

REVIEW: The Matching Method Algorithm

Need for a more general method

The *shooting method* for solving the time-independent Schrödinger equation is limited to potentials which have even parity, such as the square well potential. The parity condition enables us to use the $x = 0$ behavior of the wave-function for assigning the two initial values of $\psi(x)$ necessary to begin the iteration in position. However, not all potentials have a specific parity. This includes the Lennard-Jones potential which we used in the previous Chapter

$$V(x) = 4\epsilon \left[\left(\frac{\sigma}{x} \right)^{12} - \left(\frac{\sigma}{x} \right)^6 \right]$$

While this potential does have a pocket with a minimum at $r = (\sigma/2)^{-1/6}$, the potential is not symmetric about that minimum value. We cannot use the trigger method to solve for the eigenstates of this potential. Instead we introduce a new algorithm called the *matching method*.

The matching method for solving the Schrödinger Equation

The *matching method* for solving the Schrödinger Equation makes use of the same iteration equation as the shooting method

$$\psi_{j+1} \approx 2\psi_j - \psi_{j-1} - 2(\Delta x)^2(E - V_j)\psi_j$$

The new feature of the matching method is that the iteration is done twice starting at opposite ends, left and right, of the confining potential. At these two ends the eigenvalue energy is assumed to be below the potential energy. So the particle is bound and has near 0 value for its wave function at these two opposite ends. The slope is also assumed to be near 0 at these two ends.

The dual iteration produces a left side solution ψ_L , and a right side solution ψ_R . It is a bit like constructing a tunnel under a mountain by drilling in from opposite sides, and meeting in the middle somewhere. Now just like in a tunnel construction, the two sides of the tunnel must be at a common point, that is the same (x, y, z) point¹, and also have the slopes of the tunnel be the same at this point. In the matching method, the match position x_M can be anywhere. Typically one places x_M somewhere in the region of the minimum of the potential, but the exact point is generally not critical (with one exception) as we will verify. We can easily get the heights of the two wave-functions to be the same by using an arbitrary normalization constant. The Schrödinger Equation is linear in ψ , so any multiple of ψ is also a solution.

Getting the slopes of the two solutions to agree at the match point, after they have been normalized to the same value, is expressed mathematically as

$$\frac{d\psi_L}{dx} \Big|_{x=x_M} = \frac{d\psi_R}{dx} \Big|_{x=x_M} \quad \text{and} \quad \psi_L(x_M) = \psi_R(x_M)$$

The slope matching can only be achieved if we have a correct eigenvalue energy E . Otherwise the ψ_L and the ψ_R will go wandering in different directions, as shown in Figure 10.7 on page 316. Hence we need to home-in on the correct energy value, similar to how we homed-in using the trigger method.

¹The tunnel construction is a 3D problem, but the one-dimensional Schrödinger equation solution is essentially a 2D problem where the y coordinate represents the amplitude of the wave-function.

REVIEW: The Matching Method Algorithm

Initial values and the energy convergence procedure

For the matching method algorithm we assign initial values as follows

$$\psi_L(x_L) = 0 \quad \psi_L(x_L + \Delta x) = 0.0001\Delta x$$

$$\psi_R(x_R) = 0 \quad \psi_R(x_R - \Delta x) = 0.0001\Delta x$$

The x_L and the x_R are the left and right side endpoints. The second condition on each line above makes the slope very near, but not quite, 0.0. If one put the slopes exactly at 0, then the iteration equation would give a trivial $\psi(x) = 0$ at all x values.

The energy convergence procedure relies on looking differences in the derivatives of the wave functions at the match point x_M .

$$D(E) = \frac{d\psi_L}{dx} \Big|_{x=x_M} - \frac{d\psi_R}{dx} \Big|_{x=x_M}$$

If from one energy guess to the next, with E changing by an increment ΔE between the guesses, it is observed that this difference $D(E)$ changes sign, then the energy change ΔE is halved and reversed in sign. If not, then the energy is changed again by the original amount ΔE in the same direction. Energy convergence is assumed to have been achieved if the derivatives are predicted to match within some small difference, say $D(E) \approx 1\%$ of the average of the two derivatives.

