and the derivative is $u'(t) = Y'(t) - Z'(t) = f(t, Y(t)) - f(t, Z(t))$. The Lipschitz condition implies that

$$u' = |f(t,Y) - f(t,Z)| \le L|Y(t) - Z(t)| = L|u(t)| = Lu(t),$$

and therefore $(\ln u)' = u'/u \le L$. By the Mean Value Theorem,

$$\frac{\ln u(t) - \ln u(a)}{t - a} \le L,$$

which simplifies to

$$\ln \frac{u(t)}{u(a)} \le L(t-a)$$
$$u(t) \le u(a)e^{L(t-a)}.$$

This is the desired result. □

Returning to Example 6.4, Theorem 6.3 implies that solutions $Y(t)$ and $Z(t)$, starting at different initial values, must not grow apart any faster than a multiplicative factor of e^t for $0 \le t \le 1$. In fact, the solution at initial value Y_0 is $Y(t) = (2 + Y_0)e^{t^2/2} - t^2 - 2$, and so the difference between two solutions is

$$\begin{aligned} |Y(t) - Z(t)| &\le |(2+Y_0)e^{t^2/2} - t^2 - 2 - ((2+Z_0)e^{t^2/2} - t^2 - 2)| \\ &\le |Y_0 - Z_0|e^{t^2/2}, \end{aligned} \tag{6.14}$$

which is less than $|Y_0 - Z_0|e^t$ for $0 \le t \le 1$, as prescribed by Theorem 6.3.

6.1.3 First-order linear equations

A special class of ordinary differential equations that can be readily solved provides a handy set of illustrative examples. They are the first-order equations whose right-hand sides are linear in the y variable. Consider the initial value problem

$$\begin{cases} y' = g(t)y + h(t) \\ y(a) = y_a \\ t \text{ in } [a,b] \end{cases}. \tag{6.15}$$

First note that if $g(t)$ is continuous on $[a,b]$, a unique solution exists by Theorem 6.2, using $L = \max_{[a,b]} g(t)$ as the Lipschitz constant. The solution is found by a trick, multiplying the equation through by an "integrating factor."

The integrating factor is $e^{-\int g(t)\,dt}$. Multiplying both sides by it yields

$$\begin{aligned} (y' - g(t)y)e^{-\int g(t)\,dt} &= e^{-\int g(t)\,dt}h(t) \\ \left(ye^{-\int g(t)\,dt}\right)' &= e^{-\int g(t)\,dt}h(t) \\ ye^{-\int g(t)\,dt} &= \int e^{-\int g(t)\,dt}h(t)\,dt, \end{aligned}$$

which can be solved as

$$y(t) = e^{\int g(t)\,dt} \int e^{-\int g(t)\,dt}h(t)\,dt. \tag{6.16}$$

If the integrating factor can be expressed simply, this method allows an explicit solution of the first-order linear equation (6.15).

▶ EXAMPLE 6.6 Solve the first-order linear differential equation

$$\begin{cases} y' = ty + t^3 \\ y(0) = y_0 \end{cases}. \tag{6.17}$$

The integrating factor is

$$e^{-\int g(t)\,dt} = e^{-\frac{t^2}{2}}.$$

According to (6.16), the solution is

$$\begin{aligned} y(t) &= e^{\frac{t^2}{2}} \int e^{-\frac{t^2}{2}} t^3\,dt \\ &= e^{\frac{t^2}{2}} \int e^{-u}(2u)\,du \\ &= 2e^{\frac{t^2}{2}} \left[-\frac{t^2}{2}e^{-\frac{t^2}{2}} - e^{-\frac{t^2}{2}} + C\right] \\ &= -t^2 - 2 + 2Ce^{\frac{t^2}{2}}, \end{aligned}$$

where the substitution $u = t^2/2$ was made. Solving for the integration constant C yields $y_0 = -2 + 2C$, so $C = (2 + y_0)/2$. Therefore,

$$y(t) = (2 + y_0)e^{\frac{t^2}{2}} - t^2 - 2.$$ ◀

6.1 Exercises

1. Show that the function $y(t) = t\sin t$ is a solution of the differential equations (a) $y + t^2\cos t = ty'$ (b) $y'' = 2\cos t - y$ (c) $t(y'' + y) = 2y' - 2\sin t$.

2. Show that the function $y(t) = e^{\sin t}$ is a solution of the initial value problems (a) $y' = y\cos t, y(0) = 1$ (b) $y'' = (\cos t)y' - (\sin t)y, y(0) = 1, y'(0) = 1$ (c) $y'' = y(1 - \ln y - (\ln y)^2), y(\pi) = 1, y'(\pi) = -1$.

3. Use separation of variables to find solutions of the IVP given by $y(0) = 1$ and the following differential equations:

   (a) $y' = t$ (b) $y' = t^2 y$ (c) $y' = 2(t+1)y$

   (d) $y' = 5t^4 y$ (e) $y' = 1/y^2$ (f) $y' = t^3/y^2$

4. Find the solutions of the IVP given by $y(0) = 0$ and the following first-order linear differential equations:

   (a) $y' = t + y$ (b) $y' = t - y$ (c) $y' = 4t - 2y$

5. Apply Euler's Method with step size $h = 1/4$ to the IVPs in Exercise 3 on the interval $[0, 1]$. List the $w_i, i = 0, \ldots, 4$, and find the error at $t = 1$ by comparing with the correct solution.

6. Apply Euler's Method with step size $h = 1/4$ to the IVPs in Exercise 4 on the interval $[0, 1]$. Find the error at $t = 1$ by comparing with the correct solution.

7. (a) Show that $y = \tan(t + c)$ is a solution of the differential equation $y' = 1 + y^2$ for each c. (b) For each real number y_0, find c in the interval $(-\pi/2, \pi/2)$ such that the initial value problem $y' = 1 + y^2, y(0) = y_0$ has a solution $y = \tan(t + c)$.

8. (a) Show that $y = \tanh(t + c)$ is a solution of the differential equation $y' = 1 - y^2$ for each c. (b) For each real number y_0 in the interval $(-1, 1)$, find c such that the initial value problem $y' = 1 - y^2, y(0) = y_0$ has a solution $y = \tanh(t + c)$.

9. For which of these initial value problems on $[0, 1]$ does Theorem 6.2 guarantee a unique solution? Find the Lipschitz constants if they exist (a) $y' = t$ (b) $y' = y$ (c) $y' = -y$ (d) $y' = -y^3$.

10. Sketch the slope field of the differential equations in Exercise 9, and draw rough approximations to the solutions, starting at the initial conditions $y(0) = 1$, $y(0) = 0$, and $y(0) = -1$.

11. Find the solutions of the initial value problems in Exercise 9. For each equation, use the Lipschitz constants from Exercise 9, and verify, if possible, the inequality of Theorem 6.3 for the pair of solutions with initial conditions $y(0) = 0$ and $y(0) = 1$.

12. (a) Show that if $a \neq 0$, the solution of the initial value problem $y' = ay + b, y(0) = y_0$ is $y(t) = (b/a)(e^{at} - 1) + y_0e^{at}$. (b) Verify the inequality of Theorem 6.3 for solutions $y(t), z(t)$ with initial values y_0 and z_0, respectively.

13. Use separation of variables to solve the initial value problem $y' = y^2, y(0) = 1$.

14. Find the solution of the initial value problem $y' = ty^2$ with $y(0) = 1$. What is the largest interval $[0, b]$ for which the solution exists?

15. Consider the initial value problem $y' = \sin y,\ y(a) = y_a$ on $a \le t \le b$.
(a) On what subinterval of $[a, b]$ does Theorem 6.2 guarantee a unique solution?
(b) Show that $y(t) = 2\arctan(e^{t-a}\tan(y_a/2)) + 2\pi[(y_a + \pi)/2\pi]$ is the solution of the initial value problem, where [ ] denotes the greatest integer function.

16. Consider the initial value problem $y' = \sinh y,\ y(a) = y_a$ on $a \le t \le b$.
(a) On what subinterval of $[a, b]$ does Theorem 6.2 guarantee a unique solution?
(b) Show that $y(t) = 2\,\mathrm{arctanh}(e^{t-a}\tanh(y_a/2))$ is a solution of the initial value problem.
(c) On what interval $[a, c)$ does the solution exist?

6.1 Computer Problems

1. Apply Euler's Method with step size $h = 0.1$ on $[0, 1]$ to the initial value problems in Exercise 3. Print a table of the t values, Euler approximations, and error (difference from exact solution) at each step.

