```
   double map(double r, double x) {
      return 4*r*x*(1-x); // iterate map
   }

   public static void main(String[] args) {
      CalculationControl.createApp(new IterateMapApp());
   }
}
```

Problem 6.1 The trajectory of the logistic map

(a) Explore the dynamical behavior of the logistic map in (6.5) with $r = 0.24$ for different values of x_0. Show numerically that $x = 0$ is a *stable fixed point* for this value of r. That is, the iterated values of x converge to $x = 0$ independently of the value of x_0. If x represents the population of insects, describe the qualitative behavior of the population.

(b) Explore the dynamical behavior of (6.5) for $r = 0.26, 0.5, 0.74$, and 0.748. A fixed point is *unstable* if for almost all values of x_0 near the fixed point, the trajectories diverge from it. Verify that $x = 0$ is an unstable fixed point for $r > 0.25$. Show that for the suggested values of r, the iterated values of x do not change after an initial *transient*; that is, the long time dynamical behavior is *period* 1. In Appendix 6A we show that for $r < 3/4$ and for x_0 in the interval $0 < x_0 < 1$, the trajectories approach the *stable attractor* at $x = 1 - 1/4r$. The set of initial points that iterate to the attractor is called the *basin* of the attractor. For the logistic map, the interval $0 < x < 1$ is the basin of attraction of the attractor $x = 1 - 1/4r$.

(c) Explore the dynamical properties of (6.5) for $r = 0.752, 0.76, 0.8$, and 0.862. For $r = 0.752$ and 0.862, approximately 1000 iterations are necessary to obtain convergent results. Show that if r is greater than 0.75, x oscillates between two values after an initial transient behavior. That is, instead of a stable cycle of period 1 corresponding to one fixed point, the system has a stable cycle of period 2. The value of r at which the single fixed point x^* splits or *bifurcates* into two values x_1^* and x_2^* is $r = b_1 = 3/4$. The pair of x values, x_1^* and x_2^*, form a *stable attractor* of period 2.

(d) What are the stable attractors of (6.5) for $r = 0.863$ and 0.88? What is the corresponding period? What are the stable attractors and corresponding periods for $r = 0.89$, 0.891, and 0.8922? ■

Another way to determine the behavior of (6.5) is to plot the values of x as a function of r (see Figure 6.2). The iterated values of x are plotted after the initial transient behavior is discarded. Such a plot is generated by `BifurcateApp`. For each value of r, the first `ntransient` values of x are computed but not plotted. Then the next `nplot` values of x are plotted with the first half in one color and the second half in another. This process is repeated for a new value of r until the desired range of r values is reached. The magnitude of `nplot` should be at least as large as the longest period that you wish to observe. `BifurcateApp` extends `AbstractSimulation` rather than `AbstractCalculation` because the calculations can be time consuming. For this reason you might want to stop the calculations before they are finished.

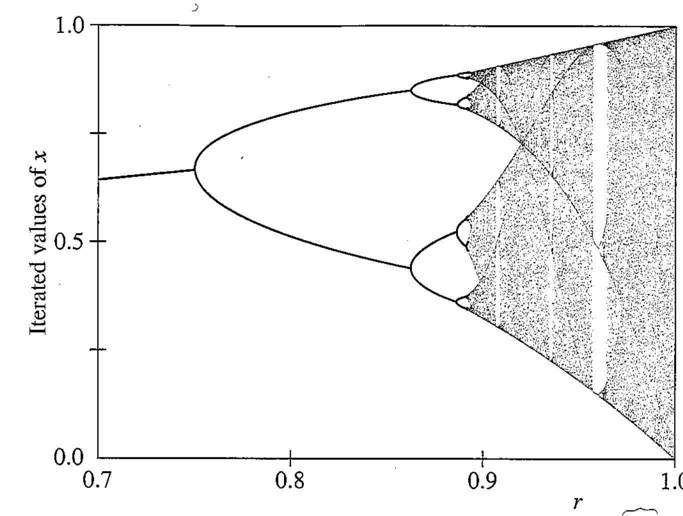

Figure 6.2 Bifurcation diagram of the logistic map. For each value of r, the iterated values of x_n are plotted after the first 1000 iterations are discarded. Note the transition from periodic to chaotic behavior and the narrow windows of periodic behavior within the region of chaos.

Listing 6.2 The `BifurcateApp` program generates a bifurcation plot of the logistic map.

```
package org.opensourcephysics.sip.ch06;
import org.opensourcephysics.controls.*;
import org.opensourcephysics.frames.*;

public class BifurcateApp extends AbstractSimulation {
   double r;         // control parameter
   double dr;        // incremental change of r, suggest dr <= 0.01
   int ntransient;   // number of iterations not plotted
   int nplot;        // number of iterations plotted
   PlotFrame plotFrame = new PlotFrame("r", "x",
        "Bifurcation diagram");

   public BifurcateApp() {
      // small marker size gives better resolution
      plotFrame.setMarkerSize(0, 0);
      plotFrame.setMarkerSize(1, 0);
   }

   public void initialize() {
      plotFrame.clearData();
      r = control.getDouble("initial r");
      dr = control.getDouble("dr");
      ntransient = control.getInt("ntransient");
      nplot = control.getInt("nplot");
   }

   public void doStep() {
      if(r<1.0) {
```