Example matching method program *oneDimensionalQM.cpp*

An example matching program *oneDimensionalQM.cpp* is being provided in this class. The was built from the previous *squareWell.cpp* program, with the main modifications related to computing two iterated solutions ψ_L and ψ_R and changing the homing-in procedure for the energy convergence as indicated above.

The “help” function for the program shows the following options (revised from previous lecture)

Optional input arguments (defaults in parentheses)

eGuess (= -1.5), *eDelta* (= 0.25), *vType* (= 1), *vScale* (= 10.),

derivativeDifferenceCheck = (0.01), *psiParity* (= +1), *xDelta* (= 0.001)

As you can see, there is an initial input guess energy E and ΔE value. The default initial guess is negative, because the default potential is the Lennard-Jones potential. The program will also work with the harmonic oscillator potential $V = Kx^2/2$ or with the square well V_0 potential, or with the power law potential $V = x^n$, or with the anharmonic potential $V = (K_1x^2 + K_2x^4)/2$. The choice of *vType* = 2 to *vType* = 5 give the different potentials, in the order just listed.

You can see the inputs for the derivative difference check which determines when the energy convergence has been attained. The *vScale* value corresponds to the ϵ parameter in the Lennard-Jones potential, or the K_2 parameter in the harmonic oscillator potential, or the V_0 of the square well potential, or the n exponent for the power law, or the K_1 for the anharmonic potential with K_2 taken as $K_1/10$.

There is also the *psiParity* and the Δx options as previously available in the *squareWell* program.

The Matching Method Algorithm with the Numerov iteration

The Numerov iteration method

The time independent Schrödinger equation, with conserved eigenvalue energy E is

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

We have been solving this differential equation for particular potential functions $V(x)$ with the use of an iteration equation based on a second-order finite differences approximation to the equation. The second order based iteration equation is

$$\psi_{j+1} \approx 2\psi_j - \psi_{j-1} - 2(\Delta x)^2(E - V_j)\psi_j$$

where we have set $\hbar = 1$ and the mass $m = 1$.

Fortunately, the linear, second order time-independent Schrödinger equation does not contain a first derivative. As such, it can be solved most accurately by what is called the *Numerov method* of iteration. The Numerov method is accurate to fifth order in the Δx scale, even more accurate than the fourth order Runge-Kutta method that we were using earlier this semester for mechanics problems.²

The general, linear second order differential equation for a function $y(x)$ in one dimension has the form

$$\frac{d^2y}{dx^2} + k^2(x)y = 0$$

Here the $k^2(x)$ is some real function of x . For the Schrödinger equation, the $k^2(x)$ is given by

$$k^2(x) = \frac{2m}{\hbar^2}(E - V(x))$$

After going through the necessary algebra, one arrives at the following Numerov iteration equation for y_{j+1} in terms of the two previous values y_j and y_{j-1}

$$y_{j+1} = \frac{2 \left[1 - \frac{5}{12}(\Delta x)^2 k^2(x_j) \right] y_j - \left[1 + \frac{1}{12}(\Delta x)^2 k^2(x_{j-1}) \right] y_{j-1}}{1 + \frac{1}{12}(\Delta x)^2 k^2(x_{j+1})}$$

The much more accurate Numerov iteration equation has been made the default choice in the class `oneDimensionalQM.cpp` program. You can switch back to the second order iteration method by given a negative value for Δx , which is then converted to the usual positive Δx for the program's use.

²A postscript file containing notes on the derivation of the Numerov method has been placed on the class web site. This set of notes has a mistake in using $E + V(x)$ in the definition of $k^2(x)$ instead of $E - V(x)$. From the Schrödinger equation, you can easily see that the $E - V(x)$ must be correct, just as it was used in the second order iteration method.

