

Solving the Time-Dependent Schrödinger Equation^a

David G. Robertson[†]

Department of Physics, Otterbein University, Westerville, OH 43081

(Dated: October 10, 2011)

Abstract

Methods for solving the time-dependent Schrödinger equation in one dimension are discussed. Possible simulation projects include the study of scattering by barriers and wells, the analysis of time evolution by expansion in energy eigenstates, and tests of time-dependent perturbation theory.

Keywords: quantum mechanics, Schrödinger equation, time dependence, scattering, perturbation theory, Crank-Nicholson method

^a Work supported by the National Science Foundation under grant CCLI DUE 0618252.

[†] Email: drobertson@otterbein.edu

CONTENTS

I. Module Overview	2
II. What You Will Need	2
III. Quantum Mechanics Background	3
IV. Solving the Time-Dependent Schrödinger Equation	5
V. Simulation Projects	12
References	16
Glossary	18

I. MODULE OVERVIEW

This module discusses methods for solving the time-dependent Schrödinger equation in one space dimension. This is the central problem in quantum mechanics, and courses in quantum physics typically devote considerable time to developing solutions in analytically tractable cases (e.g., square wells, the harmonic oscillator). The ability to determine the wavefunction numerically allows exploration of much wider class of potentials, however, as well as the study of e.g., scattering with actual wavepackets rather than plane waves. One can also test the predictions of time-dependent perturbation theory. Such exercises allow the development of deeper intuition regarding the properties of quantum systems.

II. WHAT YOU WILL NEED

The minimal physics background required includes quantum mechanics at the level of a typical sophomore-level course on modern physics, specifically the basic properties of wavefunctions and the Schrödinger equation. Students will ideally have worked through the free particle in one dimension and be familiar with wave packets, dispersion, and so on. An upper-level course in quantum mechanics will allow a richer exploration of problems with the tools developed here, for example the study of time-dependent perturbations.

On the mathematical side, facility with differential and integral calculus is essential, as is a basic familiarity with differential equations and complex numbers. Students who have passed through a sophomore-level modern physics course and the standard calculus sequence through the sophomore year should have the necessary background.

The computing resources needed are actually fairly minimal, as the “size” of the computational problem is modest. The natural framework for scientific computing is a high-level language like Fortran, C or C++, and this is an excellent project for students to sharpen their programming skills. However, the needed calculations can be done using Matlab or Mathematica. Support for complex arithmetic is essential.

Some facility for generating plots will also be necessary, for example `gnuplot` or (less ideally) Excel. Some suggestions in this regard are given with the simulation projects.

III. QUANTUM MECHANICS BACKGROUND

The central problem of quantum mechanics is to solve the Schrödinger equation, which determines the time evolutions of the wavefunction for a system [1]. For a single non-relativistic particle in one dimension this takes the form

$$i\hbar \frac{\partial \Psi(x, t)}{\partial t} = -\frac{\hbar^2}{2m} \frac{\partial^2 \Psi(x, t)}{\partial x^2} + V(x, t)\Psi(x, t), \quad (3.1)$$

where m is the particle mass and $V(x, t)$ is the potential energy. Given an initial wavefunction $\Psi(x, 0)$, the Schrödinger equation determines Ψ for all later (and earlier) times.

The physical meaning of Ψ is that it gives the probability density for finding the particle at different locations, if its position is measured [3]. Specifically, the probability dP to find the particle in a small range from x to $x + dx$ at time t is given by

$$dP = |\Psi(x, t)|^2 dx. \quad (3.2)$$

The probability to find the particle in some finite interval, say between $x = a$ and $x = b$, is then the sum of the probabilities for each infinitesimal interval between a and b :

$$P = \int_a^b |\Psi(x, t)|^2 dx. \quad (3.3)$$

A critical requirement for the consistency of the framework is that the total probability to find the particle *somewhere* be unity; hence we must require that

$$\int_{-\infty}^{\infty} |\Psi(x, t)|^2 dx = 1. \quad (3.4)$$

Wavefunctions that satisfy this condition are said to be “normalized.” An important feature of the time evolution defined by the Schrödinger equation is that if the wavefunction is normalized at one time, then it will be normalized for all other times as well. This property is known as “unitarity.”

