

The Time-Dependent Schrödinger Equation

Physics 3300, Weber State University, Spring Semester, 2012

In this project you will create a simulation to predict the time evolution of a quantum particle in one dimension. You will do this by directly solving the time-dependent Schrödinger equation (TDSE). This partial differential equation plays the same role in quantum mechanics that Newton's second law plays in classical mechanics: Given the initial state of the system, you can solve this differential equation to predict the state at any future time.

Quantum Mechanics

The state of a quantum particle in one dimension is described by its wavefunction, $\psi(x, t)$. In general, this function is complex, with “real” and “imaginary” parts (neither of which is any more real or imaginary than the other):

$$\psi(x, t) = \psi_R(x, t) + i\psi_I(x, t). \quad (1)$$

Basically, you can just think of ψ as a two-component object. The symbol i gives us a way to keep the two components distinct, and the algebraic rule $i^2 = -1$ lets us hide some subtle mathematical operations in common-looking notation.

To interpret the wavefunction, we say that the probability of finding the particle at position x (if, hypothetically, you were to measure its position) is proportional to the square modulus of the wavefunction:

$$\text{Probability density} = |\psi(x, t)|^2 = \psi_R^2 + \psi_I^2. \quad (2)$$

Notice that this is the same formula we would use to compute the square magnitude of a vector in two dimensions.

Given any initial wavefunction $\psi(x, 0)$, we can find the wavefunction at later times by solving the TDSE:

$$i\hbar \frac{\partial \psi}{\partial t} = H\psi = \left[-\frac{\hbar^2}{2m} \frac{\partial^2}{\partial x^2} + V(x) \right] \psi. \quad (3)$$

Here H is the Hamiltonian operator or total-energy operator. For a nonrelativistic particle in one dimension, this operator is given by the expression in brackets, where the first term gives the kinetic energy (related to the curvature of the wavefunction) and $V(x)$ is the potential energy associated with whatever forces are acting on the particle. Again, this equation plays the same role in quantum mechanics that Newton's second law plays in classical mechanics, governing the change in the system's state over time.

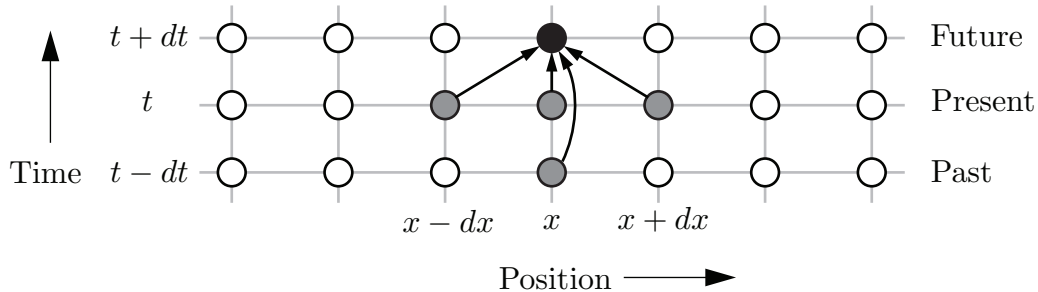
Despite its fundamental role in quantum mechanics, the TDSE doesn't get much attention in introductory quantum mechanics courses. Instead it gets overshadowed by the time-*independent* Schrödinger equation (TISE), whose solutions give the allowed energy levels of a system and the corresponding definite-energy wavefunctions. In principle, once you know these solutions, you can use them to go back and solve the TDSE if necessary. But in some situations, it's actually easier to solve the TDSE directly. That's what you'll do in this project.

Discretizing the TDSE

To solve the TDSE on a computer, you'll need to treat both space and time as discrete, rather than continuous. The derivatives of ψ can then be expressed in terms of the values at neighboring space/time locations. For example,

$$\frac{\partial\psi}{\partial t} \approx \frac{\psi(x, t + dt) - \psi(x, t - dt)}{2 dt}. \quad (4)$$

You can similarly work out $\partial^2\psi/\partial x^2$ as in the previous project, expressing it in terms of $\psi(x, t)$, $\psi(x + dx, t)$, and $\psi(x - dx, t)$. The TDSE, therefore, allows you to solve for $\psi(x, t + dt)$ (the *future* wavefunction *here*) in terms of the *present* wavefunction here and one step to either side, and the *past* wavefunction here. You can visualize it all by plotting space horizontally and time vertically:



Now that you have the general idea, get a clean sheet of paper and use the discretized TDSE to work out the explicit formula for $\psi(x, t + dt)$. Then write each ψ in terms of its real and imaginary parts, and group the real and imaginary terms to obtain *separate* formulas for $\psi_R(x, t + dt)$ and $\psi_I(x, t + dt)$. To keep the formulas reasonably simple (here and in your code), use units in which \hbar , m , and dx are all equal to 1.