Solutions to the Time Independent Schrödinger Equation

Description of Problems

We will use the *oneDimensionalQM.cpp* program in class to calculate the ground state solution for the Lennard-Jones potential, as illustrated in Figure 10.8 on page 317, which has $E_G \approx 1.969$. After that we will use the program to complete Exercise 10.2, finding the first few states of the harmonic oscillator potential, Next we will complete Exercise 10.6 finding several levels of the Lennard-Jones potential.

The (optional) homework assignment was to do Exercises 10.3, 10.4, and 10.6, which involved some modification of the *oneDimensionalQM.cpp* program to handle two more potential energy functions: $V(x) = V_0x^n$ and $V(x) = (K_1x^2 + K_2x^4)/2.0$. We will also revisit the square well solution with the *oneDimensionalQM.cpp* program.

The Lennard-Jones Potential with $\epsilon = 10$

The Lennard-Jones potential has the form

$$V(x) = 4\epsilon \left[\left(\frac{\sigma}{x} \right)^{12} - \left(\frac{\sigma}{x} \right)^6 \right]$$

This potential is the default choice (*vType=1*) for the *oneDimensionalQM.cpp* program. The ground state wave-function, using a value $\epsilon = 10$, had an energy $E = -1.88$, compared to the -1.97 being quoted by the textbook.

The new version of the program draws in the energy level, rescaled to be on the potential curve, between the two classical turning points of the actual potential. Between these classical turning points, the mass has a real kinetic energy. Outside of the classical turning points, the potential energy is larger than the total energy meaning the classically forbidden regions. In those regions the wave-function must be exponentially decaying.

An important feature of the ground state wave-function is that it does not cross the x -axis. At the two endpoints the wave-function is exponentially falling to the x -axis. A place where the wave-function would cross the x -axis is called a *node*. So the ground state wave-function is characterized by having zero nodes, but all excited state wave-functions have increasingly more nodes, as we will see.

The Lennard-Jones Potential with $\epsilon = 30$

We can repeat the Lennard-Jones potential calculation with a larger depth, $\epsilon = 30$. The ground state wave function has a correspondingly lower energy, $E_{GS} = -13.27$, again with zero nodes. However, if we choose the negative parity option in the input command line

```
oneDimensionalQM -1.8 0.25 1 30. 0.01 -1
```

we will discover a barely bound first excited state at $E_1 = -0.59$, with one node. Although a negative parity option was invoked, you can see there is no actual negative parity in the wave-function, meaning $\psi(x) = -\psi(-x)$. We don't even draw that negative side for the Lennard-Jones potential. Instead, the negative parity option works in the program to have the second point of the ψ_L solution be below the x axis, while the next-to-last point of the ψ_R solution is above the x -axis. This is the way an actual negative parity solution will behave, for example $\sin(x)$. We look next at the square well potential to see this effect better.

Solutions to the Time Independent Schrödinger Equation

The Square Well Potential

We have already solved the square well potential using the *shooting algorithm*. This square well program made use of the parity feature of the potential function in order to get the starting points for the iteration equation. However, we can just as well use the *matching algorithm* method. We will first look at the case of

`oneDimensionalQM 2.0 0.25 3 500. 0.01 +1`

This is a square well of height 500, and its ground state solution has $E_{GS} = 1.16$. You can barely see the level drawn in just above the x axis. Next, we look at the case

`oneDimensionalQM 2.0 0.25 3 500. 0.01 -1`

Now the solution shows one node, with an energy $E_1 = 4.66$. It is less bound than the ground state because it is a little closer to the top of the barrier. It clearly has negative parity³, just as the ground state solution had positive parity.

For the third example of the square well potential we take the input

`oneDimensionalQM 8.0 0.25 3 500. 0.01 +1`

The result is a positive parity state with energy $E_2 = 10.47$ and having two nodes, one more than the previous lower energy negative parity state. Finally, we look at the case

`oneDimensionalQM 2.0 0.25 3 100000. 0.01`

The ground state energy now is computed as $E_{GS} = 1.234$, quite close to the $E_{GS} = \pi^2/8 \approx 1.233$ infinite square well prediction. Moreover, wave-function outside the potential wall $x = \pm 1$ is effectively 0.0 all the way out to the limits ($x = \pm 2$). You will recall that we had a diverging solution from the square well program for such a high value of V_0 .