The physical importance of the normalization requirement cannot be over-emphasized; it is this condition that puts the “quantum” in quantum mechanics. Wavefunctions that do not satisfy this fall into two classes: functions for which the normalization integral is finite but not equal to one, and functions for which the normalization integral is infinite. Given a function in the first class, we can easily produce a normalized wavefunction. Assume that we have found a solution of the Schrödinger equation Ψ for which

$$\int_{-\infty}^{\infty} |\Psi(x, t)|^2 dx = A, \quad (3.5)$$

with A a finite number. Then the rescaled wavefunction

$$\Psi'(x, t) = \frac{1}{\sqrt{A}} \Psi(x, t) \quad (3.6)$$

will be properly normalized, i.e., will satisfy

$$\int_{-\infty}^{\infty} |\Psi'(x, t)|^2 dx = 1. \quad (3.7)$$

Note that because the Schrödinger equation is *linear* in the wavefunction, the rescaled wavefunction remains a solution.

Functions for which the normalization integral is infinite, on the other hand, cannot be normalized at all. Such wavefunctions are simply unacceptable for describing real physical systems, and must be discarded despite being solutions to the Schrödinger equation [4].

The problem of finding solutions to the Schrödinger equation is usually approached using the technique of *separation of variables*. This leads to the “time independent” Schrödinger equation, which determines the energy eigenstates $\psi(x)$:

$$-\frac{\hbar^2}{2m} \frac{d^2\psi(x)}{dx^2} + V(x)\psi(x) = E\psi(x), \quad (3.8)$$

Here E is a constant to be determined by solving the equation [5]. It has the form of an eigenvalue equation,

$$\hat{H}\psi = E\psi, \quad (3.9)$$

where \hat{H} is the Hamiltonian operator

$$\hat{H} = -\frac{\hbar^2}{2m} \frac{d^2}{dx^2} + V(x) \quad (3.10)$$

and E is the eigenvalue. It should be emphasized that the TISE determines both ψ and E , that is, we find that each solution ψ works only with some particular value of E . We can label the solutions by an index n , so that each function ψ_n has a corresponding eigenvalue E_n . We must also be sure that our wavefunctions are normalizable; this means that ψ_n themselves must be normalizable. It is convenient to require that

$$\int_{-\infty}^{\infty} |\psi_n(x)|^2 dx = 1. \quad (3.11)$$

The solution to the Schrödinger equation can then be obtained by expanding the initial wavefunction in terms of the stationary states,

$$\Psi(x, 0) = \sum_n c_n \psi_n(x) \quad (3.12)$$

for some coefficients c_n . The full solution is then

$$\Psi(x, t) = \sum_n c_n \psi_n(x) e^{-iE_n t/\hbar} \quad (3.13)$$

An alternative to this procedure is to develop a numerical approach to solving the Schrödinger equation directly, without expanding in stationary states. It is to this task that we now turn.

IV. SOLVING THE TIME-DEPENDENT SCHRÖDINGER EQUATION

In this section we will develop techniques for solving the full Schrödinger equation numerically. The first step is to introduce a grid of space points, separated by some distance Δx , on which we will determine the wavefunction. We will also discretize in time, that is, evaluate the wavefunction only for a discrete set of times separated by some Δt . Hence we replace the function $\Psi(x, t)$ with the discrete values Ψ_i^n , where i labels the space point and n labels the time value. Specifically, the grid points are located at

$$x_i = x_{min} + i\Delta x \quad (4.1)$$

and the discrete times are given by (taking the initial time to be $t = 0$)

$$t_n = n\Delta t. \quad (4.2)$$

The next step is to approximate the derivatives in the Schrödinger equation as differences. These should approximate the corresponding derivatives well as long as Δx and Δt are “small enough.” More precisely, the change in Ψ from one grid point to the next or one time step to the next should be small compared to Ψ itself.

Now, an obvious approximation to the time derivative is

$$\frac{\partial\Psi(x, t)}{dt} \approx \frac{\Psi_i^{n+1} - \Psi_i^n}{\Delta t} \quad (4.3)$$

which becomes exact as $\Delta t \rightarrow 0$. To develop a discrete approximation to the second derivative with respect to x , imagine that we Taylor expand $\Psi(x, t)$ about the point x (I will suppress the dependence on t here for clarity):

$$\Psi(x + \Delta x) = \Psi(x) + \Delta x \frac{\partial\Psi}{\partial x} + \frac{1}{2}\Delta x^2 \frac{\partial^2\Psi}{\partial x^2} + \dots \quad (4.4)$$

where all derivatives are evaluated at the point x . Eq. (4.4) implies that

$$\Psi(x + \Delta x) + \Psi(x - \Delta x) = 2\Psi(x) + \Delta x^2 \frac{\partial^2\Psi}{\partial x^2} + \mathcal{O}(\Delta x^4). \quad (4.5)$$