2. Plot the Euler's Method approximate solutions for the IVPs in Exercise 3 on $[0, 1]$ for step sizes $h = 0.1, 0.05$, and 0.025, along with the exact solution.

3. Plot the Euler's Method approximate solutions for the IVPs in Exercise 4 on $[0, 1]$ for step sizes $h = 0.1, 0.05$, and 0.025, along with the exact solution.

4. For the IVPs in Exercise 3, make a log–log plot of the error of Euler's Method at $t = 1$ as a function of $h = 0.1 \times 2^{-k}$ for $0 \le k \le 5$. Use the MATLAB `loglog` command as in Figure 6.4.

5. For the IVPs in Exercise 4, make a log–log plot of the error of Euler's Method at $t = 1$ as a function of $h = 0.1 \times 2^{-k}$ for $0 \le k \le 5$.

6. For the initial value problems in Exercise 4, make a log–log plot of the error of Euler's Method at $t = 2$ as a function of $h = 0.1 \times 2^{-k}$ for $0 \le k \le 5$.

7. Plot the Euler's Method approximate solution on $[0, 1]$ for the differential equation $y' = 1 + y^2$ and initial condition (a) $y_0 = 0$ (b) $y_0 = 1$, along with the exact solution (see Exercise 7). Use step sizes $h = 0.1$ and 0.05.

8. Plot the Euler's Method approximate solution on $[0, 1]$ for the differential equation $y' = 1 - y^2$ and initial condition (a) $y_0 = 0$ (b) $y_0 = -1/2$, along with the exact solution (see Exercise 8). Use step sizes $h = 0.1$ and 0.05.

9. Calculate the Euler's Method approximate solution on $[0, 4]$ for the differential equation $y' = \sin y$ and initial condition (a) $y_0 = 0$ (b) $y_0 = 100$, using step sizes $h = 0.1 \times 2^{-k}$ for $0 \le k \le 5$. Plot the $k = 0$ and $k = 5$ approximate solutions along with the exact solution (see Exercise 15), and make a log–log plot of the error at $t = 4$ as a function of h.

10. Calculate the Euler's Method approximate solution of the differential equation $y' = \sinh y$ and initial condition (a) $y_0 = 1/4$ on the interval $[0, 2]$ (b) $y_0 = 2$ on the interval $[0, 1/4]$, using step sizes $h = 0.1 \times 2^{-k}$ for $0 \le k \le 5$. Plot the $k = 0$ and $k = 5$ approximate solutions along with the exact solution (see Exercise 16), and make a log–log plot of the error at the end of the time interval as a function of h.

6.2 ANALYSIS OF IVP SOLVERS

Figure 6.4 shows consistently decreasing error in the Euler's Method approximation as a function of decreasing step size for Example 6.1. Is this generally true? Can we make the error as small as we want, just by decreasing the step size? A careful investigation of error in Euler's Method will illustrate the issues for IVP solvers in general.

6.2.1 Local and global truncation error

Figure 6.5 shows a schematic picture for one step of a solver like Euler's Method when solving an initial value problem of the form

$$\begin{cases} y' = f(t, y) \\ y(a) = y_a \\ t \text{ in } [a, b] \end{cases}. \tag{6.18}$$

At step i, the accumulated error from the previous steps is carried along and perhaps amplified, while new error from the Euler approximation is added. To be precise, let us define the **global truncation error**

$$g_i = |w_i - y_i|$$

to be the difference between the ODE solver (Euler's Method, for example) approximation and the correct solution of the initial value problem. Also, we will define the **local truncation error**, or one-step error, to be

$$e_{i+1} = |w_{i+1} - z(t_{i+1})|, \tag{6.19}$$

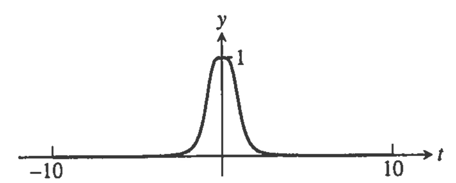

Figure 6.8 Approximation of Example 6.9 by the Trapezoid Method. Step size is $h = 10^{-3}$. Note the significant improvement in accuracy compared with Euler's Method in Figure 6.6.

Subtracting (6.32) from (6.31) gives the local truncation error as

$$y_{i+1} - w_{i+1} = O(h^3).$$

Theorem 6.4 shows that the global error of the Trapezoid Method is proportional to h^2, meaning that the method is of order two, compared with order one for Euler's Method. For small h this is a significant difference, as shown by returning to Example 6.9.

▶ EXAMPLE 6.11 Apply the Trapezoid Method to Example 6.9:

$$\begin{cases} y' = -4t^3y^2 \\ y(-10) = 1/10001. \\ t \text{ in } [-10, 0] \end{cases}$$

Revisiting Example 6.9 with a more powerful method yields a great improvement in approximating the solution, for example, at $x = 0$. The correct value $y(0) = 1$ is attained within .0015 with a step size of $h = 10^{-3}$ with the Trapezoid Method, as shown in Figure 6.8. This is already better than Euler with a step size of $h = 10^{-5}$. Using the Trapezoid Method with $h = 10^{-5}$ yields an error on the order of 10^{-7} for this relatively difficult initial value problem. ◀

6.2.3 Taylor Methods

So far, we have learned two methods for approximating solutions of ordinary differential equations. The Euler Method has order one, and the apparently superior Trapezoid Method has order two. In this section, we show that methods of all orders exist. For each positive integer k, there is a Taylor Method of order k, which we will describe next.

The basic idea is a straightforward exploitation of the Taylor expansion. Assume that the solution $y(t)$ is $(k+1)$ times continuously differentiable. Given the current point $(t, y(t))$ on the solution curve, the goal is to express $y(t+h)$ in terms of $y(t)$ for some step size h, using information about the differential equation. The Taylor expansion of $y(t)$ about t is

$$y(t+h) = y(t) + hy'(t) + \frac{1}{2}h^2y''(t) + \cdots + \frac{1}{k!}h^ky^{(k)}(t)$$
$$+ \frac{1}{(k+1)!}h^{k+1}y^{(k+1)}(c), \qquad (6.33)$$

where c lies between t and $t+h$. The last term is the Taylor remainder term. This equation motivates the following method:

Taylor Method of order k

$$w_0 = y_0$$
$$w_{i+1} = w_i + hf(t_i, w_i) + \frac{h^2}{2}f'(t_i, w_i) + \cdots + \frac{h^k}{k!}f^{(k-1)}(t_i, w_i). \qquad (6.34)$$

The prime notation refers to the total derivative of $f(t, y(t))$ with respect to t. For example,

$$f'(t, y) = f_t(t, y) + f_y(t, y)y'(t)$$
$$= f_t(t, y) + f_y(t, y)f(t, y).$$

We use the notation f_t to denote the partial derivative of f with respect to t, and similarly for f_y. To find the local truncation error of the Taylor Method, set $w_i = y_i$ in (6.34) and compare with the Taylor expansion (6.33) to get

$$y_{i+1} - w_{i+1} = \frac{h^{k+1}}{(k+1)!}y^{(k+1)}(c).$$

We conclude that the Taylor Method of order k has local truncation error h^{k+1} and has order k, according to Theorem 6.4.

The first-order Taylor Method is

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

which is identified as Euler's Method. The second-order Taylor Method is

$$w_{i+1} = w_i + hf(t_i, w_i) + \frac{1}{2}h^2(f_t(t_i, w_i) + f_y(t_i, w_i)f(t_i, w_i)).$$

▶ EXAMPLE 6.12 Determine the second-order Taylor Method for the first-order linear equation

$$\begin{cases} y' = ty + t^3 \\ y(0) = y_0 \end{cases} \qquad (6.35)$$

Since $f(t, y) = ty + t^3$, it follows that

$$f'(t, y) = f_t + f_yf$$
$$= y + 3t^2 + t(ty + t^3),$$

and the method gives

$$w_{i+1} = w_i + h(t_iw_i + t_i^3) + \frac{1}{2}h^2(w_i + 3t_i^2 + t_i(t_iw_i + t_i^3)).$$ ◀

Although second-order Taylor Method is a second-order method, notice that manual labor on the user's part was required to determine the partial derivatives. Compare this with the other second-order method we have learned, where (6.29) requires only calls to a routine that computes values of $f(t, y)$ itself.

Conceptually, the lesson represented by Taylor Methods is that ODE methods of arbitrary order exist, as shown in (6.34). However, they suffer from the problem that extra work is needed to compute the partial derivatives of f that show up in the formula. Since formulas of the same orders can be developed that do not require these partial derivatives, the Taylor Methods are used only for specialized purposes.

