Table 5.2 Determine the functional forms of $y(x)$ for the three sets of data. There are no measurement errors, but there are roundoff errors.

x	$y_1(x)$	$y_2(x)$	$y_3(x)$
0	0.00	0.00	2.00
0.5	0.75	1.59	5.44
1.0	3.00	2.00	14.78
1.5	6.75	2.29	40.17
2.0	12.00	2.52	109.20
2.5	18.75	2.71	296.83

planets and their periods, respectively. Setting the log scale option causes the `PlotFrame` to transform the data as it is being plotted and causes the axis to change how labels are rendered. Note that the plot automatically adjusts itself to fit the data because the autoscale option is true by default. Also the grid and the tick-labels change as the window is resized.

Listing 5.1 A simple program that producs a log-log plot to demonstrate Kepler's second law.

```
package org.opensourcephysics.sip.ch05;
import org.opensourcephysics.frames.PlotFrame;

public class SecondLawPlotApp {
   public static void main(String[] args) {
      PlotFrame frame = new PlotFrame("ln(a)", "ln(T)",
                              "Kepler's second law");
      frame.setLogScale(true, true);
      frame.setConnected(false);
      double[] period = {0.241, 0.615, 1.0, 1.88, 11.86,
           29.50, 84.0, 165, 248};
      double[] a = {0.387, 0.723, 1.0, 1.523, 5.202, 9.539,
           19.18, 30.06, 39.44};
      frame.append(0, a, period);
      frame.setVisible(true);
      // defines titles of table columns
      frame.setXYColumnNames(0, "T (years)", "a (AU)");
      // shows data table; can also be done from frame menu
      frame.showDataTable(true);
      frame.setDefaultCloseOperation(javax.swing.JFrame.EXIT_ON_CLOSE);
   }
}
```

Exercise 5.1 Simple functional forms

(a) Run `SecondLawPlotApp` and convince yourself that you understand the syntax.

(b) Modify `SecondLawPlotApp` so that the three sets of data shown in Table 5.2 are plotted. Generate linear, semilog, and log-log plots to determine the functional form of $y(x)$ that best fits each data set. ■

5.6 ■ SIMULATION OF THE ORBIT

We now develop a program to simulate the Earth's orbit about the Sun. The `PlanetApp` class shown in Listing 5.2 organizes the startup process and creates the visualization. Because this class extends `AbstractSimulation`, it is sufficient to know that the superclass invokes the `doStep` method periodically when the thread is running or once each time the Step button is clicked. The preferred scale and the aspect ratio for the plot frame are set in the constructor. The statement `frame.setSquareAspect(true)` ensures that a unit of distance will equal the same number of pixels in both the horizontal and vertical directions; the statement `planet.initialize(new double[]{x, vx, y, vy, 0})` in the `initialize` method is used to create an array on the fly as the argument to another method.

Listing 5.2 PlanetApp.

```
package org.opensourcephysics.sip.ch05;
import org.opensourcephysics.controls.*;
import org.opensourcephysics.frames.*;

public class PlanetApp extends AbstractSimulation {
   PlotFrame frame = new PlotFrame("x (AU)", "y (AU)",
                              "Planet Simulation");
   Planet planet = new Planet();

   public PlanetApp() {
      frame.addDrawable(planet);
      frame.setPreferredMinMax(-5, 5, -5, 5);
      frame.setSquareAspect(true);
   }

   public void doStep() {
      for(int i = 0;i<5;i++) { // do 5 steps between screen draws
         planet.doStep();      // advances time
      }
      frame.setMessage("t = "+decimalFormat.format(planet.state[4]));
   }

   public void initialize() {
      planet.odeSolver.setStepSize(control.getDouble("dt"));
      double x = control.getDouble("x");
      double vx = control.getDouble("vx");
      double y = control.getDouble("y");
      double vy = control.getDouble("vy");
      // create an array on the fly as the argument to another method
      planet.initialize(new double[] {x, vx, y, vy, 0});
      frame.setMessage("t = 0");
   }

   public void reset() {
      control.setValue("x", 1);
      control.setValue("vx", 0);
      control.setValue("y", 0);
      control.setValue("vy", 6.28);
      control.setValue("dt", 0.01);
      initialize();
   }
```