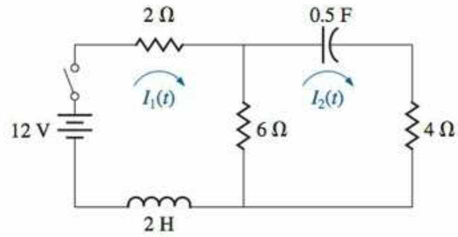
$$LI'(t) + RI(t) + \frac{1}{C}\int I(t)dt = E(t).$$

The currents $I_1(t)$ and $I_2(t)$ in the left and right loops, respectively, of the circuit shown in Figure 5.5 are the solutions to the system of equations

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

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

Figure 5.5



Suppose that the switch in the circuit is closed at time $t = 0$. This implies that $I_1(0)$ 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

$$I_1' = f_1(t, I_1, I_2) = -4I_1 + 3I_2 + 6, \quad \text{with } I_1(0) = 0,$$
$$I_2' = f_2(t, I_1, I_2) = 0.6I_1' - 0.2I_2 = -2.4I_1 + 1.6I_2 + 3.6, \quad \text{with } I_2(0) = 0.$$

The exact solution to this system is

$$I_1(t) = -3.375e^{-2t} + 1.875e^{-0.4t} + 1.5 \quad \text{and} \quad I_2(t) = -2.25e^{-2t} + 2.25e^{-0.4t}.$$

We will apply the Runge-Kutta method of order 4 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

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

and

$$k_{4,2} = (0.1)f_2(0.1, 0.54072, 0.321264) = 0.28162944.$$

As a consequence,

$$I_1(0.1) \approx w_{1,1} = w_{1,0} + \frac{1}{6}[k_{1,1} + 2k_{2,1} + 2k_{3,1} + k_{4,1}]$$
$$= 0 + \frac{1}{6}[0.6 + 2(0.534) + 2(0.54072) + 0.4800912] = 0.5382552$$

and

$$I_2(0.1) \approx w_{2,1} = w_{2,0} + \frac{1}{6}[k_{1,2} + 2k_{2,2} + 2k_{3,2} + k_{4,2}] = 0.3196263.$$

The remaining entries in Table 5.15 are generated in a similar manner. ∎

Table 5.15

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}

Any of the methods implemented in MATLAB can be used for systems of differential equations. For example, to use `ode45` to solve our system given in Example 1 we first define the right-hand sides using an `M-file` called `F.m` that contains the statements

```
function dy = F(t,y)
dy = zeros(2,1);
dy(1) = -4*y(1)+3*y(2)+6;
dy(2) = -2.4*y(1)+1.6*y(2)+3.6;
```

Then make the right-hand side of the system of differential equations known to MATLAB with

```
FF = @F
```

We now define the t values at which we want to approximate the solutions

```
tspan = [0 0.1 0.2 0.3 0.4 0.5]
```

The following command computes the solution to the system at the given values of t. The initial conditions $I_1(0) = 0$ and $I_2(0) = 0$ are given as [0 0].

```
[T,YY]=ode45(FF,tspan,[0 0])
```

The MATLAB response places the t values in the array `T` and the approximate solution values in `YY`, with the approximations for $I_1(t)$ in the first column and $I_2(t)$ in the second.

$$T = \begin{bmatrix} 0 \\ 0.100000000000000 \\ 0.200000000000000 \\ 0.300000000000000 \\ 0.400000000000000 \\ 0.500000000000000 \end{bmatrix} \quad \text{and} \quad YY = \begin{bmatrix} 0 & 0 \\ 0.538263922676270 & 0.319632054268176 \\ 0.968513005638230 & 0.568791683477228 \\ 1.310736555252393 & 0.760744806883952 \\ 1.581284356153020 & 0.906333359733513 \\ 1.793527044389029 & 1.014415449337470 \end{bmatrix}$$

Higher-Order Differential Equations

Many important physical problems—for example, electrical circuits and vibrating systems—involve initial-value problems whose equations have order higher than 1. 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 ***m*th-order initial-value problem** has the form