If we now drop the higher order terms as unimportant for sufficiently small Δx , then this can be interpreted as an approximate formula for the second derivative:

$$\frac{\partial^2\Psi}{\partial x^2} \approx \frac{\Psi(x + \Delta x) + \Psi(x - \Delta x) - 2\Psi(x)}{\Delta x^2}. \quad (4.6)$$

This difference formula is analogous to, e.g.,

$$\frac{\partial\Psi}{\partial x} \approx \frac{\Psi(x + \Delta x) - \Psi(x)}{\Delta x} \quad (4.7)$$

which gives a discrete approximation to the first derivative. Actually, it is more analogous to the “symmetric” first derivative:

$$\frac{\partial\Psi}{\partial x} \approx \frac{\Psi(x + \Delta x) - \Psi(x - \Delta x)}{2\Delta x}, \quad (4.8)$$

showing that there is actually quite a bit of freedom in how these approximations are constructed. Both eqs. (4.7) and (4.8) approach the exact derivative as $\Delta x \rightarrow 0$ and so either is

a perfectly valid approximation for finite Δx . The latter is often preferred, however, because it is more accurate for a given value of Δx .

Switching to the grid notation where $\Psi(x_i \pm \Delta x, t) = \Psi_{i\pm 1}^n$, the discretized version of the Schrödinger equation thus takes the form

$$i\hbar \left(\frac{\Psi_i^{n+1} - \Psi_i^n}{\Delta t} \right) = -\frac{\hbar^2}{2m} \left(\frac{\Psi_{i+1}^n - 2\Psi_i^n + \Psi_{i-1}^n}{\Delta x^2} \right) + V_i^n \Psi_i^n \quad (4.9)$$

Now, imagine that we know the wavefunction at any one time; eq. (4.9) could then be used to evolve it forward to the next time step. One has only to solve for Ψ_i^{n+1} , which will be given in terms of Ψ at the current time step and the potential function. The new Ψ then becomes the basis for a further step, and so on. Hence given some initial wavefunction we can calculate Ψ at any later (or earlier) time. Note that in general the value of Ψ^{n+1} at some grid point will depend on the value of Ψ at that point and its nearest neighbors – this is the effect of the space derivatives in the Hamiltonian.

Unfortunately, this approach is totally useless because it is numerically unstable [2]. The basic reason for this can be seen as follows. Let us adopt a matrix notation and write eq. (4.9) as

$$i\hbar \left(\frac{\Psi^{n+1} - \Psi^n}{\Delta t} \right) = \hat{H} \Psi^n \quad (4.10)$$

Here Ψ^n represents a vector of values (across the space grid) and \hat{H} is a (hermitian) matrix defined by the requirement that it reproduce the right-hand side of eq. (4.9). Solving for Ψ^{n+1} then gives

$$\Psi^{n+1} = \left(1 - \frac{i\Delta t}{\hbar} \hat{H} \right) \Psi^n \quad (4.11)$$

Now, imagine that we expand Ψ^n in terms of the eigenvectors of \hat{H} , that is the vectors φ_α satisfying:

$$\hat{H} \varphi_\alpha = \epsilon_\alpha \varphi_\alpha \quad (4.12)$$

where the eigenvalues ϵ_α are real (since \hat{H} is hermitian). We write

$$\Psi^n = \sum_\alpha c_\alpha^n \varphi_\alpha \quad (4.13)$$

and in the exact solution to the Schrödinger equation we would have

$$\Psi^{n+1} = \sum_\alpha c_\alpha^n \varphi_\alpha e^{-i\epsilon_\alpha \Delta t / \hbar} \quad (4.14)$$

The approximate solution eq. (4.11) amounts to replacing the exponential by the first-order expression

$$e^{-i\epsilon_\alpha \Delta t / \hbar} \approx 1 - \frac{i\epsilon_\alpha \Delta t}{\hbar} \quad (4.15)$$

But this is a complex number with magnitude greater than one:

$$\left| 1 - \frac{i\epsilon_\alpha \Delta t}{\hbar} \right| = \left(1 + \frac{\epsilon_\alpha^2 \Delta t^2}{\hbar^2} \right)^{1/2} \geq 1 \quad (4.16)$$

Hence the solution will be unstable; all modes grow without bound as we iterate. In fact it is “unconditionally unstable” – it is unstable no matter how small we make Δt . It is just doomed.

