Visual Schrödinger: A Visualizer-Solver Applet for the One-Dimensional Schrödinger Equation

Kevin Schmidt
Department of Physics and Astronomy
Arizona State University
Tempe, Arizona 85287

Michael A. Lee
Department of Physics
Kent State University
Kent, Ohio 44242

September 27, 1998

This Web-based applet numerically solves the one-dimensional Schrödinger equation for a variety of standard Hamiltonians and permits users to define their own potential functions and rapidly display the results.

The applet has been designed primarily as a pedagogical tool. The first time user may go directly to the applet and, with the brief introduction, try a few typical problems.

After the initial orientation, investigation of more detailed example problems should prove illuminating. We include sections giving more documentation on advanced techniques, numerical methods and future direction, and discuss our design philosophy and goals.

This applet was prepared using freely available software tools. It is written in (almost) pure Java under Linux, with documentation written directly in latex and HTML. The applet is provided with a license that permits redistribution.

1

Contents

1 The Applet

The applet screen showing a solution to the double harmonic oscillator is pictured below with the caption that appears on the applet Web page.

The applet, shown below, has been initialized with the settings necessary to calculate the ground state of a one-dimensional harmonic oscillator. Click the Solve button and the plotting window will display the wave function in red and the potential in blue, with a green horizontal line at the energy value. (The first calculation may take longer than subsequent ones since the applet may need to download some Java class files.) The scale of the graph is set so that the maximum value of the unnormalized wave function is 1.0. The value of the energy and potential curves is obtained by multiplying the graph value by the V_{scale} value given in the left panel.
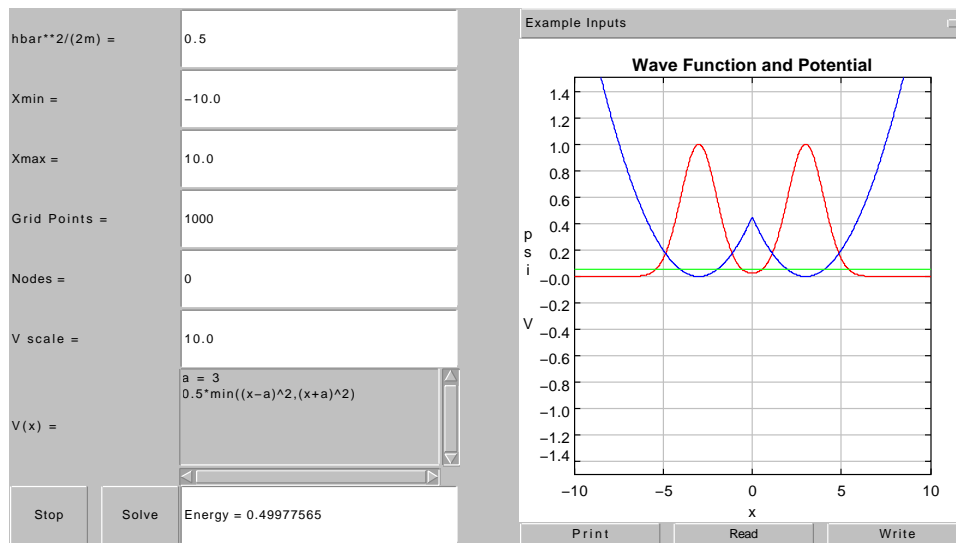
2 Documentation

The Schrödinger applet provides a numerical solution of the time-independent one-dimensional Schrödinger equation,

$$-\frac{\hbar^2}{2m}\frac{d^2}{dx^2}\psi(x) + V(x)\psi(x) = E\psi(x)\,. \tag{1}$$

The wave function is set to zero at the ends of the range to define the boundary conditions.

2.1 Panel Documentation

The problem to be solved is specified by the contents of the seven panel windows on the left side of the applet. Each can be changed by the user.

The input fields (from top to bottom) are:

- the input value of $\frac{\hbar^2}{2m}$.

- the minimum and maximum values of x which give the range of the wave function; the wave function is zero at $X_{\min}$ and $X_{\max}$.

- the number of grid points used by the differential equation integrator, uniformly distributed over the range.

- the number of nodes (zeros) in the solution, i.e. the excited state number.

- the scaling factor used to plot the potential and the energy.

- the analytic potential.

The analytic potential can be quite complicated or can be a simple expression such as 1/2*x**2. The rules for parsing (translating to computer code) the potential are given in Sec. 2.2.

The floating point numbers $\frac{\hbar^2}{2m}$, $X_{\min}$, $X_{\max}$, the energy scaling factor, and any numbers in the potential can be specified either in fixed point (e.g. 1.0 or 7.678, etc.) or by using scientific notation as in Fortran, (e.g. 1.0e-5, or 1.0e4, etc., where the integer after the e is the exponent).

While the user may specify any input field, the applet also provides an "Example Inputs" bar. Clicking the bar, located at the top of the right side,

will produce a menu of sample inputs to be selected. Choosing an item from the list will write a set of values in the input boxes that will allow you to calculate that system by clicking the Solve button. This list merely provides a survey of the types of problems which the user can examine and some examples for the potential parser. Go to Sec. 3 to explore the examples in detail.