Your code, then, will need to use *six* arrays (each with index x) to store the real and imaginary parts of ψ at the past, present, and future times. Each step of the simulation will use your formulas to calculate the “future” ψ_R and ψ_I from the past and present ones, then get ready for the next step by copying present to past and future to present.

Program Design and Testing

Call your program `TDSE.java`. Extend the `Canvas` class to plot the wavefunction with x along the horizontal axis, using a scale of one pixel for each x value. Your total array size (and `Canvas` width) should be 800 or more. Make your wavefunction arrays one element larger than the number of pixels, with the first and last array elements always held at zero; this is equivalent to putting an infinite potential energy barrier at each edge of the space.

A good choice for the initial wavefunction is a *Gaussian wavepacket*,

$$\psi(x, 0) = e^{-((x-x_0)/a)^2} \cdot e^{ipx}, \quad (5)$$

where the first factor (called the *envelope*) localizes the wavefunction within a limited region (centered on x_0 with a half-width of approximately a), while the second factor gives it an average momentum of p (in units where $\hbar = 1$). Make the width large enough to enclose several oscillations of the wavefunction, but keep the momentum small enough so the wavelength of the oscillations is at least several units (pixels). There's no need to "normalize" the wavefunction.

Your `paint` method should plot the real and imaginary parts of the (present) wavefunction, using different colors, each as a series of connected line segments. Multiply the wavefunction by an arbitrary scale factor so it looks good when plotted. Once you've written code to do all these things, test it to be sure that both your initialization code and your `paint` method are working correctly. Then add an option to plot the probability density instead of (or in addition to) the real and imaginary parts of ψ . You may also want to try coloring the area under the probability density graph using hues that correspond to the phase angle of ψ in the complex plane.

You're now ready to write the simulation code that will move the wavefunction forward in time. Set up a separate thread to handle these calculations, and be careful typing in the formulas that you wrote out earlier. Create an array to hold the potential energy values, but keep them all set to zero for now. Use a time step of 0.25 in natural units. To get the simulation started, you'll also need to initialize the *past* wavefunction. You can do this with a nearly identical version of the discretized TDSE, but with equation (4) replaced by the less accurate first-order approximation

$$\frac{\partial\psi}{\partial t} \approx \frac{\psi(x, t) - \psi(x, t - dt)}{dt}. \quad (6)$$

Once you have everything working, you should find that the wavepacket moves to the right (for positive p), spreads out, and bounces off the edges of the space, producing intricate interference patterns. Describe some of the details of this behavior in your lab report.

Before adding potential energy to the system, you should do some tests for accuracy and stability. Check that the center of the wavepacket really moves at the expected speed, for two or more different values of the momentum. Add a numerical readout for the total area under the probability density graph, and check that this quantity (which represents the total probability of finding the particle somewhere) is unchanged over time. Finally, experiment with larger time steps and note when the simulation is stable and unstable.

Barrier Scattering

At this point you could set up any potential energy function you like, and use your simulation to predict the corresponding wavefunction evolution. Feel free to invent your own experiments! In addition to any other scenarios, though, you should explore the scattering of a wavepacket from a barrier, as follows.

Set the potential energy to zero everywhere except inside a narrow region near the middle of your space, where the potential energy should have some nonzero constant value. If this value is positive, this region is a “barrier”; if the value is negative, it’s a “well”. Be sure that your initial wavepacket is entirely to the left of the barrier (or well). The idea is to aim the wavepacket at the barrier (or well), let it interact once, and measure the fractions that are reflected and transmitted.

You’ll want to create GUI controls that let you adjust the width and “height” of the barrier, as well as the wavefunction momentum (or kinetic energy, which is $p^2/2$). Also have the program add up the total area under the probability density graph on each side of the barrier, and use these to calculate and display the fraction on either side. A good experimental procedure is to hold the wavefunction energy and barrier height fixed, then measure the transmitted percentage as a function of barrier width. Take plenty of data and plot it in a spreadsheet. Try this for several values of the wavepacket and barrier energies. Note that in classical mechanics, a particle would always penetrate a barrier whose energy is less than the particle’s, while it would never penetrate a barrier whose energy is greater. These two cases are qualitatively different in quantum mechanics as well, but only in extreme circumstances will you find that the transmission probability is 100% or 0%. Be sure to spend plenty of time looking for patterns in your data. Feel free to consult a quantum mechanics textbook if you wish, but remember that your wavepacket’s energy is somewhat fuzzy (due to the uncertainty principle), unlike the plane-wave states usually treated in textbooks. Make sure your lab report documents the patterns you’ve discovered. Do your best to explain the physical reason for these patterns.

In summary, you should turn in a working TDSE simulation with GUI controls and data readouts for doing barrier scattering experiments. Your lab report should document the formulas that the simulation uses, the challenges that you encoun-

tered while writing your code, the results of your diagnostic tests and your barrier scattering experiments, and anything else you tried along the way.