6.2 Exercises

1. Using initial condition $y(0) = 1$ and step size $h = 1/4$, calculate the Trapezoid Method approximation $w_0, \ldots, w_4$ on the interval $[0, 1]$. Find the error at $t = 1$ by comparing with the correct solution found in Exercise 6.1.3.

   (a) $y' = t$ (b) $y' = t^2y$ (c) $y' = 2(t+1)y$

(d) $y' = 5t^4 y$ (e) $y' = 1/y^2$ (f) $y' = t^3/y^2$

2. Using initial condition $y(0) = 0$ and step size $h = 1/4$, calculate the Trapezoid Method approximation on the interval $[0, 1]$. Find the error at $t = 1$ by comparing with the correct solution found in Exercise 6.1.4.

(a) $y' = t + y$ (b) $y' = t - y$ (c) $y' = 4t - 2y$

3. Find the formula for the second-order Taylor Method for the following differential equations: (a) $y' = ty$ (b) $y' = ty^2 + y^3$ (c) $y' = y \sin y$ (d) $y' = e^{yt^2}$
4. Apply the second-order Taylor Method to the initial value problems in Exercise 1. Using step size $h = 1/4$, calculate the second-order Taylor Method approximation on the interval $[0, 1]$. Compare with the correct solution found in Exercise 6.1.3, and find the error at $t = 1$.
5. (a) Prove (6.22) (b) Prove (6.23).

6.2 Computer Problems

1. Apply the Explicit Trapezoid Method on a grid of step size $h = 0.1$ in $[0, 1]$ to the initial value problems in Exercise 1. Print a table of the t values, approximations, and global truncation error at each step.
2. Plot the approximate solutions for the IVPs in Exercise 1 on $[0, 1]$ for step sizes $h = 0.1, 0.05$, and 0.025, along with the true solution.
3. For the IVPs in Exercise 1, plot the global truncation error of the explicit Trapezoid Method at $t = 1$ as a function of $h = 0.1 \times 2^{-k}$ for $0 \le k \le 5$. Use a log–log plot as in Figure 6.4.
4. For the IVPs in Exercise 1, plot the global truncation error of the second-order Taylor Method at $t = 1$ as a function of $h = 0.1 \times 2^{-k}$ for $0 \le k \le 5$.
5. Plot the Trapezoid Method approximate solution on $[0, 1]$ for the differential equation $y' = 1 + y^2$ and initial condition (a) $y_0 = 0$ (b) $y_0 = 1$, along with the exact solution (see Exercise 6.1.7). Use step sizes $h = 0.1$ and 0.05.
6. Plot the Trapezoid Method approximate solution on $[0, 1]$ for the differential equation $y' = 1 - y^2$ and initial condition (a) $y_0 = 0$ (b) $y_0 = -1/2$, along with the exact solution (see Exercise 6.1.8). Use step sizes $h = 0.1$ and 0.05.
7. Calculate the Trapezoid Method approximate solution on $[0, 4]$ for the differential equation $y' = \sin y$ and initial condition (a) $y_0 = 0$ (b) $y_0 = 100$, using step sizes $h = 0.1 \times 2^{-k}$ for $0 \le k \le 5$. Plot the $k = 0$ and $k = 5$ approximate solutions along with the exact solution (see Exercise 6.1.15), and make a log–log plot of the error at $t = 4$ as a function of h.
8. Calculate the Trapezoid Method approximate solution of the differential equation $y' = \sinh y$ and initial condition (a) $y_0 = 1/4$ on the interval $[0, 2]$ (b) $y_0 = 2$ on the interval $[0, 1/4]$, using step sizes $h = 0.1 \times 2^{-k}$ for $0 \le k \le 5$. Plot the $k = 0$ and $k = 5$ approximate solutions along with the exact solution (see Exercise 6.1.16), and make a log–log plot of the error at the end of the time interval as a function of h.

6.3 SYSTEMS OF ORDINARY DIFFERENTIAL EQUATIONS

Approximation of systems of differential equations can be done as a simple extension of the methodology for a single differential equation. Treating systems of equations greatly extends our ability to model interesting dynamical behavior.

The ability to solve systems of ordinary differential equations lies at the core of the art and science of computer simulation. In this section, we introduce two physical systems whose simulation has motivated a great deal of development of ODE solvers: the pendulum and orbital mechanics. The study of these examples will provide the reader some practical experience in the capabilities and limitations of the solvers.

The **order** of a differential equation refers to the highest order derivative appearing in the equation. A first-order system has the form

$$\begin{aligned} y_1' &= f_1(t, y_1, \ldots, y_n) \\ y_2' &= f_2(t, y_1, \ldots, y_n) \\ &\vdots \\ y_n' &= f_n(t, y_1, \ldots, y_n). \end{aligned}$$

In an initial value problem, each variable needs its own initial condition.

▶ EXAMPLE 6.13 Apply Euler's Method to the first-order system of two equations:

$$\begin{aligned} y_1' &= y_2^2 - 2y_1 \\ y_2' &= y_1 - y_2 - ty_2^2 \\ y_1(0) &= 0 \\ y_2(0) &= 1. \end{aligned} \tag{6.36}$$

Check that the solution of the system (6.36) is the vector-valued function

$$\begin{aligned} y_1(t) &= te^{-2t} \\ y_2(t) &= e^{-t}. \end{aligned}$$

For the moment, forget that we know the solution, and apply Euler's Method. The scalar Euler's Method formula is applied to each component in turn as follows:

$$\begin{aligned} w_{i+1,1} &= w_{i,1} + h(w_{i,2}^2 - 2w_{i,1}) \\ w_{i+1,2} &= w_{i,2} + h(w_{i,1} - w_{i,2} - t_i w_{i,2}^2). \end{aligned}$$

Figure 6.9 shows the Euler Method approximations of y_1 and y_2, along with the correct solution. The MATLAB code that carries this out is essentially the same as Program 6.1, with a few adjustments to treat y as a vector:

```
% Program 6.2  Vector version of Euler Method
% Input: interval inter, initial vector y0, number of steps n
% Output: time steps t, solution y
% Example usage: euler2([0 1],[0 1],10);
function [t,y]=euler2(inter,y0,n)
t(1)=inter(1); y(1,:)=y0;
h=(inter(2)-inter(1))/n;
for i=1:n
  t(i+1)=t(i)+h;
```

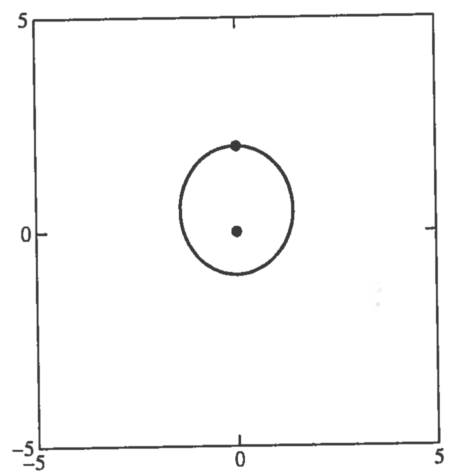

Figure 6.13 One-body problem approximated by the Trapezoid Method. Step size $h = 0.01$. The orbit appears to close, at least to the resolution visible in the plot.

lead to large deviations at a later time. In our terms, this is the statement that the solution of the system of differential equations is ill-conditioned with respect to the input of initial conditions.

The restricted three-body problem is a system of 12 equations, 4 for each body, that are also derived from Newton's second law. For example, the equations of the first body are

$$\begin{aligned} x_1' &= v_{1x} \\ v_{1x}' &= \frac{gm_2(x_2 - x_1)}{((x_2 - x_1)^2 + (y_2 - y_1)^2)^{3/2}} + \frac{gm_3(x_3 - x_1)}{((x_3 - x_1)^2 + (y_3 - y_1)^2)^{3/2}} \\ y_1' &= v_{1y} \\ v_{1y}' &= \frac{gm_2(y_2 - y_1)}{((x_2 - x_1)^2 + (y_2 - y_1)^2)^{3/2}} + \frac{gm_3(y_3 - y_1)}{((x_3 - x_1)^2 + (y_3 - y_1)^2)^{3/2}}. \end{aligned} \tag{6.45}$$

The second and third bodies, at (x_2, y_2) and (x_3, y_3), respectively, satisfy similar equations.

Computer Problems 9 and 10 ask the reader to computationally solve the two- and three-body problems. The latter problem illustrates severe sensitive dependence on initial conditions.