Next to the Solve button is a button marked Stop normally this button will be disabled and will be "greyed out." When you click Solve, the Stop button will be enabled and by clicking it, you can stop the calculation. This can be useful if you have inadvertently typed in input data that will cause the integrator to run for a long time, e.g. by using an enormous number of grid points.

Below the plotting area are three buttons labeled Print, Read, and Write. The Print button will bring up a dialog that will allow you to print the current applet appearance. The preferred method of printing is to use the Browser's print function for those Browsers that support printing applets (e.g. Netscape 4.06). This applet Print button will allow printing on some other browsers, and when the applet is being run as a Java application or in a separate Window. The quality of the printing is platform dependent. Printing the applet in Landscape orientation normally provides the best quality.

The Read and Write buttons bring up file dialogs. Selecting Write and giving a file name will write a file with the values of the 7 input fields of the applet, the calculated energy, followed by the values of x, the wave function, and the potential. The file is a standard ASCII text file. Here is the start of a file generated for a double harmonic oscillator:

```
# 0.5  hbar^2/(2m)
# -10.0 xmin
# 10.0 xmax
# 1000 ngrid
# 0 nnode
# 0.1 vscale
# 0.49977564535774377  Energy
#a = 3
#0.5*min((x-a)^2,(x+a)^2)
-10.0 0.0 0.0
-9.97997997997998 6.367057436351622E-12 24.36006026046066
-9.95995995995996 1.2855968537246738E-11 24.220521322122924
```

```
-9.93993993993994 1.9589595863550545E-11 24.081383184986787
          .
          .
          .
```

The first 6 lines contain the first 6 input fields from the applet, the next line is the computed energy, followed by the potential definition. All of these lines begin with a # character. The remaining lines are x, $\psi(x)$, and $v(x)$, at the grid points. The plotting package gnuplot views the # character as a comment, so these files can be directly plotted using gnuplot. These files allow users to save cases of interest, with numerical tables of wave functions that can be used to calculate matrix elements etc. in other programs.

Clicking the Read button, and then selecting a previously saved file, will read the 7 input fields from the file. You need to click Solve to recalculate and display the results of this input.

These three buttons are the only parts of the applet that use Java 1.1 features and will require you to use one of the latest browsers. Early browsers such as Netscape 2.0 or 3.0 will run the rest of the applet, but will not allow printing or file manipulation. For more information, see section 2.7

2.2   Parser Function

The applet parser allows users to define their own potential functions for which the applet will solve the Schrödinger equation. By clicking in the panel window labeled $V(x) =$, the user can type one or more lines of assignment statements in the style of Fortran or C. For example, the parser can recognize expressions like the set

```
L=0
D0=3145
alpha=2.352709
r0=2.413683
ex=exp(-alpha*(x-r0))
D0*ex*(ex-2.)+.96415129*L*(L+1)/x^2
```

which defines a Morse potential for a pair of Cl atoms.

The operators that are recognized are $+$, $-$, $*$, $/$, $\hat{}$, and $**$. Both $\hat{}$ and $**$ can be used as the exponent operator. In addition, the functions sin, cos, exp, log, max, min, atan, sqrt, abs, H, and pulse are recognized.

The max, min and pulse functions have 2 arguments, and the rest take just one. H is the Heaviside function; H(y) is zero when y is negative and 1 when y is positive. The pulse(y,z) function is 1 when $|y| < |z|$ and zero otherwise.

Since the potential panel is editable, as are the other input panels, the arrow, insert, delete, home, end, etc. keys, as well as text selection with the mouse, provide simple editing features.

2.3   Plotter

The middle right of the applet panel is the plotting area which, upon successful completion of a solution, displays the wave function in red and the potential in blue with a green horizontal line at the energy value. (Note that the wave function is scaled so that its maximum magnitude is 1.0, and is *not* normalized so that its square integral is unity. Recall that the energy values shown need to be multiplied by the V_{scale} value that was typed in.)

The plotting routines are from ptplot by Edward A. Lee and Christopher Hylands, copyright University of California.

When the results have been displayed, the user can zoom, by using the mouse to click and pull down and to the right to produce a small square of the plot, which will be expanded into the full plotting region. You can do this repeatedly. This feature is extremely useful for viewing fine details of a wave function. You can restore the plot to its original form by clicking Solve again.

2.4   Natural Units

Natural units are exploited in several of the example inputs. One way to extend the range of numerical solutions is to use dimensional analysis. In the harmonic oscillator Hamiltonian,

$$H = -\frac{\hbar^2}{2m}\frac{d^2}{dx^2} + \frac{1}{2}m\omega^2 x^2 \tag{2}$$

we have three parameters, $\hbar$, m, and ω. Nature tells us the value of $\hbar$ in Joule-seconds. However, we aren't required to use SI as our measuring system. A more "natural" set of units can be found by looking at the dimensions of our three quantities: $\hbar$ has units of mass length^2 time^{-1}, m has units of mass,

7

and ω has units of time^{-1}. For any particular harmonic oscillator, we can choose our natural units to be

$$
\begin{aligned}
\text{Length Unit} &= \sqrt{\frac{\hbar}{m\omega}} \\
\text{Mass Unit} &= m \\
\text{Time Unit} &= \frac{1}{\omega}
\end{aligned}
\tag{3}
$$