To fix this we need a new discrete equation. Fortunately we have a great deal of freedom in this regard – the main restriction is that it must reduce to the correct Schrödinger equation in the limits $\Delta t \rightarrow 0$ and $\Delta x \rightarrow 0$. For example, we could replace the right hand side of eq. (4.9) with its value at the future time step, that is,

$$i\hbar \left(\frac{\Psi_i^{n+1} - \Psi_i^n}{\Delta t} \right) = -\frac{\hbar^2}{2m} \left(\frac{\Psi_{i+1}^{n+1} - 2\Psi_i^{n+1} + \Psi_{i-1}^{n+1}}{\Delta x^2} \right) + V_i^{n+1} \Psi_i^{n+1} \quad (4.17)$$

or, in matrix notation,

$$i\hbar \left(\frac{\Psi^{n+1} - \Psi^n}{\Delta t} \right) = \hat{H} \Psi^{n+1} \quad (4.18)$$

This clearly has the same continuum limit as the original equation – it is essentially what we obtain by taking the “backwards” difference for the time derivative. It is also implicit, in the sense that Ψ^{n+1} appears on both sides of the equation, and we must solve this system of linear equations to determine it. Still, this could be done (details to follow) so this is another possible discretization we can use.

But does it help? Well, yes and no. The evolution is now stable; solving for Ψ^{n+1} we find

$$\Psi^{n+1} = \frac{1}{1 + \frac{i\Delta t}{\hbar} \hat{H}} \Psi^n \quad (4.19)$$

Now the energy eigenstates are multiplied at each time step by a complex number of magnitude

$$\left| \frac{1}{1 + \frac{i\epsilon_\alpha \Delta t}{\hbar}} \right| = \left(\frac{1}{1 + \frac{\epsilon_\alpha^2 \Delta t^2}{\hbar^2}} \right)^{1/2} \quad (4.20)$$

which is less than one. However, the “time evolution operator” in eq. (4.19) – the matrix that multiplies Ψ^n to give Ψ^{n+1} – is not unitary! (Recall that \hat{H} is a hermitian matrix.) This

means that the norm of the wavefunction will not be constant; in fact it will decrease monotonically with time. Since conservation of probability is a central requirement of quantum theory, we should not use *this* discretization either.

However, a useful approach is now close at hand. What if we instead replace the right hand side of eq. (4.9) with the average of those terms at timestep n and timestep $n + 1$:

$$i\hbar \left(\frac{\Psi^{n+1} - \Psi^n}{\Delta t} \right) = \frac{1}{2} \left(\hat{H}\Psi^{n+1} + \hat{H}\Psi^n \right) \quad (4.21)$$

This clearly leads to the same continuum limit ($\Delta x \rightarrow 0$ and $\Delta t \rightarrow 0$), and so is a valid approximation. It leads to the solution

$$\Psi^{n+1} = \left(\frac{1}{1 + \frac{i\Delta t}{2\hbar} \hat{H}} \right) \left(1 - \frac{i\Delta t}{2\hbar} \hat{H} \right) \Psi^n \quad (4.22)$$

Now the evolution operator is clearly unitary, so that probability will be conserved in this discrete evolution. In addition, the complex number multiplying each eigenmode has exactly unit modulus, so the evolution should be stable. This version of the discretization is known as the Crank-Nicholson scheme, and is the one we shall adopt.

Let us write it out in detail so that we can see the computational problem(s) to be solved. We begin by putting eq. (4.22) in the form

$$\left(1 + \frac{i\Delta t}{2\hbar} \hat{H} \right) \Psi^{n+1} = \left(1 - \frac{i\Delta t}{2\hbar} \hat{H} \right) \Psi^n \quad (4.23)$$

Now recall that \hat{H} acts according to the difference formula we derived earlier; that is

$$\hat{H} = -\frac{\hbar^2}{2m} \frac{\partial^2}{\partial x^2} + V(x) \quad (4.24)$$

with

$$\frac{\partial^2 \Psi}{\partial x^2} \rightarrow \frac{\Psi_{i+1} - 2\Psi_i + \Psi_{i-1}}{\Delta x^2} \quad (4.25)$$

With this substitution and some algebra, you can convince yourself that eq. (4.23) can be written in the form

$$\Psi_{i+1}^{n+1} + \Psi_{i-1}^{n+1} + A_i \Psi_i^{n+1} = B_i \quad (4.26)$$

where

$$A_i = -2 + \frac{4im\Delta x^2}{\hbar\Delta t} - \frac{2m\Delta x^2}{\hbar^2} V_i^{n+1} \quad (4.27)$$

$$B_i = -\Psi_{i+1}^n - \Psi_{i-1}^n + \Psi_i^n \left(2 + \frac{4im\Delta x^2}{\hbar\Delta t} + \frac{2m\Delta x^2}{\hbar^2} V_i^n \right) \quad (4.28)$$

Eq. (4.26) is a set of linear equations that determine Ψ^{n+1} in terms of Ψ^n .