6.3 Exercises

1. Apply the Euler's Method with step size $h = 1/4$ to the initial value problem on $[0, 1]$.

(a) $\begin{cases} y_1' = y_1 + y_2 \\ y_2' = -y_1 + y_2 \\ y_1(0) = 1 \\ y_2(0) = 0 \end{cases}$ (b) $\begin{cases} y_1' = -y_1 - y_2 \\ y_2' = y_1 - y_2 \\ y_1(0) = 1 \\ y_2(0) = 0 \end{cases}$

(c) $\begin{cases} y_1' = -y_2 \\ y_2' = y_1 \\ y_1(0) = 1 \\ y_2(0) = 0 \end{cases}$ (d) $\begin{cases} y_1' = y_1 + 3y_2 \\ y_2' = 2y_1 + 2y_2 \\ y_1(0) = 5 \\ y_2(0) = 0 \end{cases}$

Find the global truncation errors of y_1 and y_2 at $t = 1$ by comparing with the correct solutions
(a) $y_1(t) = e^t \cos t, y_2(t) = -e^t \sin t$ (b) $y_1(t) = e^{-t} \cos t, y_2(t) = e^{-t} \sin t$
(c) $y_1(t) = \cos t, y_2(t) = \sin t$ (d) $y_1(t) = 3e^{-t} + 2e^{4t}, y_2(t) = -2e^{-t} + 2e^{4t}$.

2. Apply the Trapezoid Method with $h = 1/4$ to the initial value problems in Exercise 1. Find the global truncation error at $t = 1$ by comparing with the correct solutions.

3. Convert the higher-order ordinary differential equation to a first-order system of equations. (a) $y'' - ty = 0$ (Airy's equation) (b) $y'' - 2ty' + 2y = 0$ (Hermite's equation) (c) $y'' - ty' - y = 0$

4. Apply the Trapezoid Method with $h = 1/4$ to the initial value problems in Exercise 3, using $y(0) = y'(0) = 1$.

5. (a) Show that $y(t) = (e^t + e^{-t} - t^2)/2 - 1$ is the solution of the initial value problem $y''' - y' = t$, with $y(0) = y'(0) = y''(0) = 0$. (b) Convert the differential equation to a system of three first-order equations. (c) Use Euler's Method with step size $h = 1/4$ to approximate the solution on $[0, 1]$. (d) Find the global truncation error at $t = 1$.

6.3 Computer Problems

1. Apply Euler's Method with step sizes $h = 0.1$ and $h = 0.01$ to the initial value problems in Exercise 1. Plot the approximate solutions and the correct solution on $[0, 1]$, and find the global truncation error at $t = 1$. Is the reduction in error for $h = 0.01$ consistent with the order of Euler's Method?

2. Carry out Computer Problem 1 for the Trapezoid Method.

3. Adapt `pend.m` to model the damped pendulum. Run the resulting code with $d = 0.1$. Except for the initial condition $y_1(0) = \pi$, $y_2(0) = 0$, all trajectories move toward the straight-down position as time progresses. Check the exceptional initial condition: Does the simulation agree with theory? with a physical pendulum?

4. Adapt `pend.m` to build a forced, damped version of the pendulum. Run the Trapezoid Method in the following: (a) Set damping $d = 1$ and the forcing parameter $A = 10$. Set the step size $h = 0.005$ and the initial condition of your choice. After moving through some transient behavior, the pendulum will settle into a periodic (repeating) trajectory. Describe this trajectory qualitatively. Try different initial conditions. Do all solutions end up at the same "attracting" periodic trajectory? (b) Now increase the step size to $h = 0.01$, and repeat the experiment. Try initial condition $[\pi/2, 0]$ and others. Describe what happens, and give a reasonable explanation for the anomalous behavior at this step size.

5. Run the forced damped pendulum as in Computer Problem 4, but set $A = 12$. Use the Trapezoid Method with $h = 0.005$. There are now two periodic attractors that are mirror images of one another. Describe the two attracting trajectories, and find two initial conditions $(y_1, y_2) = (a, 0)$ and $(b, 0)$, where $|a - b| \le 0.1$, that are attracted to different periodic trajectories. Set $A = 15$ to view chaotic motion of the forced damped pendulum.

6. Adapt `pend.m` to build a damped pendulum with oscillating pivot. The goal is to investigate the phenomenon of parametric resonance, by which the inverted pendulum becomes stable! The equation is

$$y'' + dy' + \left(\frac{g}{l} + A\cos 2\pi t\right)\sin y = 0,$$

where A is the forcing strength. Set $d = 0.1$ and the length of the pendulum to be 2.5 meters. In the absence of forcing $A = 0$, the downward pendulum $y = 0$ is a stable equilibrium, and the inverted pendulum $y = \pi$ is an unstable equilibrium. Find as accurately as possible the range of parameter A for which the inverted pendulum becomes stable. (Of course, $A = 0$ is too small; it turns out that $A = 30$ is too large.) Use the initial condition $y = 3.1$ for your test, and call the inverted position "stable" if the pendulum does not pass through the downward position.

7. Use the parameter settings of Computer Problem 6 to demonstrate the other effect of parametric resonance: The stable equilibrium can become unstable with an oscillating pivot. Find the smallest (positive) value of the forcing strength A for which this happens. Classify the downward position as unstable if the pendulum eventually travels to the inverted position.

8. Adapt `pend.m` to build the double pendulum. A new pair of `rod` and `bob` must be defined for the second pendulum. Note that the pivot end of the second rod is equal to the formerly free end of the first rod: The (x, y) position of the free end of the second rod can be calculated by using simple trigonometry.

9. Adapt `orbit.m` to solve the two-body problem. Set the masses to $m_1 = 0.3$, $m_2 = 0.03$, and plot the trajectories with initial conditions $(x_1, y_1) = (2, 2)$, $(x_1', y_1') = (0.2, -0.2)$ and $(x_2, y_2) = (0, 0)$, $(x_2', y_2') = (-0.01, 0.01)$.

10. Adapt `orbit.m` to solve the three-body problem. Set the masses to $m_1 = 0.3$, $m_2 = m_3 = 0.03$. (a) Plot the trajectories with initial conditions $(x_1, y_1) = (2, 2)$, $(x_1', y_1') = (0.2, -0.2)$, $(x_2, y_2) = (0, 0)$, $(x_2', y_2') = (0, 0)$ and $(x_3, y_3) = (-2, -2)$, $(x_3', y_3') = (-0.2, 0.2)$. (b) Change the initial condition of x_1' to 0.20001, and compare the resulting trajectories. This is a striking visual example of sensitive dependence.

11. A remarkable three-body figure-eight orbit was discovered by C. Moore in 1993. In this configuration, three bodies of equal mass chase one another along a single figure-eight loop. Set the masses to $m_1 = m_2 = m_3 = 1$ and gravity $g = 1$. (a) Adapt `orbit.m` to plot the trajectory with initial conditions $(x_1, y_1) = (-0.970, 0.243)$, $(x_1', y_1') = (-0.466, -0.433)$, $(x_2, y_2) = (-x_1, -y_1)$, $(x_2', y_2') = (x_1', y_1')$ and $(x_3, y_3) = (0, 0)$, $(x_3', y_3') = (-2x_1', -2y_1')$. (b) Are the trajectories sensitive to small changes in initial conditions? Investigate the effect of changing x_3' by 10^{-k} for $1 \le k \le 5$. For each k, decide whether the figure-eight pattern persists, or a catastrophic change eventually occurs.

6.4 RUNGE–KUTTA METHODS AND APPLICATIONS

The Runge–Kutta Methods are a family of ODE solvers that include the Euler and Trapezoid Methods, and also more sophisticated methods of higher order. In this section, we introduce a variety of one-step methods and apply them to simulate trajectories of some key applications.

6.4.1 The Runge–Kutta family

We have seen that the Euler Method has order one and the Trapezoid Method has order two. In addition to the Trapezoid Method, there are other second-order methods of the Runge–Kutta type. One important example is the Midpoint Method.