The numerical values of $\hbar$, m and ω in natural units are all equal to 1 (m is 1 natural mass unit. ω is 1 natural inverse time unit, etc.). We can therefore do the calculations in these units, and at the end convert to physical units. For example, we will find the ground state energy of the harmonic oscillator is 0.5 in our natural units. A natural energy unit is

$$
\text{Energy Unit} = (\text{Mass Unit})(\text{Length Unit})^2(\text{Time Unit})^{-2} = \hbar\omega, \tag{4}
$$

so the value of the ground state energy is $0.5\hbar\omega$ as usual. In natural units, the Harmonic Oscillator Schrödinger Equation is

$$
\left[-\frac{1}{2}\frac{d^2}{dx^2} + \frac{1}{2}x^2 \right] \psi_n(x) = E_n\psi_n(x) \tag{5}
$$

which has no other free parameters. Therefore all harmonic oscillators are described by this equation.

2.5   Influence of Boundary Conditions

If you solve the above harmonic oscillator you will find that as you solve for higher and higher excited states, the wave function has a larger extent in x. For the first 20 or so states, you will find that you can enforce boundary conditions that $\psi_n(x) = 0$ at $x = \pm 10$ without changing these solutions. Note that this boundary condition introduces a new parameter, the distance where we set the wave function to zero. You will get different answers for different values if you increase n enough.

2.6   Numerical Errors

As described in section 2.8, the differential equation integration algorithm uses a finite difference formula for computing the wave function. If the interval is divided into N steps, the error in integration is of order N^{-5} if the

potential is well behaved, i.e., if the potential can be expanded in a Taylor series. Obviously discontinuous potentials such as finite square wells or steps will produce larger errors. The recommended way to estimate the step size error is to repeat the calculation with, for example, double and half the number of integration points to make sure that the values have converged to an acceptable error.

Similar tests can be done to make sure that the boundaries are not influencing the results for cases where the original problem's solution extends to infinity.

2.7   Error Messages and Failure Modes

The applet is relatively stable, but there are ways to break it.

- Parser Errors

  The parser, Sec. 2.2, accepts simple algebraic expressions and requires a fairly rigid syntax. If there is an input or parsing error, a message will be written to the energy panel.

- Failure to Find a Solution

  The numerical integration may fail because its searching procedure (integrating inward from the two end points) cannot match the the solutions at the midpoint. If the integration fails, an "Integration failed" message will be written to the energy panel. Generally this is caused by $X_{\min}$ and $X_{\max}$ chosen too far apart so that the wave function overflows or underflows during the calculation. It can also be caused by too few grid points. Adjusting the grid points or range specified for the solution usually fixes the problem unless a pathological potential is being solved.

- Java Errors

  Java is an evolving language. We have tested the applet using various versions of sun's appletviewer, Netscape and Internet Explorer. In addition the applet can be run as an application if you have a Java runtime environment installed. For full functionality, the applet requires Java version 1.1. The Print, Read, and Write buttons will not work under version 1.0, but the rest of the applet should. Browsers that

9

support the Java 1.1 features we use include Internet Explorer 4.0 and Netscape 4.06.

To reiterate, Browsers that don't or only partially support Java 1.1 may not recognize the print, read and write functions, but everything else will likely work fine.

2.8 Solution Algorithm

The applet solves the Schrödinger equation numerically on a discrete grid. The interval between $X_{\max}$ and $X_{\min}$ is divided into $N-1$ equal intervals of width $\Delta = (X_{\max} - X_{\min})/(N-1)$ and, as explained below, the Numerov method is used to convert the differential equation into a difference equation for the values of the wave function on each grid point.

The method begins by rewriting the Schrödinger equation as an expression for the second derivative in terms of the value of the wave function,

$$\psi''(x) = \frac{2m}{\hbar^2}\left[V(x) - E\right]\psi(x) \equiv \tilde{v}(x)\psi(x). \tag{6}$$

Here ψ'' is the second derivative of ψ, and the function $\tilde{v}(x)$ is defined to simplify notation. The Numerov method uses this form to integrate numerically the second derivative starting from the known value of Ψ at a boundary and using a guessed value of E.

The Numerov method is derived by writing the Taylor series expansions

$$\psi(x \pm \Delta) = \psi(x) \pm \Delta\psi'(x) + \frac{\Delta^2}{2}\psi''(x) \pm \frac{\Delta^3}{6}\psi'''(x) + \frac{\Delta^4}{24}\psi''''(x) \pm O(\Delta^5) \tag{7}$$

and

$$\psi''(x \pm \Delta) = \psi''(x) \pm \Delta\psi'''(x) + \frac{\Delta^2}{2}\psi''''(x) \pm O(\Delta^3). \tag{8}$$

The two equations defined by taking opposite signs in the second equation above can be added and solved for ψ'''' to give the usual second difference formula,

$$\psi''''(x) = \frac{\psi''(x + \Delta) - 2\psi''(x) + \psi''(x - \Delta)}{\Delta^2} + O(\Delta^2). \tag{9}$$