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

for $a \le t \le b$, with initial conditions

$$y(a) = \alpha_1, y'(a) = \alpha_2, \ldots, y^{(m-1)}(a) = \alpha_m.$$

To convert this into a system of first-order differential equations, define

$$u_1(t) = y(t), u_2(t) = y'(t), \ldots, u_m(t) = y^{(m-1)}(t).$$

Using this notation, we obtain the first-order system

$$\begin{aligned}
\frac{du_1}{dt} &= \frac{dy}{dt} = u_2,\\
\frac{du_2}{dt} &= \frac{dy'}{dt} = u_3,\\
&\vdots\\
\frac{du_{m-1}}{dt} &= \frac{dy^{(m-2)}}{dt} = u_m,
\end{aligned}$$

and

$$\frac{du_m}{dt} = \frac{dy^{(m-1)}}{dt} = y^{(m)} = f\big(t, y, y', \ldots, y^{(m-1)}\big) = f(t, u_1, u_2, \ldots, u_m),$$

with initial conditions

$$u_1(a) = y(a) = \alpha_1,\ u_2(a) = y'(a) = \alpha_2, \ldots, u_m(a) = y^{(m-1)}(a) = \alpha_m.$$

Example 2 Transform 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 of order 4 with $h = 0.1$ to approximate the solution.

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

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

with initial conditions $u_1(0) = -0.4,\ u_2(0) = -0.6$.

The initial conditions give $w_{1,0} = -0.4$ and $w_{2,0} = -0.6$. The Runge-Kutta method of order 4 for systems described on page 216 with $j = 0$ give

$$k_{1,1} = hf_1(t_0, w_{1,0}, w_{2,0}) = hw_{2,0} = -0.06,$$

$$k_{1,2} = hf_2(t_0, w_{1,0}, w_{2,0}) = h[e^{2t_0}\sin t_0 - 2w_{1,0} + 2w_{2,0}] = -0.04,$$

$$k_{2,1} = hf_1\left(t_0 + \frac{h}{2}, w_{1,0} + \frac{1}{2}k_{1,1}, w_{2,0} + \frac{1}{2}k_{1,2}\right) = h\left[w_{2,0} + \frac{1}{2}k_{1,2}\right] = -0.062,$$

$$k_{2,2} = hf_2\left(t_0 + \frac{h}{2}, w_{1,0} + \frac{1}{2}k_{1,1}, w_{2,0} + \frac{1}{2}k_{1,2}\right)$$

$$= h\left[e^{2(t_0+0.05)}\sin(t_0 + 0.05) - 2\left(w_{1,0} + \frac{1}{2}k_{1,1}\right) + 2\left(w_{2,0} + \frac{1}{2}k_{1,2}\right)\right]$$

$$= -0.03247644757,$$

$$k_{3,1} = h\left[w_{2,0} + \frac{1}{2}k_{2,2}\right] = -0.06162832238,$$

$$k_{3,2} = h\left[e^{2(t_0+0.05)}\sin(t_0 + 0.05) - 2\left(w_{1,0} + \frac{1}{2}k_{2,1}\right) + 2\left(w_{2,0} + \frac{1}{2}k_{2,2}\right)\right]$$

$$= -0.03152409237,$$

$$k_{4,1} = h\left[w_{2,0} + k_{3,2}\right] = -0.06315240924,$$

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

$$w_{1,1} = w_{1,0} + \frac{1}{6}(k_{1,1} + 2k_{2,1} + 2k_{3,1} + k_{4,1}) = -0.4617333423 \quad \text{and}$$

$$w_{2,1} = w_{2,0} + \frac{1}{6}(k_{1,2} + 2k_{2,2} + 2k_{3,2} + k_{4,2}) = -0.6316312421.$$

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 5.16 and are compared to the actual values of $u_1(t) = 0.2e^{2t}(\sin t - 2\cos t)$ and $u_2(t) = u_1'(t) = 0.2e^{2t}(4\sin t - 3\cos t)$. ■

Table 5.16

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	−0.60000000	−0.60000000	0	0
0.1	−0.46173297	−0.46173334	−0.63163105	−0.63163124	3.72×10^{-7}	1.92×10^{-7}
0.2	−0.52555905	−0.52555988	−0.64014866	−0.64014895	8.36×10^{-7}	2.84×10^{-7}
0.3	−0.58860005	−0.58860144	−0.61366361	−0.61366381	1.39×10^{-6}	1.99×10^{-7}
0.4	−0.64661028	−0.64661231	−0.53658220	−0.53658203	2.02×10^{-6}	1.68×10^{-7}
0.5	−0.69356395	−0.69356666	−0.38873906	−0.38873810	2.71×10^{-6}	9.58×10^{-7}
0.6	−0.72114849	−0.72115190	−0.14438322	−0.14438087	3.41×10^{-6}	2.35×10^{-6}
0.7	−0.71814890	−0.71815295	0.22899243	0.22899702	4.06×10^{-6}	4.59×10^{-6}
0.8	−0.66970677	−0.66971133	0.77198383	0.77199180	4.55×10^{-6}	7.97×10^{-6}
0.9	−0.55643814	−0.55644290	1.5347686	1.5347815	4.77×10^{-6}	1.29×10^{-5}
1.0	−0.35339436	−0.35339886	2.5787466	2.5787663	4.50×10^{-6}	1.97×10^{-5}

Other one-step approximation methods can be extended to systems. If the Runge-Kutta-Fehlberg method is extended, then 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 must be recomputed.