To solve these, observe that the system is tri-diagonal, meaning that if we write it as a matrix equation $M\Psi^{n+1} = B$:

$$\begin{pmatrix} A_1 & 1 & 0 & 0 & 0 & \dots \\ 1 & A_2 & 1 & 0 & 0 & \dots \\ 0 & 1 & A_3 & 1 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \begin{pmatrix} \Psi_1^{n+1} \\ \Psi_2^{n+1} \\ \Psi_3^{n+1} \\ \vdots \end{pmatrix} = \begin{pmatrix} B_1 \\ B_2 \\ B_3 \\ \vdots \end{pmatrix} \quad (4.29)$$

then the only nonzero elements in M are on the central diagonal and the two neighboring ones. This reflects that the derivatives couple grid points to their nearest neighbors only. This allows a straightforward solution by row reduction, as follows.

Let us first convert eq. (4.29) to the form

$$\begin{pmatrix} 1 & U_1 & 0 & 0 & 0 & \dots \\ 0 & 1 & U_2 & 0 & 0 & \dots \\ 0 & 0 & 1 & U_3 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \begin{pmatrix} \Psi_1^{n+1} \\ \Psi_2^{n+1} \\ \Psi_3^{n+1} \\ \vdots \end{pmatrix} = \begin{pmatrix} R_1 \\ R_2 \\ R_3 \\ \vdots \end{pmatrix} \quad (4.30)$$

by means of the following manipulations. Multiply the first row of eq. (4.29) by $1/A_1 \equiv U_1$, and further define $R_1 \equiv B_1/A_1 = B_1U_1$. This gives

$$\begin{pmatrix} 1 & U_1 & 0 & 0 & 0 & \dots \\ 1 & A_2 & 1 & 0 & 0 & \dots \\ 0 & 1 & A_3 & 1 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \begin{pmatrix} \Psi_1^{n+1} \\ \Psi_2^{n+1} \\ \Psi_3^{n+1} \\ \vdots \end{pmatrix} = \begin{pmatrix} R_1 \\ B_2 \\ B_3 \\ \vdots \end{pmatrix} \quad (4.31)$$

Next, subtract the first row from the second to obtain

$$\begin{pmatrix} 1 & U_1 & 0 & 0 & 0 & \dots \\ 0 & A_2 - U_1 & 1 & 0 & 0 & \dots \\ 0 & 1 & A_3 & 1 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \begin{pmatrix} \Psi_1^{n+1} \\ \Psi_2^{n+1} \\ \Psi_3^{n+1} \\ \vdots \end{pmatrix} = \begin{pmatrix} R_1 \\ B_2 - R_1 \\ B_3 \\ \vdots \end{pmatrix} \quad (4.32)$$

Dividing the second row by $A_2 - U_1$ then gives

$$\begin{pmatrix} 1 & U_1 & 0 & 0 & 0 & \dots \\ 0 & 1 & U_2 & 0 & 0 & \dots \\ 0 & 1 & A_3 & 1 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \begin{pmatrix} \Psi_1^{n+1} \\ \Psi_2^{n+1} \\ \Psi_3^{n+1} \\ \vdots \end{pmatrix} = \begin{pmatrix} R_1 \\ R_2 \\ B_3 \\ \vdots \end{pmatrix} \quad (4.33)$$

where

$$U_2 \equiv \frac{1}{A_2 - U_1} \quad (4.34)$$

and

$$R_2 \equiv \frac{A_2 - R_1}{A_2 - U_1} = (A_2 - R_1)U_2 \quad (4.35)$$

The procedure generalizes, so that eq. (4.30) is obtained with

$$U_1 = \frac{1}{D_1}, \quad U_i = \frac{1}{A_i - U_{i-1}} \quad (i > 1) \quad (4.36)$$