Notice that all the odd order terms in Δ cancel so the error is $O(\Delta^2)$. Adding the two equations defined by taking opposite signs in the first equation and

10

substituting for the fourth derivative gives the result

$$\psi(x+\Delta)+\psi(x-\Delta) = 2\psi(x)+\Delta^2\psi''(x)+\frac{\Delta^2}{12}\left[\psi''(x+\Delta) - 2\psi''(x) + \psi''(x-\Delta)\right].$$
(10)

We haven't used the differential equation yet. Substituting Eq. 6 for ψ'' gives the Numerov three term recursion relation

$$\psi(x+\Delta)[1-\frac{\Delta^2}{12}\tilde{v}(x+\Delta)]+\psi(x-\Delta)[1-\frac{\Delta^2}{12}\tilde{v}(x-\Delta)] = \psi(x)[2+\frac{10\Delta^2}{12}\tilde{v}(x)].$$
(11)

The error in this expression is $O(\Delta^6)$. By starting with the values of ψ at two adjacent grid points, with a guessed value of E, the Numerov recursion relation produces the wave function at all the grid points.

The code enforces the boundary conditions that $\psi = 0$ at $X_{\min}$ and $X_{\max}$. You can think of the system as being placed in an infinite square well potential that forces the wave function to zero at these points. The values are chosen by you. Other input that you need to provide are the value of $\frac{\hbar^2}{2m}$, the number of grid points which sets the step size Δ, the number of nodes (zero crossings) that you want in the wave function, which determines the corresponding eigenstate, and the potential.

Since $\psi = 0$ at the end points, the value at the neighbor grid point is arbitrary since it varies with the normalization. The code uses zero at the end points and an arbitrary small value at the neighboring points to start the Numerov recursion relation to generate the value of the wave function everywhere. The code integrates from $X_{\min}$ to a point near the center, and from $X_{\max}$ to just beyond the same point. The logarithmic derivative from the left and from the right at the match point are approximately calculated. These use the same Taylor series expressions above, but you subtract rather than add them to get a formula for the derivative. If the E value initially guessed is exactly right the two approximate derivatives from the left and right will be equal, and there will be exactly the correct number of nodes. If the energy is too high, there will either be too many nodes found, or the correct number of nodes and the difference of the approximate logarithmic derivatives will be positive. If the energy is too low, there will be either too few nodes, or the correct number of nodes and the difference in the approximate logarithmic derivatives will be negative. Therefore, we always know if the guessed energy is too high or too low. The integrator starts out with an energy of approximately the energy of an infinite square well from

$X_{\min}$ to $X_{\max}$ with the same number of nodes as the solution we want, and doubles this energy until it is too high. It then uses an energy that is the negative of the infinite square well energy and doubles it until it is too low. The actual starting values are unimportant as long as the order of magnitude is sensible. Once the correct energy is bracketed, a binary chop is used to home in on the correct energy. The average of the upper and lower bounds is used, and after integrating this value replaces either the old upper or lower bound. This method by itself gives 1 binary place per iteration, so typically around 50 iterations would give machine accuracy. To speed this up, when both the upper and lower bounds have the correct number of nodes, Newton's method is used to predict the new energy. Typically the energy converges within 10 or 20 iterations.

3   Prepared Examples

The example inputs for many textbook quantum problems are available by clicking the "Example Inputs" menu bar on the applet. In this section, each of the examples is described to provide a starting point to explore some solutions of Schrödinger's equation.

One way to use these is to click on the "Click Here to create a separate Solver Window" button below the applet, to get a solver Window that you can use while reading about the examples. An advantage of using the solver in a separate window is that it can be resized.

3.1   1-D Harmonic Oscillator

When first loaded or when "1-d Harmonic Oscillator" is selected, a simple harmonic oscillator is set up where natural units are chosen so that $\hbar = m = \omega = 1$, and the energies are in units of $\hbar\omega$. Clicking Solve will display the ground state energy and the wave function. Try changing the Nodes input variable to calculate higher excited states. For example, you might want to look at states with Nodes = 20 to see how the wave function looks for this excited state. You will notice that the green energy line is off scale unless you change the V_{scale} variable to something like 20. For Nodes greater than 40 or so, you will see that the wave function is being affected by the boundaries at ± 10. To calculate accurately for higher excited states, you will need to change these. You will also need to add more integration points by changing

Grid Points input variable to a larger value. Note that if you choose values for $X_{\min}$ and $X_{\max}$ too far apart, the integration may fail due to overflows or underflows. The applet does not do much error checking, does no rescaling, and does not use a variable step size. Our philosophy behind this omission is given in section 5.

3.2 L=1, Radial 3-d Harmonic Oscillator

The radial Schrödinger equation for central potentials has the same form as the one-dimensional Schrödinger equation. The boundary condition is that the radial wave function goes to zero at $r = 0$, so we need to choose $X_{\min}$ to be 0. However, the radial Schrödinger equation also contains the r^{-2} centripetal term that blows up at the origin. The usual technique is to expand the solutions at the origin. A sleazier technique, which we adopt, is to simply ignore the potential at the end points. Note the potential is given by

```
L=1
0.5*(x^2+ L*(L+1)/x^2)
```

By simply changing L=1 to L=0 or L=5, etc. you can change the orbital angular momentum of the solution. The potential area is scrollable and editable to allow these changes to be made easily. For large L values you will have to move $X_{\min}$ out from the origin to avoid overflows and underflows.

