19. Consider a double pendulum, as shown in Figure 3.11. Its Lagrangian is

$$\frac{m_1+m_2}{2}L_1^2\dot{\theta}_1^2 + \frac{m_2}{2}L_2^2\dot{\theta}_2^2 + m_2 g L_2\cos\theta_2$$
$$+ m_2 L_1 L_2 \dot{\theta}_1\dot{\theta}_2\cos(\theta_1-\theta_2) + (m_1+m_2)gL_1\cos\theta_1$$

where $\dot{\theta} \equiv d\theta/dt$. (a) Use the Lagrangian to find the equations of motion. [Pencil] (b) Write a program that uses adaptive Runge-Kutta to simulate the motion of the double pendulum. Take $g = 9.81$ m/s^2, $m_1 = m_2$, and $L_1 = L_2 = 0.1$ m; compute examples of the motion for various initial conditions. Show that in some cases, the lower mass spins completely around with an aperiodic motion. [Computer]

3.4 *CHAOS IN THE LORENZ MODEL

Optional sections, marked with an asterisk, may be omitted without loss of continuity.

Unwinding the Mechanical Universe

Newton's success in solving the Kepler problem had effects far beyond physics. It inspired the mechanistic picture of the universe, a philosophy developed by Laplace and others. The orbits of the planets had the regularity of a well-made clock. Even long-term events, such as solar eclipses and comet returns, were predictable to high accuracy. For centuries it was believed that other physical phenomena, such as weather, were only unpredictable due to the large number of variables in the problem. With the arrival of modern computers it was hoped that long-range weather prediction would soon be within our grasp.

In the early 1960s, however, an MIT meteorologist named Ed Lorenz saw that it would not be so. He found that the weather was intrinsically unpredictable, not because of its complexity, but because of the nonlinear nature of the governing equations. Lorenz formulated a simple model of the global weather, reducing the problem to a 12-variable system of nonlinear ODEs. What he observed was aperiodic behavior that was extremely sensitive to the initial conditions.†

To study this effect more easily, he introduced an even simpler model with only three variables. The Lorenz model [117] is

$$\begin{aligned}\frac{dx}{dt} &= \sigma(y-x)\\ \frac{dy}{dt} &= rx - y - xz \qquad (3.32)\\ \frac{dz}{dt} &= xy - bz\end{aligned}$$

where σ, r, and b are positive constants. These simple equations were originally developed as a model for buoyant convection in a fluid. Their derivation

†For a historical account of Lorenz's discovery, see Gleick [56].

Table 3.6: Outline of program `lorenz`, which computes the time series for the Lorenz model (3.32).

- Set initial state $[x,y,z]$ and parameters $[r,\sigma,b]$.
- Loop over the desired number of steps.
  - Record values of x, y, z, t, and τ for plotting.
  - Find new state using `rka`, the adaptive Runge-Kutta function.
- Print maximum and minimum time step returned by `rka`.
- Graph the time series $x(t)$.
- Graph the (x,y,z) phase space trajectory.

See pages 94 and 102 for program listings.

Table 3.7: Outline of function `lorzrk`, which is used by the Runge-Kutta routines to evaluate the Lorenz equations.

- *Inputs*: $\mathbf{x}(t)$, t (not used), $[r,\sigma,b]$.
- *Output*: $d\mathbf{x}(t)/dt$.
- Compute $d\mathbf{x}(t)/dt = [dx/dt\ dy/dt\ dz/dt]$ (see (3.32)).

See pages 96 and 104 for program listings.

is beyond the scope of this text, but, briefly, x measures the rate of convective overturning, and y and z measure the horizontal and vertical temperature gradients. The parameters σ and b depend on the fluid properties and the geometry of the container; commonly, the values $\sigma = 10$ and $b = 8/3$ are used. The parameter r is proportional to the applied temperature gradient.

Lorenz Model Program

A program, called `lorenz`, which solves the Lorenz model using our adaptive Runge-Kutta method is outlined in Table 3.6. This program does little more than repeatedly call `rka` and graph the results. The function `lorzrk` (see Table 3.7) specifies Equations (3.32) for use in the Runge-Kutta routines.