Midpoint Method

$$w_0 = y_0$$
$$w_{i+1} = w_i + hf\left(t_i + \frac{h}{2}, w_i + \frac{h}{2}f(t_i, w_i)\right). \qquad (6.46)$$

To verify the order of the Midpoint Method, we must compute its local truncation error. When we did this for the Trapezoid Method, we found the expression (6.31) useful:

$$y_{i+1} = y_i + hf(t_i, y_i) + \frac{h^2}{2}\left(\frac{\partial f}{\partial t}(t_i, y_i) + \frac{\partial f}{\partial y}(t_i, y_i)f(t_i, y_i)\right) + \frac{h^3}{6}y'''(c). \qquad (6.47)$$

To compute the local truncation error at step i, we assume that $w_i = y_i$ and calculate $y_{i+1} - w_{i+1}$. Repeating the use of the Taylor series expansion as for the Trapezoid Method, we can write

$$w_{i+1} = y_i + hf\left(t_i + \frac{h}{2}, y_i + \frac{h}{2}f(t_i, y_i)\right)$$
$$= y_i + h\left(f(t_i, y_i) + \frac{h}{2}\frac{\partial f}{\partial t}(t_i, y_i) + \frac{h}{2}f(t_i, y_i)\frac{\partial f}{\partial y}(t_i, y_i) + O(h^2)\right). \qquad (6.48)$$

Comparing (6.47) and (6.48) yields

$$y_{i+1} - w_{i+1} = O(h^3),$$

so the Midpoint Method is of order two by Theorem 6.4.

Each function evaluation of the right-hand side of the differential equation is called a **stage** of the method. The Trapezoid and Midpoint Methods are members of the family of two-stage, second-order Runge–Kutta Methods, having form

$$w_{i+1} = w_i + h\left(1 - \frac{1}{2\alpha}\right)f(t_i, w_i) + \frac{h}{2\alpha}f(t_i + \alpha h, w_i + \alpha hf(t_i, w_i)) \qquad (6.49)$$

for some $\alpha \neq 0$. Setting $\alpha = 1$ corresponds to the Explicit Trapezoid Method and $\alpha = 1/2$ to the Midpoint Method. Exercise 5 asks you to verify the order of methods in this family.

Figure 6.14 illustrates the intuition behind the Trapezoid and Midpoint Methods. The Trapezoid Method uses an Euler step to the right endpoint of the interval, evaluates the slope there, and then averages with the slope from the left endpoint. The Midpoint Method uses an Euler step to move to the midpoint of the interval, evaluates the slope there as $f(t_i + h/2, w_i + (h/2)f(t_i, w_i))$, and uses that slope to move from w_i to the new approximation w_{i+1}. These methods use different approaches to solving the same problem: acquiring a slope that represents the entire interval better than the Euler Method, which uses only the slope estimate from the left end of the interval.

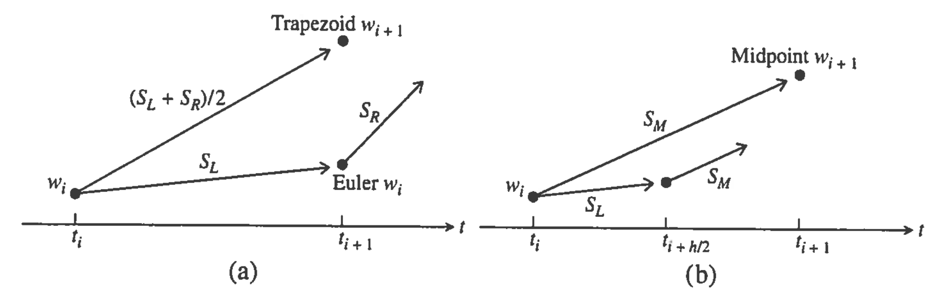

Figure 6.14 Schematic view of two members of the RK2 family. (a) The Trapezoid Method uses an average from the left and right endpoints to traverse the interval. (b) The Midpoint Method uses a slope from the interval midpoint.

from a lower warm medium (such as the ground) to a higher cool medium (like the upper atmosphere). In this model of a two-dimensional atmosphere, a circulation of air develops that can be described by the following system of three equations:

$$\begin{aligned} x' &= -sx + sy \\ y' &= -xz + rx - y \\ z' &= xy - bz. \end{aligned} \tag{6.53}$$

The variable x denotes the clockwise circulation velocity, y measures the temperature difference between the ascending and descending columns of air, and z measures the deviation from a strictly linear temperature profile in the vertical direction. The Prandtl number s, the Rayleigh number r, and b are parameters of the system. The most common setting for the parameters is $s = 10, r = 28$, and $b = 8/3$. These settings were used for the trajectory shown in Figure 6.17, computed by order four Runge–Kutta, using the following code to describe the differential equation.

```
function z=ydot(t,y)
%Lorenz equations
s=10; r=28; b=8/3;
z(1)=-s*y(1)+s*y(2);
z(2)=-y(1)*y(3)+r*y(1)-y(2)
z(3)=y(1)*y(2)-b*y(3)
```

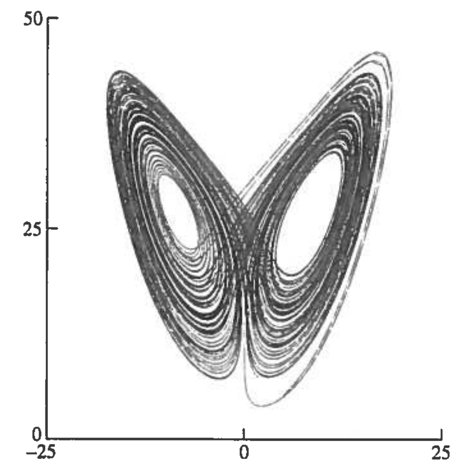

Figure 6.17 One trajectory of the Lorenz equations (6.53), projected to the xz-plane. Parameters are set to $s = 10, r = 28$, and $b = 8/3$.

The Lorenz equations are an important example because the trajectories show great complexity, despite the fact that the equations are deterministic and fairly simple (almost linear). The explanation for the complexity is similar to that of the double pendulum or three-body problem: sensitive dependence on initial conditions. Computer Problems 12 and 13 explore the sensitive dependence of this so-called chaotic attractor.

6.4 Exercises

1. Apply the Midpoint Method for the IVPs

   (a) $y' = t$ (b) $y' = t^2 y$ (c) $y' = 2(t+1)y$

   (d) $y' = 5t^4 y$ (e) $y' = 1/y^2$ (f) $y' = t^3/y^2$

   with initial condition $y(0) = 1$. Using step size $h = 1/4$, calculate the Midpoint Method approximation on the interval $[0, 1]$. Compare with the correct solution found in Exercise 6.1.3, and find the global truncation error at $t = 1$.

2. Carry out the steps of Exercise 1 for the IVPs

   (a) $y' = t + y$ (b) $y' = t - y$ (c) $y' = 4t - 2y$

   with initial condition $y(0) = 0$. The exact solutions were found in Exercise 6.1.4.

3. Apply fourth-order Runge–Kutta Method to the IVPs in Exercise 1. Using step size $h = 1/4$, calculate the approximation on the interval $[0, 1]$. Compare with the correct solution found in Exercise 6.1.3, and find the global truncation error at $t = 1$.

4. Carry out the steps of Exercise 3 for the IVPs in Exercise 2.

5. Prove that for any $\alpha \neq 0$, the method (6.49) is second order.

6. Consider the initial value problem $y' = \lambda y$. The solution is $y(t) = y_0 e^{\lambda t}$. (a) Calculate w_1 for RK4 in terms of w_0 for this differential equation. (b) Calculate the local truncation error by setting $w_0 = y_0 = 1$ and determining $y_1 - w_1$. Show that the local truncation error is of size $O(h^5)$, as expected for a fourth-order method.

7. Assume that the right-hand side $f(t, y) = f(t)$ does not depend on y. Show that $s_2 = s_3$ in fourth-order Runge–Kutta and that RK4 is equivalent to Simpson's Rule for the integral $\int_{t_i}^{t_i+h} f(s)\, ds$.

6.4 Computer Problems

1. Apply the Midpoint Method on a grid of step size $h = 0.1$ in $[0, 1]$ for the initial value problems in Exercise 1. Print a table of the t values, approximations, and global truncation error at each step.

2. Apply the fourth-order Runge–Kutta Method solution on a grid of step size $h = 0.1$ in $[0, 1]$ for the initial value problems in Exercise 1. Print a table of the t values, approximations, and global truncation error at each step.

3. Carry out the steps of Computer Problem 2, but plot the approximate solutions on $[0, 1]$ for step sizes $h = 0.1, 0.05$, and 0.025, along with the true solution.

4. Carry out the steps of Computer Problem 2 for the equations of Exercise 2.

5. Plot the fourth-order Runge–Kutta Method approximate solution on $[0, 1]$ for the differential equation $y' = 1 + y^2$ and initial condition (a) $y_0 = 0$ (b) $y_0 = 1$, along with the exact solution (see Exercise 6.1.7). Use step sizes $h = 0.1$ and 0.05.