3.3 H Atom 1s State and H Atom 2p State

Choosing "H atom 1s State" sets up the usual hydrogen atom bound state problem. The units chosen for the Hydrogen atom are $\hbar = m = e = 1$ which give results in atomic units. Lengths are in Bohr radii (0.529 Angstrom), and energies are in Hartree = 2 Rydbergs ≈ 27.2 electron Volts.

The hydrogen atom states have a few extra problems that should be mentioned. For the ground state, the attractive coulomb potential cusp cannot be produced accurately near the origin with a fixed step size Numerov method and our simple boundary conditions. However, the results are probably good enough for our purposes. The default integration parameters given produce an energy that is good to 4 places for the ground state. The potential is given by

```
L=0
-1/x + L*(L+1)/(2*x^2)
```

for the s state with $X_{\max}$ given by 10 Bohr. The only difference in the p state parameters is that $L = 1$ is substituted for $L = 0$ in the potential, and the $X_{\max}$ value is extended to 30 Bohr. You can see the standard degeneracies by integrating the equations with the sum of L and Nodes constant.

3.4 Cl_2 Morse Potential

To demonstrate a more complicated potential, this example uses the Morse potential given in the introduction for 2 Chlorine atoms. This is a good approximation to the electronic ground state Born-Oppenheimer potential surface. The mass is the reduced mass of two Chlorine atoms. The units are energies in cm^{-1} and length in Angstroms. The L=0 state is given, but L is given as a potential parameter so it can be easily changed. You can expand the Morse potential about its minimum and see where it begins to deviate from the harmonic approximation, and calculate the vibrational and rotational spectrum of Cl_2.

3.5 Malfliet-Tjon Deuteron

Correct models of the deuteron require a tensor force. A central potential that gives some of the correct features is the Malfliet-Tjon potential. The units are length in Fermis (or femtometers) and energy in MeV. The calculated binding energy of the deuteron is 0.4 MeV which should be compared to the correct value of 2.2 MeV. The weak binding with most of the probability outside the attractive potential region is correct.

3.6 30th Excited of 0.5*x**4

This is given as another example of a simple one dimensional potential. You can compare the states of this potential to those of the harmonic oscillator.

3.7 The Double Harmonic Oscillator

The double harmonic oscillator can be studied using analytic methods (see E. Merzbacher, Quantum Mechanics), and is a favorite sort of problem from which to study perturbations. The potential is written as

```
a = 3
0.5*min((x-a)^2,(x+a)^2)
```

where 2a is the separation of the two wells and can be adjusted by simply typing in a new value. You can study the various states of the double well and look at the energy splittings as you increase the number of nodes and vary the spacing.

3.8   Square Well

This is simply a square well of depth 1 running from $x = -1$ to $x = 1$. It shows how to use the pulse function.

3.9   Two Narrow Pulses

While the applet doesn't support true delta functions, they can be approximated by narrow pulses. You should be careful that the grid is fine enough to achieve the desired accuracy. A favorite analytic problem is to calculate the bonding and antibonding (nodes = 0 and nodes = 1) states of two delta functions. This example allows us to solve and visualize an approximation to this problem. The variable a sets the spacing of the pulses, and w is their width.

3.10   "Atom" for Solid Slab and Solid Slab

These two examples can be used to look at what happens when we bring atoms together to make a solid. We choose for the "atom" a potential

```
pi=4.*atan(1.)
b=pi/2.
-10*pulse(x,b)*cos(x)^2
```

which looks complicated, but the pulse value is simply 0 for $x > |b|$ and 1 for $x < |b|$. This forms a cosine shaped attractive well. A large V_{scale} value is chosen to keep the plots from getting too crowded. By incrementing the Nodes value, you will find that this potential has 3 bound states.

Now select "Solid Slab" and the only change to the input data is that

```
b=21*pi/2
```

and the $X_{\min}$ and $X_{\max}$ values are changed to ± 40. Calculating, you will see that we now have 21 of these "atom" potentials brought close together. The wave function looks like a cosine modulating the "atomic" orbitals as expected from tight binding theory. You can increase the number of nodes and you will find the lowest 21 states give the beginnings of an energy band. The next 21 give a second band and the following 21 give a third band. You can calculate an approximate bandwidth from these values. You will also see that the higher energy states are more affected by the surface and tend to stick out beyond the slab as expected. If you increase the number of nodes to 63 you will see that the wave function is heavily damped inside the "solid." For these systems you may wish to use the zoom feature of the plots.