The multistep methods and predictor-corrector techniques can also be extended easily 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.

EXERCISE SET 5.7

1. Use the Runge-Kutta method of order 4 for systems to approximate the solutions of the following systems of first-order differential equations and compare the results to the actual solutions.

   a. $u_1' = 3u_1 + 2u_2 - (2t^2+1)e^{2t}$, for $0 \le t \le 1$ with $u_1(0) = 1$;
   $u_2' = 4u_1 + u_2 + (t^2+2t-4)e^{2t}$, for $0 \le t \le 1$ with $u_2(0) = 1$;
   $h = 0.2$; 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. $u_1' = -4u_1 - 2u_2 + \cos t + 4\sin t$, for $0 \le t \le 2$ with $u_1(0) = 0$;
   $u_2' = 3u_1 + u_2 - 3\sin t$, for $0 \le t \le 2$ with $u_2(0) = -1$;
   $h = 0.1$; actual solutions $u_1(t) = 2e^{-t} - 2e^{-2t} + \sin t$ and $u_2(t) = -3e^{-t} + 2e^{-2t}$.

   c. $u_1' = u_2$, for $0 \le t \le 2$ with $u_1(0) = 1$;
   $u_2' = -u_1 - 2e^t + 1$, for $0 \le t \le 2$ with $u_2(0) = 0$;
   $u_3' = -u_1 - e^t + 1$, for $0 \le t \le 2$ with $u_3(0) = 1$;
   $h = 0.5$; 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. $u_1' = u_2 - u_3 + t$, for $0 \le t \le 1$ with $u_1(0) = 1$;
   $u_2' = 3t^2$, for $0 \le t \le 1$ with $u_2(0) = 1$;
   $u_3' = u_2 + e^{-t}$, for $0 \le t \le 1$ with $u_3(0) = -1$;
   $h = 0.1$; 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 higher-order differential equations and compare the results to the actual solutions.

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

   b. $t^2y'' - 2ty' + 2y = t^3 \ln t$, for $1 \le t \le 2$ with $y(1) = 1$, $y'(1) = 0$, and $h = 0.1$; actual solution $y(t) = \frac{7}{4}t + \frac{1}{2}t^3 \ln t - \frac{3}{4}t^3$.

   c. $y''' + 2y'' - y' - 2y = e^t$, for $0 \le t \le 3$ with $y(0) = 1$, $y'(0) = 2$, $y''(0) = 0$, and $h = 0.2$; 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. $t^3y''' - t^2y'' + 3ty' - 4y = 5t^3 \ln t + 9t^3$, for $1 \le t \le 2$ with $y(1) = 0$, $y'(1) = 1$, $y''(1) = 3$, and $h = 0.1$; actual solution $y(t) = -t^2 + t\cos(\ln t) + t\sin(\ln t) + t^3 \ln t$.

3. Change the Adams Fourth-Order Predictor-Corrector method to obtain approximate solutions to systems of first-order equations.

4. Repeat Exercise 1 using the method developed in Exercise 3.

5. 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 this century by A. J. Lotka and V. Volterra. 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

$$x_1'(t) = k_1x_1(t) - k_2x_1(t)x_2(t) \quad \text{and} \quad x_2'(t) = k_3x_1(t)x_2(t) - k_4x_2(t).$$

Use Runge-Kutta of order 4 for systems to 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 $k_4 = 0.5$. Is there a stable solution to this population model? If so, for what values x_1 and x_2 is the solution stable?

6. In Exercise 5 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

$$\frac{dx_1(t)}{dt} = x_1(t)[4 - 0.0003x_1(t) - 0.0004x_2(t)]$$

and

$$\frac{dx_2(t)}{dt} = x_2(t)[2 - 0.0002x_1(t) - 0.0001x_2(t)].$$

If it is known that the initial population of each species is 10,000, find the solution to this system for $0 \le t \le 4$. Is there a stable solution to this population model? If so, for what values of x_1 and x_2 is the solution stable?

5.8 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 step sizes increase, the error can be kept in relative control, provided that the solution also grows in magnitude. Problems frequently arise, however, where 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 have an exact solution with 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. A 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, and for large values of c it can grow very large. In addition, the derivative in the error term is evaluated not at t, but at a number between zero and t, so the derivative terms may increase as t increases—and very rapidly indeed. Fortunately, stiff equations can generally 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.