6. Plot the fourth-order Runge–Kutta Method approximate solution on $[0, 1]$ for the differential equation $y' = 1 - y^2$ and initial condition (a) $y_0 = 0$ (b) $y_0 = -1/2$, along with the exact solution (see Exercise 6.1.8). Use step sizes $h = 0.1$ and 0.05.

7. Calculate the fourth-order Runge–Kutta Method approximate solution on $[0, 4]$ for the differential equation $y' = \sin y$ and initial condition (a) $y_0 = 0$ (b) $y_0 = 100$, using step sizes $h = 0.1 \times 2^{-k}$ for $0 \le k \le 5$. Plot the $k = 0$ and $k = 5$ approximate solutions along with the exact solution (see Exercise 6.1.15), and make a log–log plot of the error as a function of h.

8. Calculate the fourth-order Runge–Kutta Method approximate solution of the differential equation $y' = \sinh y$ and initial condition (a) $y_0 = 1/4$ on the interval $[0, 2]$ (b) $y_0 = 2$ on the interval $[0, 1/4]$, using step sizes $h = 0.1 \times 2^{-k}$ for $0 \le k \le 5$. Plot the $k = 0$ and $k = 5$ approximate solutions along with the exact solution (see Exercise 6.1.16), and make a log–log plot of the error as a function of h.

9. For the IVPs in Exercise 1, plot the global error of the RK4 method at $t = 1$ as a function of h, as in Figure 6.4.

10. Consider the Hodgkin–Huxley equations (6.52) with default parameters. (a) Find as accurately as possible the minimum threshold, in microamps, for generating a spike with a 1 msec pulse. (b) Does the answer change if the pulse is 5 msec long? (c) Experiment with the shape of the pulse. Does a triangular pulse of identical enclosed area cause the same effect as a square pulse? (d) Discuss the existence of a threshold for constant sustained input.

11. Adapt the `orbit.m` MATLAB program to animate a solution to the Lorenz equations by the order four Runge–Kutta Method with step size $h = 0.001$. Draw the trajectory with initial condition $(x_0, y_0, z_0) = (5, 5, 5)$.

12. Assess the conditioning of the Lorenz equations by following two trajectories from two nearby initial conditions. Consider the initial conditions $(x, y, z) = (5, 5, 5)$ and another initial condition at a distance $\Delta = 10^{-5}$ from the first. Compute both trajectories by fourth-order Runge–Kutta with step size $h = 0.001$, and calculate the error magnification factor after $t = 10$ and $t = 20$ time units.

13. Follow two trajectories of the Lorenz equations with nearby initial conditions, as in Computer Problem 12. For each, construct the binary symbol sequence consisting of 0 if the trajectory traverses the "negative x" loop in Figure 6.17 and 1 if it traverses the positive loop. For how many time units do the symbol sequences of the two trajectories agree?

Reality Check 6 The Tacoma Narrows Bridge

A mathematical model that attempts to capture the Tacoma Narrows Bridge incident was proposed by McKenna and Tuama [2001]. The goal is to explain how torsional, or twisting, oscillations can be magnified by forcing that is strictly vertical.

Consider a roadway of width $2l$ hanging between two suspended cables, as in Figure 6.18(a). We will consider a two-dimensional slice of the bridge, ignoring the dimension of the bridge's length for this model, since we are only interested in the side-to-side motion. At rest, the roadway hangs at a certain equilibrium height due to gravity; let y denote the current distance the center of the roadway hangs below this equilibrium.

Hooke's law postulates a linear response, meaning that the restoring force the cables apply will be proportional to the deviation. Let θ be the angle the roadway makes with the horizontal. There are two suspension cables, stretched $y - l\sin\theta$ and $y + l\sin\theta$ from equilibrium, respectively. Assume that a viscous damping term is given that is proportional to the velocity. Using Newton's law $F = ma$ and denoting Hooke's constant by K, the equations of motion for y and θ are as follows:

$$y'' = -dy' - \left[\frac{K}{m}(y - l\sin\theta) + \frac{K}{m}(y + l\sin\theta)\right]$$
$$\theta'' = -d\theta' + \frac{3\cos\theta}{l}\left[\frac{K}{m}(y - l\sin\theta) - \frac{K}{m}(y + l\sin\theta)\right].$$

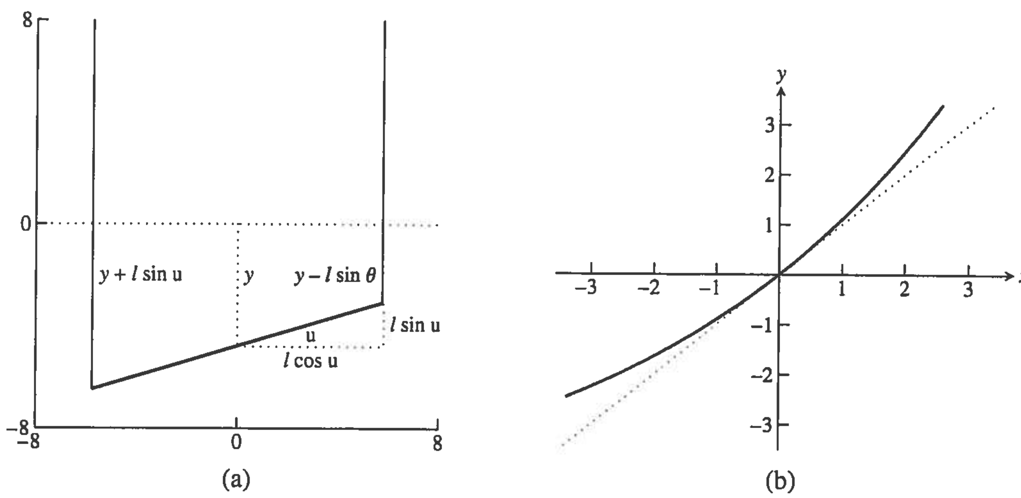

Figure 6.18 Schematics for the McKenna-Tuama model of the Tacoma Narrows Bridge. (a) Denote the distance from the roadway center of mass to its equilibrium position by y, and the angle of the roadway with the horizontal by θ. (b) Exponential Hooke's law curve $f(y) = (K/a)(e^{ay} - 1)$.

However, Hooke's law is designed for springs, where the restoring force is more or less equal whether the springs are compressed or stretched. McKenna and Tuama hypothesize that cables pull back with more force when stretched than they push back when compressed. (Think of a string as an extreme example.) They replace the linear Hooke's law restoring force $f(y) = Ky$ with a nonlinear force $f(y) = (K/a)(e^{ay} - 1)$, as shown in Figure 6.18(b). Both functions have the same slope K at $y = 0$; but for the nonlinear force, a positive y (stretched cable) causes a stronger restoring force than the corresponding negative y (slackened cable). Making this replacement in the preceding equations yields

$$y'' = -dy' - \frac{K}{ma}\left[e^{a(y-l\sin\theta)} - 1 + e^{a(y+l\sin\theta)} - 1\right]$$
$$\theta'' = -d\theta' + \frac{3\cos\theta}{l}\frac{K}{ma}\left[e^{a(y-l\sin\theta)} - e^{a(y+l\sin\theta)}\right]. \quad (6.54)$$

As the equations stand, the state $y = y' = \theta = \theta' = 0$ is an equilibrium. Now turn on the wind. Add the forcing term $0.2W \sin\omega t$ to the right-hand side of the y equation, where W is the wind speed in km/hr. This adds a strictly vertical oscillation to the bridge.