3.11   Infinite Well Super Symmetric Partner

Most quantum texts include a paragraph or two on the super symmetric partner potentials for one particle in one dimension. See for example, Quantum Mechanics, Richard Liboff, Introductory Quantum Mechanics, third edition, Addison Wesley, 1998, page 343. Write the Schrödinger equation as

$$\left\{ -\frac{\hbar^2}{2m} \left[\frac{d}{dx} + \ln(\psi_0)' \right] \left[\frac{d}{dx} - \ln(\psi_0)' \right] + E_0 \right\} \psi_n = E_n \psi_n, \qquad (12)$$

where $\ln(\psi_0)'$ is the logarithmic derivative of the ground state wave function. This is in the form

$$-\frac{\hbar^2}{2m} A^\dagger A \psi_n = (E_n - E_0)\psi_n \qquad (13)$$

Multiplying through by A gives the equation,

$$-\frac{\hbar^2}{2m} AA^\dagger (A\psi_n) = (E_n - E_0)(A\psi_n). \qquad (14)$$

which shows that $A\psi_n$ is the eigenvector of the Hamiltonian

$$H_S = -\frac{\hbar^2}{2m} AA^\dagger + E_0 \qquad (15)$$

with the same eigenvalues as the original Hamiltonian, with the exception that since $A\psi_0 = 0$, the eigenvalue E_0 does not occur. For the infinite square well running from -1 to 1, with $\hbar^2/(2m) = 0.5$, the potential for H_S is easily verified to be $(\pi/(2\cos(\pi x/2)))^2$. You can verify that the energy eigenvalues of this potential are $E_n = \pi^2(n+2)^2/8$ where n is the selected value of Nodes.

4   A Sample Problem and its Solution

Besides being able to investigate many solutions of the one-dimensional Schrödinger equation, we can also use the applet as a tool to allow students to get answers to more interesting problems.

A problem that we have given in our first year graduate quantum classes is the polarizability of a single hydrogen atom and the van der Waals interaction strength between two well separated hydrogen atoms. This requires the students to work out the solution in terms of matrix elements, to calculate those matrix elements numerically using wave functions produced by our applet, and to combine the results to get a value that can be compared to experiments and other calculations.

We also provide a sketch of our solution which students would receive after the completion of the assignment.

4.1   Hydrogen Atom Polarizability Problem

A hydrogen atom is in its ground state. The Hamiltonian of the hydrogen atom in atomic units $e = \hbar = m = 1$, and with the center of mass at the origin is

$$H = -\frac{1}{2}\nabla^2 - \frac{1}{r} \tag{16}$$

When a weak constant electric field along $\hat{z}$ of magnitude E is applied to this hydrogen atom, the perturbing potential is

$$V = zE, \tag{17}$$

and a dipole moment is induced proportional to the electric field. The proportionality constant is α, the polarizability.

Use perturbation theory to write an expression for the polarizability. Show that polarizability has units of volume, so the expression in atomic units needs to be multiplied by the Bohr radius cubed to convert to other units.

A direct expansion of the Schrödinger differential equation in powers of E, first done by I. Waller, Zeits. f. Physik **38**, 635 (1926) and P.S. Epstein, Phys. Rev. **28**, 695 (1926), gives an analytic value of the polarizability of 4.5 a.u.

Physically you should expect the weak electric field applied to the atom to distort the ground state wave function only slightly. However, usual perturbation theory uses the unperturbed atomic p states which have a much larger spatial extent than the 1s ground state. Convince yourself that enforcing the boundary condition that all wave functions go to zero at a radius $r = r_c$ will have no effect on the ground state with or without a weak field if r_c is large enough that the perturbed wave function has gone substantially to zero. By inspection, the value of the 1s state is less than 10^{-4} atomic units at $r = 10$ a.u. However, the p states are much more excited by being confined to this sphere of radius r_c. Because of this, these confined atomic orbitals have been named "fireballs" by Otto Sankey who has used them to do electronic structure calculations in solids (see for example, O.F. Sankey and Niklewski, Phys. Rev. **B40**, 3979 (1989)). Use the Schrödinger applet to calculate numerically the energies and wave functions of the first ten unnormalized p fireball radial functions, and save the results to 10 files. Do the angular integrations analytically, and write a computer program to calculate the matrix elements

$$\langle np_z|z|1s\rangle \tag{18}$$

by doing the radial integrations numerically. Trapezoidal rule integration is adequate. Show that you get the correct polarizability to 3 places (about all that can be expected from our numerics) with only 6 or 7 fireball states.

4.2   Hydrogen Atom van der Waals Interaction

Now assume that you have two well-separated ground-state hydrogen atoms with atomic states that do not overlap. You can assume the protons are infinitely massive with fixed positions in space. The vector from proton 2 to proton 1 defines the z axis, and the separation distance is R. Take the electron coordinates $\vec{r}_1$ and $\vec{r}_2$ as the vectors pointing from proton 1 to electron 1 and from proton 2 to electron 2 respectively as in figure 1. Show that ignoring the coulomb interactions between the particles in different hydrogen atoms gives the unperturbed Hamiltonian

$$H = -\frac{1}{2}\nabla_1^2 - \frac{1}{r_1} - \frac{1}{2}\nabla_2^2 - \frac{1}{r_2} \tag{19}$$