Although an adaptive scheme has many advantages, it may not always be the best method to use for a particular problem. Note that for the runs of the

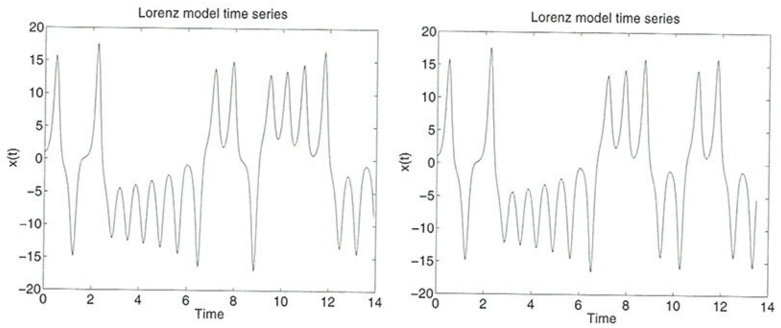

Figure 3.12: Time series $x(t)$ for the Lorenz model as computed by `lorenz`. The parameters are $\sigma = 10$, $b = 8/3$, and $r = 28$. The initial condition is $[x\ y\ z] = [1\ 1\ 20]$ for the plot on the left and $[x\ y\ z] = [1\ 1\ 20.01]$ for the plot on the right.

`lorenz` program described below, the value of the time step τ does not vary by much more than one order of magnitude. This is one argument for returning to our nonadaptive methods. Another is that nonadaptive methods automatically produce data points that are evenly spaced in time, the form required by most data analysis techniques. In an exercise, you are asked to make a comparison between simple and adaptive Runge-Kutta for the Lorenz problem and judge for yourself.

Two examples of time series obtained by the `lorenz` program are shown in Figure 3.12. The values of $x(t)$ oscillate in a fashion that does not seem much more complicated than simple harmonic motion; results for $y(t)$ and $z(t)$ are similar. However, in Chapter 5 we analyze the power spectra for these time series and find they have a complex structure. More importantly, Figure 3.12 shows that slightly different initial conditions will produce significantly different time series. Comparing the two plots of $x(t)$ shows that the evolution is initially very similar, but later the two time series are completely different. This extreme sensitivity to initial conditions led Lorenz to speculate that, if weather obeyed similar dynamics, long-term prediction was impossible. He termed this the butterfly effect: Even a single butterfly flapping its wings could, in the long run, influence the world's weather. Because the trajectories of the Lorenz model are extremely sensitive to initial conditions, the motion is considered chaotic.

Figure 3.13 shows the trajectory in the three dimensional space of the variables x, y, and z. Now the motion looks far more interesting. The trajectory is said to lie on an attractor; you may think of this motion as a sort of aperiodic orbit. This picture helps us understand the butterfly effect and the origin of the chaotic motion. The center portion of the attractor mixes trajectories, sending some to the left lobe, some to the right. Trajectories with nearly identical initial conditions are eventually separated in much the same way as adjacent particles of flour are separated in the kneading of bread.[63] The three points marked by asterisks in Figure 3.13 are the steady states of the Lorenz model; their definition is discussed in the next chapter.

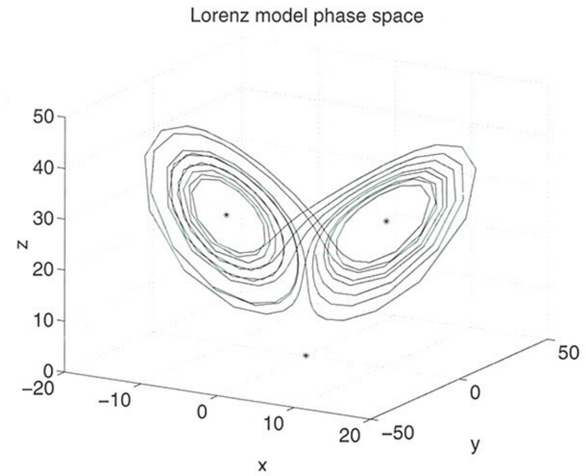

Figure 3.13: Phase space trajectory for the Lorenz model. The initial condition is $[x\ y\ z] = [1\ 1\ 20]$; parameters are the same as in Figure 3.12. Steady states are indicated by asterisks.