Useful estimates for the physical constants can be made. The mass of a one-foot length of roadway was about 2500 kg, and the spring constant K has been estimated at 1000 Newtons. The roadway was about 12 meters wide. For this simulation, the damping coefficient was set at $d = 0.01$, and the Hooke's nonlinearity coefficient $a = 0.2$. An observer counted 38 vertical oscillations of the bridge in one minute shortly before the collapse—set $\omega = 2\pi(38/60)$. These coefficients are only guesses, but they suffice to show ranges of motion that tend to match photographic evidence of the bridge's final oscillations. MATLAB code that runs the model (6.54) is as follows:

```
%Program 6.6 Animation program for bridge using IVP solver
%Inputs: time interval inter,
% ic=[y(1,1) y(1,2) y(1,3) y(1,4)],
% number of steps n, steps per point plotted p
%Calls a one-step method such as trapstep.m
%Example usage: tacoma([0 1000],[1 0 0.001 0],25000,5)
function tacoma(inter,ic,n,p)
```

greatly increases solving efficiency. For example, note the difference in steps needed when one of MATLAB's stiff solvers are used:

```
>> opts=odeset('RelTol',1e-4);
>> [t,y]=ode23s(@(t,y) 10*(1-y),[0 100],.5,opts);
>> length(t)

ans=
        39
```

Figure 6.20(b) plots the solution points from the solver ode23s. Relatively few points are needed to keep the numerical solution within the tolerance. We will investigate how to build methods that handle this type of difficulty in the next section. ◄

6.5 Computer Problems

1. Write a MATLAB implementation of RK23 (Example 6.19), and apply to approximating the solutions of the IVPs in Exercise 6.1.3 with a relative tolerance of 10^{-8} on $[0, 1]$. Ask the program to stop exactly at the endpoint $t = 1$. Report the maximum step size used and the number of steps.
2. Compare the results of Computer Problem 1 with the application of MATLAB's ode23 to the same problem.
3. Carry out the steps of Computer Problem 1 for the Runge–Kutta–Fehlberg Method RKF45.
4. Compare the results of Computer Problem 3 with the application of MATLAB's ode45 to the same problem.
5. Apply a MATLAB implementation of RKF45 to approximating the solutions of the systems in Exercise 6.3.1 with a relative tolerance of 10^{-6} on $[0, 1]$. Report the maximum step size used and the number of steps.

6.6 IMPLICIT METHODS AND STIFF EQUATIONS

The differential equations solvers we have presented so far are **explicit**, meaning that there is an explicit formula for the new approximation w_{i+1} in terms of known data, such as h, t_i, and w_i. It turns out that some differential equations are poorly served by explicit methods, and our first goal is to explain why. In Example 6.23, a sophisticated variable step-size solver seems to spend most of its energy overshooting the correct solution in one direction or another.

The stiffness phenomenon can be more easily understood in a simpler context. Accordingly, we begin with Euler's Method.

► **EXAMPLE 6.24** Apply Euler's Method to Example 6.23.

Euler's Method for the right-hand side $f(t, y) = 10(1 - y)$ with step size h is

$$\begin{aligned} w_{i+1} &= w_i + hf(t_i, w_i) \\ &= w_i + h(10)(1 - w_i) \\ &= w_i(1 - 10h) + 10h. \end{aligned} \tag{6.68}$$

Since the solution is $y(t) = 1 - e^{-10t}/2$, the approximate solution must approach 1 in the long run. Here we get some help from Chapter 1. Notice that (6.68) can be viewed as a fixed-point iteration with $g(x) = x(1 - 10h) + 10h$. This iteration will converge to the fixed point at $x = 1$ as long as $|g'(1)| = |1 - 10h| < 1$. Solving this inequality yields $0 < h < 0.2$. For any larger h, the fixed point 1 will repel nearby guesses, and the solution will have no hope of being accurate. ◄

Figure 6.21 shows this effect for Example 6.24. The solution is very tame: an attracting equilibrium at $y = 1$. An Euler step of size $h = 0.3$ has difficulty finding the equilibrium because the slope of the nearby solution changes greatly between the beginning and the end of the h interval. This causes overshoot in the numerical solution.

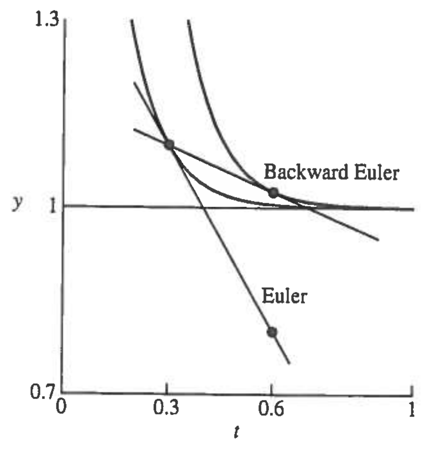

Figure 6.21 Comparison of Euler and Backward Euler steps. The differential equation in Example 6.23 is stiff. The equilibrium solution $y = 1$ is surrounded by other solutions with large curvature (fast-changing slope). The Euler step overshoots, while the Backward Euler step is more consistent with the system dynamics.

Differential equations with this property—that attracting solutions are surrounded with fast-changing nearby solutions—are called **stiff**. This is often a sign of multiple timescales in the system. Quantitatively, it corresponds to the linear part of the right-hand side f of the differential equation, in the variable y, being large and negative. (For a system of equations, this corresponds to an eigenvalue of the linear part being large and negative.) This definition is a bit relative, but that is the nature of stiffness—the more negative, the smaller the step size must be to avoid overshoot. For Example 6.24, stiffness is measured by evaluating $\partial f/\partial y = -10$ at the equilibrium solution $y = 1$.

One way to solve the problem depicted in Figure 6.21 is to somehow bring in information from the right side of the interval $[t_i, t_i + h]$, instead of relying solely on information from the left side. That is the motivation behind the following variation on Euler's Method:

Backward Euler Method

$$\begin{aligned} w_0 &= y_0 \\ w_{i+1} &= w_i + hf(t_{i+1}, w_{i+1}). \end{aligned} \tag{6.69}$$

Note the difference: While Euler's Method uses the left-end slope to step across the interval, Backward Euler would like to somehow cross the interval so that the slope is correct at the right end.

A price must be paid for this improvement. Backward Euler is our first example of an **implicit** method, meaning that the method does not directly give a formula for the new approximation w_{i+1}. Instead, we must work a little to get it. For the example $y' = 10(1 - y)$, the Backward Euler Method gives

$$w_{i+1} = w_i + 10h(1 - w_{i+1}),$$

which, after a little algebra, can be expressed as

$$w_{i+1} = \frac{w_i + 10h}{1 + 10h}.$$

Setting $h = 0.3$, for example, the Backward Euler Method gives $w_{i+1} = (w_i + 3)/4$. We can again evaluate the behavior as a fixed point iteration $w \to g(w) = (w + 3)/4$. There is a fixed point at 1, and $g'(1) = 1/4 < 1$, verifying convergence to the true equilibrium solution $y = 1$. Unlike the Euler Method with $h = 0.3$, at least the correct qualitative behavior is followed by the numerical solution. In fact, note that the Backward Euler Method solution converges to $y = 1$ no matter how large the step size h (Exercise 3).

Because of the better behavior of implicit methods like Backward Euler in the presence of stiff equations, it is worthwhile performing extra work to evaluate the next step, even though it is not explicitly available. Example 6.24 was not challenging to solve for w_{i+1}, due to the fact that the differential equation is linear, and it was possible to change the original implicit formula to an explicit one for evaluation. In general, however, this is not possible, and we need to use more indirect means.

If the implicit method leaves a nonlinear equation to solve, we must refer to Chapter 1. Both Fixed-Point Iteration and Newton's Method are often used to solve for w_{i+1}. This means that there is an equation-solving loop within the loop advancing the differential equation. The next example shows how this can be done.

▶ EXAMPLE 6.25 Apply the Backward Euler Method to the initial value problem

$$\begin{cases} y' = y + 8y^2 - 9y^3 \\ y(0) = 1/2 \\ t \text{ in } [0,3]. \end{cases}$$

This equation, like the previous example, has an equilibrium solution $y = 1$. The partial derivative $\partial f/\partial y = 1 + 16y - 27y^2$ evaluates to -10 at $y = 1$, identifying this equation as moderately stiff. There will be an upper bound, similar to that of the previous example, for h, such that Euler's Method is successful. Thus, we are motivated to try the Backward Euler Method

$$\begin{aligned} w_{i+1} &= w_i + hf(t_{i+1}, w_{i+1}) \\ &= w_i + h(w_{i+1} + 8w_{i+1}^2 - 9w_{i+1}^3). \end{aligned}$$

This is a nonlinear equation in w_{i+1}, which we need to solve in order to advance the numerical solution. Renaming $z = w_{i+1}$, we must solve the equation $z = w_i + h(z + 8z^2 - 9z^3)$, or

$$9hz^3 - 8hz^2 + (1 - h)z - w_i = 0 \tag{6.70}$$

for the unknown z. We will demonstrate with Newton's Method.

To start Newton's Method, an initial guess is needed. Two choices that come to mind are the previous approximation w_i and the Euler's Method approximation for w_{i+1}. Although the latter is accessible since Euler is explicit, it may not be the best choice for stiff problems, as shown in Figure 6.21. In this case, we will use w_i as the starting guess.