Figure 1: Geometry for the van der Waals problem
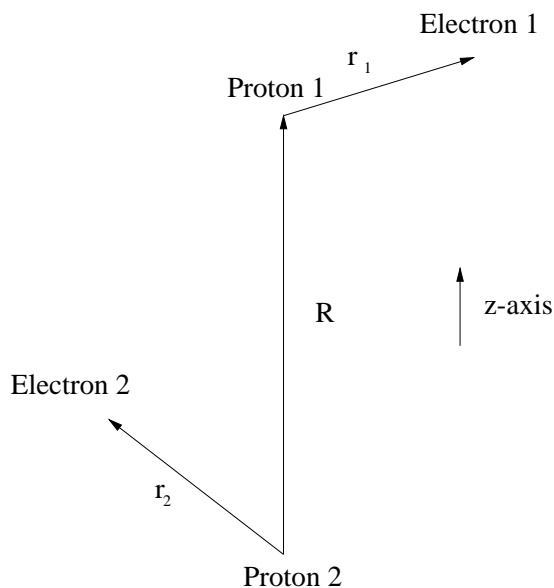
and if R is large compared to the atomic size, the remaining coulomb interactions can be approximated by the dipole-dipole interaction

$$V = \frac{x_1 x_2 + y_1 y_2 - 2 z_1 z_2}{R^3}. \tag{20}$$

Use your same 10 matrix elements to calculate the van der Waals interaction between two Hydrogen atoms given by the second order perturbed energy. Compare your numerical result with the result of L. Pauling and J.Y. Beach, Phys. Rev. **47**, 686 (1935), who calculated using a variational method,

$$V_{vdW}(r) = -\frac{6.499}{R^6}, \tag{21}$$

in atomic units.

4.3 Solution for Hydrogen Polarizabilty and van der Waals Strength

The perturbed wave function is

$$|\psi\rangle = |1s\rangle + \sum_{n\neq 1s} \frac{|n\rangle\langle n|V|1s\rangle}{E_{1s} - E_n} \qquad (22)$$

and the dipole moment is

$$-\frac{\langle\psi|z|\psi\rangle}{\langle\psi|\psi\rangle}. \qquad (23)$$

Keeping terms linear in V gives the linear response result

$$\alpha = -2 \sum_{n\neq 1s} \frac{|\langle n|z|1s\rangle|^2}{E_{1s} - E_n}. \qquad (24)$$

The z operator only couples the $1s$ state to p states. If the usual H atom states are used, the continuum states also must be included by taking the appropriate limit of the sum to integrate over these continuum states.

The angular integrations can be done directly, using for example,

$$z = r\sqrt{\frac{4\pi}{3}}Y_{10}(\theta,\phi) \qquad (25)$$

to give

$$\langle np_z|z|1s\rangle = \frac{1}{\sqrt{3}}\int_0^\infty dr\ r^3 R_{1s}(r)R_{np}(r) \qquad (26)$$

The fireballs radial functions were calculated using the applet with the input given by the H-atom 2p state example input except we change $X_{\max}$ to 10.0 and increment nodes from 0 to 9. The output was saved into the files 2p.out, 3p.out, ... 11p.out by using the Write button. The ground state energy is -0.5 a.u., and the following C code was used to calculate the matrix elements.

```
#include <stdio.h>
#include <math.h>
#define NBUF 1024

main()
```

```
{
    char s[NBUF];
    double r,r2,psi,psi0,anorm0,anorm,dip,e;
    int i,j,c;
    anorm = anorm0 = dip = 0.0;
    i = j = 0;
    while ((c = getchar()) != EOF) {
        if (c == '\n') {
            i = 0;
            if (j++ == 6) sscanf(s,"# %lg",&e);
            if (s[0] == '#') continue;
/*
 read r and psi, calculate the normalizations and the dipole transition
 matrix elements to the 1s ground state using trapezoidal rule
*/
            sscanf(s,"%lg %lg",&r,&psi);
            psi0 = exp(-r);
            r2 = r*r;
            anorm0 += r2*psi0*psi0;
            anorm += psi*psi;
            dip += r2*psi*psi0;
        } else {
            s[i++] = (char) c;
        }
    }
    dip /= sqrt(anorm0*anorm*3.0);
    printf("%lg %lg\n",dip,e);
}
```

The matrix elements and energy denominators were combined in the following C code (which also calculates the van der Waals strength, see below):

```
#include <stdio.h>
#define N 10

main()
{
    double dipmat[N],e[N],d,vdw;
    int i,j;
```

```
/*
 read in matrix elements and energies
*/
   for (i=0;i<N;i++) scanf("%lg %lg\n",dipmat+i,e+i);
/*
 calculate polarizability
*/
   printf("State       Summed Polarizability\n");
   d = 0.0;
   for (i=0;i<N;i++) {
      d += 2.*dipmat[i]*dipmat[i]/(0.5+e[i]);
      printf("%d              %4.4lg\n",i,d);
   }
/*
 calculate van der Waals strength
*/
   vdw = 0.0;
   for (i=0;i<N;i++) {
      for (j=0;j<N;j++) {
         vdw += 6.*dipmat[i]*dipmat[i]*dipmat[j]*dipmat[j]/(1.0+e[i]+e[j]);
      }
   }
   printf("Van der Waals Strength = %4.4lg\n",vdw);
}
```

and combining these together using the unix script (dip is the matrix element executable, and polar is the polarizability and van der Waals strength executable).

```
#!/bin/sh
(for i in 2p.out 3p.out 4p.out 5p.out 6p.out 7p.out 8p.out 9p.out 10p.out\
   11p.out
   do
   dip <$i
   done) | polar
```

gives the result

```
State        Summed Polarizability
0               3.391
1               4.243
2               4.442
3               4.484
4               4.494
5               4.497
6               4.498
7               4.498
8               4.498
9               4.498
Van der Waals Strength = 6.497
```

which demonstrates that we only miss the correct result by .002. Interested students can play around to see what needs to be done to improve this result.

