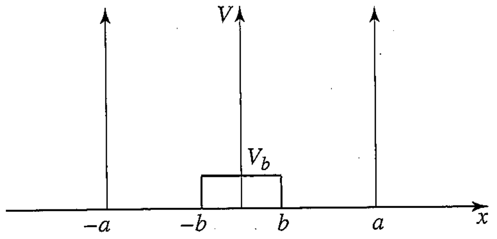

Figure 16.1 An infinite square well with a potential bump of height V_b in the middle.

either ϕ or ϕ' at $x = 0$. Hence, the parity of ϕ determines one of the boundary conditions. For simplicity, choose $\phi(0) = 1$ and $\phi'(0) = 0$ for even parity solutions and $\phi(0) = 0$ and $\phi'(0) = 1$ for odd parity solutions.

Problem 16.7 Symmetric potentials

(a) Modify `Schroedinger` to make use of symmetric potential boundary conditions for the harmonic oscillator:

$$V(x) = \frac{1}{2}x^2. \tag{16.18}$$

Start the solution at $x = 0$ using appropriate conditions for even and odd quantum numbers and find the first four energy eigenvalues such that the wave function approaches zero for large values of x. Because the computed $\phi(x)$ will diverge for sufficiently large x, we seek values of the energy such that a small decrease in E causes the wave function to diverge in one direction, and a small increase causes the wave function to diverge in the opposite direction. Initially choose `xmax` $= 5$ so that the classically forbidden region is sufficiently large so that $\phi(x)$ can decay to zero for the first few eigenstates. Increase `xmax` if necessary for the higher energy eigenvalues. Is there any pattern in the values of the energy eignevalues you found?

(b) Repeat part (a) for the linear potential $V(x) = |x|$. Describe the differences between your results for this potential and for the harmonic oscillator potential. The quantum mechanical treatment of the linear potential can be used to model the energy spectrum of a bound quark-antiquark system known as quarkonium.

(c) Obtain a numerical solution of the anharmonic oscillator $V(x) = \frac{1}{2}x^2 + bx^4$. In this case there are no analytic solutions, and numerical solutions are necessary for large values of b. How do the ground state energy and eigenstate depend on b for small b? ■

Problem 16.8 Finite square well

The finite square-well potential is given by

$$V(x) = \begin{cases} 0 & \text{for } |x| \le a \\ V_0 & \text{for } |x| > a. \end{cases} \tag{16.19}$$

The input parameters are the well depth V_0 and the half-width of the well a.

(a) Choose $V_0 = 10$ and $a = 1$. How do you expect the value of the ground state energy to compare to its corresponding value for the infinite square well? Compute the ground state eigenvalue and eigenstate by determining a value of E such that $\phi(x)$ has no nodes and is approximately zero for large x. (See Problem (16.7a) for the procedure for finding the eigenvalues.)

(b) Because the well depth is finite, $\phi(x)$ is nonzero in the classically forbidden region for which $E < V_0$ and $x > |a|$. Define the penetration distance as the distance from $x = a$ to a point where ϕ is $\sim 1/e \approx 0.37$ of its value at $x = a$. Determine the qualitative dependence of the penetration distance on the magnitude of V_0.

(c) What is the total number of bound excited states? Why is the total number of bound states finite? ■

As we have found, it is difficult to find bound state solutions of the time-independent Schrödinger equation because the exponential solution allows numerical errors to dominate when $V(x) - E > 0$ is large. Because we want to easily generate eigenstates in subsequent sections, we have written a general-purpose eigenstate solver that examines the maxima and minima of the solution as well as the nodes to determine the eigenstate's quantum number. The code for the `Eigenstate` class is in the ch16 package. The `EigenstateApp` target class shows how the `Eigenstate` class is used.

Listing 16.3 The `EigenstateApp` program tests the `Eigenstate` class.

```java
package org.opensourcephysics.sip.ch16;
import org.opensourcephysics.frames.PlotFrame;
import org.opensourcephysics.numerics.Function;

public class EigenstateApp {
   public static void main(String[] args) {
      PlotFrame drawingFrame =
           new PlotFrame("x", "|phi|", "eigenstate");
      int numberOfPoints = 300;
      double xmin = -5, xmax = +5;
      Eigenstate eigenstate =
           new Eigenstate(new Potential(), numberOfPoints, xmin, xmax);
      int n = 3;      // quantum number
      double[] phi = eigenstate.getEigenstate(n);
      double[] x = eigenstate.getXCoordinates();
      if(eigenstate.getErrorCode()==Eigenstate.NO_ERROR) {
         drawingFrame.setMessage("energy = "+eigenstate.energy);
      } else {
         drawingFrame.setMessage("eigenvalue did not converge");
      }
      drawingFrame.append(0, x, phi);
      drawingFrame.setVisible(true);
      drawingFrame.setDefaultCloseOperation(
           javax.swing.JFrame.EXIT_ON_CLOSE);
   }
}
class Potential implements Function {
   public double evaluate(double x) {
      return(x*x)/2;
   }
}
```