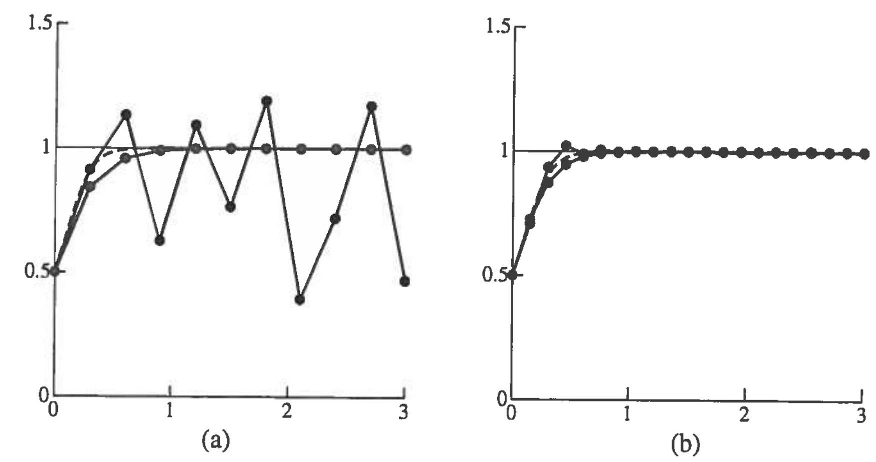

Figure 6.22 Numerical solution of the initial value problem of Example 6.25. True solution is the dashed curve. The black circles denote the Euler Method approximation; the blue circles denote Backward Euler. (a) $h = 0.3$ (b) $h = 0.15$.

Assembling Newton's Method for (6.70) yields

$$z_{new} = z - \frac{9hz^3 - 8hz^2 + (1 - h)z - w_i}{27hz^2 - 16hz + 1 - h}. \tag{6.71}$$

After evaluating (6.71), replace z with z_{new} and repeat. For each Backward Euler step, Newton's Method is run until $z_{new} - z$ is smaller than a preset tolerance (smaller than the errors that are being made in approximating the differential equation solution).

Figure 6.22 shows the results for two different step sizes. In addition, numerical solutions from Euler's Method are shown. Clearly, $h = 0.3$ is too large for Euler on this stiff problem. On the other hand, when h is cut to 0.15, both methods perform at about the same level. ◀

So-called stiff solvers like Backward Euler allow sufficient error control with comparatively large step size, increasing efficiency. MATLAB's `ode23s` is a higher order version with a built-in variable step-size strategy.

6.6 Exercises

1. Using initial condition $y(0) = 0$ and step size $h = 1/4$, calculate the Backward Euler approximation on the interval [0, 1]. Find the error at $t = 1$ by comparing with the correct solution found in Exercise 6.1.4.

   (a) $y' = t + y$ (b) $y' = t - y$ (c) $y' = 4t - 2y$

2. Find all equilibrium solutions and the value of the Jacobian at the equilibria. Is the equation stiff? (a) $y' = y - y^2$ (b) $y' = 10y - 10y^2$ (c) $y' = -10 \sin y$

3. Show that for every step size h, the Backward Euler approximate solution converges to the equilibrium solution $y = 1$ as $t_i \to \infty$ for Example 6.24.

4. Consider the linear differential equation $y' = ay + b$ for $a < 0$. (a) Find the equilibrium. (b) Write down the Backward Euler Method for the equation. (c) View Backward Euler as a Fixed-Point Iteration to prove that the method's approximate solution will converge to the equilibrium as $t \to \infty$.

6.6 Computer Problems

1. Apply Backward Euler, using Newton's Method as a solver, for the initial value problems. Which of the equilibrium solutions are approached by the approximate solution? Apply Euler's Method. For what approximate range of h can Euler be used successfully to converge to the equilibrium? Plot approximate solutions given by Backward Euler, and by Euler with an excessive step size.

$$\text{(a)} \begin{cases} y' = y^2 - y^3 \\ y(0) = 1/2 \\ t \text{ in } [0,20] \end{cases} \qquad \text{(b)} \begin{cases} y' = 6y - 6y^2 \\ y(0) = 1/2 \\ t \text{ in } [0,20] \end{cases}$$

2. Carry out the steps in Computer Problem 1 for the following initial value problems:

$$\text{(a)} \begin{cases} y' = 6y - 3y^2 \\ y(0) = 1/2 \\ t \text{ in } [0,20] \end{cases} \qquad \text{(b)} \begin{cases} y' = 10y^3 - 10y^4 \\ y(0) = 1/2 \\ t \text{ in } [0,20] \end{cases}$$

6.7 MULTISTEP METHODS

The Runge–Kutta family that we have studied consists of one-step methods, meaning that the newest step w_{i+1} is produced on the basis of the differential equation and the value of the previous step w_i. This is in the spirit of initial value problems, for which Theorem 6.2 guarantees a unique solution starting at an arbitrary w_0.

The multistep methods suggest a different approach: using the knowledge of more than one of the previous w_i to help produce the next step. This will lead to ODE solvers that have order as high as the one-step methods, but much of the necessary computation will be replaced with interpolation of already computed values on the solution path.

6.7.1 Generating multistep methods

As a first example, consider the following two-step method:

Adams–Bashforth Two-Step Method

$$w_{i+1} = w_i + h\left[\frac{3}{2}f(t_i, w_i) - \frac{1}{2}f(t_{i-1}, w_{i-1})\right]. \tag{6.72}$$

While the second-order Midpoint Method,

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

needs two function evaluations of the ODE right-hand side f per step, the Adams–Bashforth Two-Step Method requires only one new evaluation per step (one is stored from the previous step). We will see subsequently that (6.72) is also a second-order method. Therefore, multistep methods can achieve the same order with less computational effort—usually just one function evaluation per step.

Since multistep methods use more than one previous w value, they need help getting started. The start-up phase for an s-step method typically consists of a one-step method that uses w_0 to produce $s-1$ values $w_1, w_2, \ldots, w_{s-1}$, before the multistep method can be used. The Adams–Bashforth Two-Step Method (6.72) needs w_1, along with the given initial condition w_0, in order to begin. The following MATLAB code uses the Trapezoid Method to provide the start-up value w_1.


```
% Program 6.7 Multistep method
% Inputs: time interval inter,
%   ic=[y0]  initial condition, number of steps n,
%   s=number of (multi)steps, e.g. 2 for 2-step method
% Output: time steps t, solution y
% Calls a multistep method such as ab2step.m
% Example usage: [t,y]=exmultistep([0,1],1,20,2)
function [t,y]=exmultistep(inter,ic,n,s)
h=(inter(2)-inter(1))/n;
% Start-up phase
y(1,:)=ic;t(1)=inter(1);
for i=1:s-1                % start-up phase, using one-step method
  t(i+1)=t(i)+h;
  y(i+1,:)=trapstep(t(i),y(i,:),h);
  f(i,:)=ydot(t(i),y(i,:));
end
for i=s:n                  % multistep method loop
  t(i+1)=t(i)+h;
  f(i,:)=ydot(t(i),y(i,:));
  y(i+1,:)=ab2step(t(i),i,y,f,h);
end
plot(t,y)

function y=trapstep(t,x,h)
%one step of the Trapezoid Method from section 6.2
z1=ydot(t,x);
g=x+h*z1;
z2=ydot(t+h,g);
y=x+h*(z1+z2)/2;

function z=ab2step(t,i,y,f,h)
%one step of the Adams-Bashforth 2-step method
z=y(i,:)+h*(3*f(i,:)/2-f(i-1,:)/2);

function z=unstable2step(t,i,y,f,h)
%one step of an unstable 2-step method
z=-y(i,:)+2*y(i-1,:)+h*(5*f(i,:)/2+f(i-1,:)/2);

function z=weaklystable2step(t,i,y,f,h)
%one step of a  weakly-stable 2-step method
z=y(i-1,:)+h*2*f(i,:);

function z=ydot(t,y)   % IVP from section 6.1
z=t*y+t^3;
```

Figure 6.23(a) shows the result of applying the Adams–Bashforth Two-Step Method to the initial value problem (6.5) from earlier in the chapter, using step size $h = 0.05$ and applying the Trapezoid Method for start-up. Part (b) of the figure shows the use of a different two-step method. Its instability will be the subject of our discussion of stability analysis in the next sections.

A general s-step method has the form

$$\begin{aligned} w_{i+1} = a_1 w_i + a_2 w_{i-1} + \cdots + a_s w_{i-s+1} + h[b_0 f_{i+1} + b_1 f_i \\ + b_2 f_{i-1} + \cdots + b_s f_{i-s+1}]. \end{aligned} \tag{6.73}$$