Although many celestial mechanics problems are accurately modeled using two-body interactions, objects moving in our solar system experience a gravitational attraction to all the planets. Specifically, the orbits of highly elliptical comets can be significantly influenced by the gas giants, especially Jupiter, and, given these perturbations, their motion may actually be chaotic![32, 97]

EXERCISES

20. Try running the `lorenz` program with the following values for the parameter r: (a) 0, (b) 1, (c) 14, (d) 20, (e) 100. Use the initial condition $[x\ y\ z] = [1\ 1\ 20]$. Describe the different types of behavior found and compare with Figure 3.13. [Computer]

21. For $r = 28$, try the following initial conditions: $[x\ y\ z]$ = (a) [0 0 0]; (b) [0 0 20]; (c) [0.01 0.01 0.01]; (d) [100 100 100]; (e) [8.5 8.5 27]. Describe the different types of behavior found and compare with Figure 3.13. [Computer]

22. The following set of nonlinear ODEs is known as the Lotka-Volterra model:

$$\frac{dx}{dt} = (a - bx - cy)x; \qquad \frac{dy}{dt} = (-d + ex)y$$

where a, b, c, d, and e are positive constants. (a) These equations model a simple ecological system of predators and prey.[99] For example, the variables x and y could represent the number of hares and foxes in a forest. Describe the physical meaning of each of the five parameters. [Pencil] (b) Write a program using adaptive Runge-Kutta to compute the trajectory $(x(t), y(t))$ and plot $y(t)$ versus $x(t)$ for a variety of initial conditions using $a = 10$, $b = 10^{-5}$, $c = 0.1$, $d = 10$, and $e = 0.1$. Take $x(0) > 0$, $y(0) > 0$, since the number of animals should be positive. [Computer]

23. Consider the Hopf model, given by the nonlinear ODEs:

$$\frac{dx}{dt} = ax + y - x(x^2 + y^2); \qquad \frac{dy}{dt} = -x + ay - y(x^2 + y^2)$$

(a) By transforming these equations into polar coordinates show that when $a < 0$, trajectories spiral toward the origin, and when $a > 0$ they spiral toward a circle of radius $\sqrt{a}$ centered at the origin. [Pencil] (b) Write a program that uses the adaptive Runge-Kutta routine to compute the trajectories of the Hopf model. Plot these trajectories and confirm the result proven in part (a). [Computer]

24. Write a nonadaptive version of the `lorenz` program that uses `rk4`. Run the nonadaptive version using the minimum time step used by the adaptive version. Remember that `rka` is effectively using a time step of $\frac{1}{2}\tau$, since this is the step size for the small steps. Modify the main loop so that the iteration stops at $t = 10$. Determine the relative efficiency of the two methods (use the MATLAB `flops` command to count floating-point operations or the C++ timing routines in the `<time.h>` library). [Computer]

25. One characteristic of chaotic dynamics is sensitivity to initial conditions. Using `rk4`, write a nonadaptive version of the `lorenz` program that simultaneously computes the trajectories for two different initial conditions. Use initial conditions that are very close together (e.g., [1 1 20] and [1 1 20.001]). Plot the distance between these trajectories as a function of time, using both normal and logarithmic scales. What can you say about how the distance varies with time? [Computer]

26. Repeat the previous exercise using the: (a) Lotka-Volterra equations (see Exercise 3.22); (b) Hopf model (see Exercise 3.23). [Computer]

BEYOND THIS CHAPTER

While adaptive fourth-order Runge-Kutta is a good general-purpose algorithm, for some problems it is useful to employ more advanced methods. Specifically, if the solution is smooth and you want to minimize the number of evaluations of $f(x)$, you should consider trying Bulirsch-Stoer or predictor-corrector methods.[118] These are high-accuracy methods that, under the right conditions, allow you to use very large time steps. I especially recommend that you try Bulirsch-Stoer if your computational budget is limited and the routine is available in a library package.

On some problems you may find that the adaptive Runge-Kutta method demands an extremely small time step. For example, suppose that you wanted to simulate a pendulum consisting of bob of mass m at the end of a massless rod of stiffness k and rest length L (see Exercise 3.18). The period of oscillation for a simple pendulum is $T_p = 2\pi\sqrt{L_p/g}$, where $L_p \approx L$ is the length of the pendulum. The period of vibration for a spring is $T_s = 2\pi\sqrt{m/k}$. If the rod is very stiff (large k), then $T_s \ll T_p$. The time step will have to be less than the period of vibration so $\tau \ll T_p$. As you discover in Exercise 3.18, we may need to evaluate 10^4 time steps to simulate a single swing of the pendulum.

Systems of ordinary differential equations arising from physical problems with vastly different time scales, such as this spring-pendulum system, are said