Harmonic Oscillator Potential

The well-known harmonic oscillator potential has $V(x) = Kx^2/2$ where the upper-case K is the spring constant, not the wave number which is lower-case k . The energy values for the harmonic oscillator wave-functions are given by $E_n = (n + \frac{1}{2})\hbar\omega$ with $\omega \equiv \sqrt{K/m}$. In our reduced units, the eigenvalues become $E_n = (n + 1/2)\sqrt{K}$. For example, choosing to have $K = 3$ for the input

`oneDimensionalQM 1.0 0.25 2 3. 0.01 +1`

produces a ground state wave function with $E_{GS} = 0.865967$, which is extremely close to the analytic value of $E_{GS} = \sqrt{3}/2 = 0.866025$. The first excited state, having negative parity, is computed to have an energy $E_1 = 2.5977$, again very close to the analytic result of 2.5981. The program input to generate this state numerically is

`oneDimensionalQM 1.0 0.25 2 3. 0.01 -1`

³For the `oneDimensionalQM.cpp` program I originally chose the match point $x_M = 0$ for both the square well and the harmonic oscillator potentials. Can you figure out why that was not a good idea for the negative parity cases? You can see from the `oneDimensionalInitial.ps` figure that the match point is $x_M = 0.5$ for the square well potential. You should be able to confirm that the actual value of the x_M match point doesn't much matter, except that $x_M = 0$ is a poor general choice.

Solutions to the Time Independent Schrödinger Equation

The Power Law Potential

The power law potential has $V = V_0 x^n$ where n could be any exponent. For $n = 2$ the power law is equivalent to a harmonic oscillator potential with $V_0 = K/2$. The power law choice for the *oneDimensionalQM* program is obtained with `vType = 4`. You can run the example

oneDimensionalQM 1.0 0.25 4 2

which is the same as the harmonic oscillator with $K = 2$. The computed energy for this case is $E_{GS} = 0.70703$, compared with the analytic value $E_{GS} = \sqrt{2}/2 = 0.70711$.

By the way, you can check how good are the $\psi(x)$ functions for the harmonic oscillator by recalling that the analytic harmonic oscillator eigenfunctions are related to the Hermite polynomials H_n as

$$\psi_n(x) = \sqrt{\frac{1}{2^n n!}} \left[\frac{m\omega}{\pi\hbar} \right]^{\frac{1}{2}} \exp\left(-\frac{m\omega x^2}{2\hbar}\right) H_n\left(x\sqrt{\frac{m\omega}{\hbar}}\right)$$

The first few Hermite polynomials are

$$H_0 = 1$$

$$H_1(z) = 2z$$

$$H_2(z) = 4z^2 - 2$$

$$H_3(z) = 8z^3 - 12z$$

$$H_4(z) = 16z^4 - 48z^2 + 12$$

$$H_5(z) = 32z^5 - 160z^3 + 120z$$

Higher order polynomials can be generated from the recursion relation

$$H_{n+1}(z) = 2zH_n(z) - 2nH_{n-1}(z)$$

Anharmonic Oscillator Potential

The anharmonic oscillator potential adds a perturbation term to the simple harmonic oscillator potential. The result is

$$V(x) = \frac{1}{2}(K_1 x^2 + K_2 x^3)$$

For example, one could have $K_1 = 1$ and $K_2 = 0.1$ (in appropriate units), as in Figure 10.13 on page 321. With the input command

oneDimensionalQM 1.0 0.25 5 1 0.01

you will get the even parity ground state wave function with $E_{GS} = 0.5327$, which is slightly higher than the harmonic oscillator ground state $E_{GS} = 0.5$ value. With the input command

oneDimensionalQM 8.0 0.25 5 1 0.01

you will see a six-node wave-function having $E_6 = 8.477$. The equivalent six-node wave-function for the harmonic oscillator, with the spring constant $K_2 = 1$, appears at $E_6 = 6.5$, according to the analytic $E_n = (n + 0.5)$ formula in reduced units.