```
    void update(double time) {
        // set real and imaginary parts of wave function to zero
        System.arraycopy(zeroArray, 0, realPsi, 0, realPsi.length);
        System.arraycopy(zeroArray, 0, imagPsi, 0, imagPsi.length);
        for(int i = 0, nstates = realCoef.length;i<nstates;i++) {
            double[] phi = states[i];
            double re = realCoef[i];
            double im = imagCoef[i];
            double sin = Math.sin(time*eigenvalues[i]);
            double cos = Math.cos(time*eigenvalues[i]);
            for(int j = 1, n = phi.length-1;j<n;j++) {
                realPsi[j] += (re*cos-im*sin)*phi[j];
                imagPsi[j] += (im*cos+re*sin)*phi[j];
            }
        }
    }
}
```

The BoxSuperpositionApp class in Listing 16.6 implements the eigenstate superposition and displays the wave function by extending the AbstractAnimation class and implementing the doStep method.

Listing 16.6 BoxSuperpositionApp shows the evolution of a particle in a box.

```
package org.opensourcephysics.sip.ch16;
import org.opensourcephysics.controls.*;
import org.opensourcephysics.frames.ComplexPlotFrame;

public class BoxSuperpositionApp extends AbstractSimulation {
    ComplexPlotFrame psiFrame = new ComplexPlotFrame("x", "|Psi|",
        "Time dependent wave function");
    BoxSuperposition superposition;
    double time, dt;

    public BoxSuperpositionApp() {
        psiFrame.limitAutoscaleY(-1, 1);
    }

    public void initialize() {
        time = 0;
        psiFrame.setMessage("t = "+decimalFormat.format(time));
        dt = control.getDouble("dt");
        double[] re = (double[]) control.getObject("real coef");
        double[] im = (double[]) control.getObject("imag coef");
        int numberOfPoints = control.getInt("number of points");
        superposition = new BoxSuperposition(numberOfPoints, re, im);
        psiFrame.append(superposition.x, superposition.realPsi,
            superposition.imagPsi);
    }

    public void doStep() {
        time += dt;
        superposition.update(time);
        psiFrame.clearData();
        psiFrame.append(superposition.x, superposition.realPsi,
            superposition.imagPsi);
        psiFrame.setMessage("t = "+decimalFormat.format(time));
    }

    public void reset() {
        control.setValue("dt", 0.005);
        control.setValue("real coef", new double[] {0.707, 0, 0.707});
        control.setValue("imag coef", new double[] {0, 0, 0});
        control.setValue("number of points", 50);
        initialize();
    }

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


Because wave functions have real and imaginary components, BoxSuperpositionApp uses a ComplexPlotFrame for plotting. ComplexPlotFrame renders data using an envelope whose height is proportional to the magnitude, and the region between the envelope is colored from red to blue to show the phase. A more traditional plotting style showing the real and imaginary parts of the wave function is available from the frame's Tools menu. (Also see Appendix 16A.) We use BoxSuperpositionApp to study the periodicity of the wave function in Problems 16.10 and 16.11.

Problem 16.10 Time-dependent wave function for the infinite square well

(a) Add a second visualization to the BoxSuperpositionApp class that displays the probability density $\Psi(x,t)$.
(b) Change the coefficient array so that the particle is in the ground state. Show that the wave function changes in time, but that the probability density does not. At what times does the ground state wave function return to its initial condition? Find the corresponding times for the first and second excited states.
(c) Choose the coefficient array so that the particle is in a 50:50 superposition of the ground state and the first excited state. At what times does the wave function return to its initial condition? After what time does the probability density return to its initial condition?
(d) Change the coefficient array so that the particle is in a 50:50 superposition of the first and second excited states. After what time does the wave function return to its initial condition? After what time does the probability density return to its initial condition?
(e) Will the initial wave function always revive, that is, return to its initial condition? Explain. ■

Problem 16.11 Time-dependent wave function for the simple harmonic oscillator

(a) Modify BoxSuperpositionApp and BoxSuperposition to superimpose the eigenstates of the simple harmonic oscillator using the Eigenstate class to compute the eigenstates. What are the periods of the ground state and the first excited state wave functions? What are the periods for the probability densities?
(b) Change the coefficient array so that the particle is in a 50:50 superposition of the ground state and the first excited state. At what times does the wave function return