The van der Waals interaction goes through in a similar way. Taylor series expanding the coulomb interactions gives the dipole-dipole term at lowest order. The second order energy is then given by

$$
\begin{aligned}
\Delta E &= \frac{1}{R^6} \sum_{n,m \neq 1s,1s} \frac{|\langle nm|x_1 x_2 + y_1 y_2 - 2z_1 z_2|1s1s\rangle|^2}{2E_{1s} - E_n - E_m} \\
&= \frac{6}{R^6} \sum_{n,m \neq 1s,1s} \frac{|\langle nm|z_1 z_2|1s1s\rangle|^2}{2E_{1s} - E_n - E_m} \\
&= \frac{6}{R^6} \sum_{n,m} \frac{|\langle np|z|1s\rangle\langle mp|z|1s\rangle|^2}{2E_{1s} - E_n - E_m}
\end{aligned}
\tag{27}
$$

The numerical result is given above and agrees with Pauling and Beach to about .002 atomic units.

For students interested in continuing, the strength of the Axilrod-Teller (see B.M. Axilrod and E. Teller, J. Chem. Phys. **11**, 293 (1943)) triple-dipole three-body potential between three well separated hydrogen atoms can be calculated using these same techniques and matrix elements.

5   Goals and Pedagogy

Our goal is to enhance the teaching of physics not to replace instructors. While we have given examples of some possible systems to explore, there are

obviously many others. We hope that individual students and instructors will use our integrator applet to study systems that we haven't thought about or that are too complex to include here.

Examples of what we mean are:

- A study of perturbation theory could compare the correct answers to perturbative results for systems beyond the usual hydrogen atom, harmonic oscillator,etc.

- By adding an additional potential to our solid slab example, the wave functions and energy levels of impurity states can be studied.

- The relationship of quantum and semiclassical methods can be directly calculated and a detailed comparison can be made.

- More advanced uses might include studying partial wave expansions in scattering theory. Other numerical techniques are usually better suited for calculating scattering properties, but if the outer boundary, where the wave function is set to zero is R_{max}, and the energy computed for the radial wave function with angular momentum l is E, then the phase shift δ_l at that energy is

$$\tan(\delta_l) = \frac{j_l(kR_{max})}{n_l(kR_{max})} \tag{28}$$

  where k is defined as

$$E = \frac{\hbar^2 k^2}{2m} \tag{29}$$

  and the j_l and n_l are spherical Bessel functions.

- The results of the applet can be used to give students numerical problems such as the van der Waals strength problem described in section 4 to allow students to develop their own programming skills

We have taken a middle road in producing a robust application. The applet could be much more robust in treating arbitrary input. For example, the current applet fails if the potential is so large in certain region that the wave function integrator underflows or overflows in that region. Our code could easily detect when this is about to happen and rescale the wave function in that region. Similarly, adaptive step size integrators would allow students

to select an acceptable error (or we could select one for them), and they would not need to enter the number of integrator grid points. We could also make the integrator accurately include delta function and discontinuous potentials. This sort of robustness is desirable if the goal is to hide the numerics in a black box. One of our goals in our courses is to give our students a feel for how to solve numerical problems, and the kind of numerical errors that they will encounter as research physicists. We believe that an applet such as ours which is designed to fail gracefully (i.e. it shouldn't crash) but which can fail, when asked to do a numerically difficult problem is a better pedagogical tool for advanced students.

Finally, why did we write an applet in Java? The original version of this integrator was written in C using lex and yacc for the parser generator, and the user interface used the X Windowing system. The main problem with this approach was that students needed to go to one of the physics computer rooms where they could run the code and see the results. Few students had computers where they could run let alone compile the codes. With our students having a mix of Windows, MacIntosh and Unix computers, the only truly common platform is the Web browser. The fact that both Web browsers and the Java development kits can be obtained free makes them ideal for pedagogical use. The performance of Java is far below Fortran in most of our tests, so we don't see it as a viable language for serious computing, but the platform independence and price makes it no contest for our educational codes. By making the applet available on our class web server, many more students routinely use it to check their work and study new problems.

6   License

Our HTML version uses a Java code AppletButton.java, which is Copyright ©1995, 1996 Sun Microsystems, Inc. See the copyright notice in that file for its allowed use.

As mentioned above, the plotting routines are from ptplot by Edward A. Lee and Christopher Hylands, copyright University of California.

Everything thing else including the applet and documentation are Copyright ©1998, Kevin Schmidt and Michael A. Lee, all rights reserved, and distributed under the terms of the GNU public license.

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free

Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Index