$$R_1 = B_1 U_1, \quad R_i = (B_i - R_{i-1})U_i \quad (i > 1) \quad (4.37)$$

We can now solve equations (4.30) in reverse order, starting from the last row, which reads simply

$$\Psi_N^{n+1} = R_N \quad (4.38)$$

The next-to-last row reads

$$\Psi_{N-1}^{n+1} + U_{N-1} \Psi_N^{n+1} = R_{N-1} \quad (4.39)$$

so that

$$\Psi_{N-1}^{n+1} = R_{N-1} - U_{N-1} \Psi_N^{n+1} \quad (4.40)$$

and so on. In general we have

$$\Psi_i^{n+1} = R_i - U_i \Psi_{i+1}^{n+1} \quad (4.41)$$

The following sequence of operations summarizes the above steps (written in C):

```

U[1] = 1.0/A[1];
for (i=1; i<N; i++)
    U[i] = 1.0/(A[i] - U[i-1]);

R[1] = B[1]*U[1];
for (i=1; i<N; i++)
    R[i] = (B[i] - R[i-1])*U[i];

psi[N] = R[N];
for (i=N-1; i>0; i--)
    psi[i] = R[i] - U[i]*psi[i+1];

```

As presented, this is actually somewhat wasteful of space – not all quantities need to be stored in separate arrays, for example. For the present problem this inefficiency is relatively harmless, although for more computationally demanding problems it might be worth eliminating.

V. SIMULATION PROJECTS

The projects suggested here are all based on the above scheme. Once the basic code is in place and working (Project 1), the others should be very straightforward – they amount mainly to modifying the potential and/or initial conditions. I recommend that you implement these calculations in a programming language like C, C++ or Fortran, but Matlab or Mathematica could be employed if necessary.

A good way of visualizing the solutions will significantly enhance the value of the projects. One can simply write data files containing the wavefunction (real and imaginary parts and/or magnitude, as desired) at different times. A plotting program like `gnuplot` can then be used to view the action.

Even better is to produce animations, e.g., of scattering processes. A simple and effective way to do this with `gnuplot` in a Unix environment makes use of `gnuplot`'s facility for plotting data from the command line. One has merely to make the code print the necessary `gnuplot` commands to `stdout` and then run it, redirecting (or “piping”) the output directly to `gnuplot`. For example, consider the following code snippet:

```
printf("plot '-' with lines notitle\n");
for (i=1; i<N; i++)
    printf ("%lf %lf\n", x[i], psi[i]);
printf("e\n");
```

The first `printf` issues a `gnuplot` command indicating that data will be entered via the standard input. (It also indicates that the data points will be connected with lines, and that the title will be omitted.) If you entered this on the `gnuplot` command line, it would subsequently expect you to enter pairs of (x, y) data via the keyboard, separated by spaces or tabs and followed by carriage returns. To signal the end of the data entry you would type “e,” at which point your plot would be displayed. The subsequent code above prints just

these items to the standard output. If this code is part of a loop over time steps, then at each time step it generates a new plot with the current values. If the code is then run as, for example,

```
./a.out | gnuplot
```

then the result is an animation of the time evolution.

A detailed example of this is given in the sample code `tdse-gnuplot.c`.

1. The Free Particle

The first project should be a simulation of a free particle ($V = 0$) with a gaussian initial wavefunction. This problem can be solved exactly, so the correctness (or otherwise) of the code can be checked easily.

For complete details see any text on quantum mechanics [1]. The normalized initial wavefunction is given by

$$\Psi(x, 0) = \frac{1}{\pi^{1/4}\sqrt{\sigma}} \exp\left[-\frac{(x - x_0)^2}{2\sigma^2}\right] \exp[ip_0x/\hbar] \quad (5.1)$$

This represents a particle localized to within $\Delta x \sim \sigma$ of x_0 , and with average momentum p_0 . The solution to the Schrödinger equation is then (for $V = 0$):

$$\Psi(x, t) = \frac{1}{\pi^{1/4}\sqrt{\sigma(1 + i\hbar t/m\sigma^2)}} \exp\left[-\frac{(x - (x_0 + p_0t/m))^2}{2\sigma^2(1 + i\hbar t/m\sigma^2)}\right] \exp[i(p_0x - Et)/\hbar] \quad (5.2)$$

where $E = p_0^2/2m$.

In building the simulation you will need an array to hold the values of the wavefunction at each spatial grid point. Say there are N of these. Recall that to update each grid point we need the values on the neighboring grid points; obviously something will need to be done for the first and last point, which are missing one neighbor each. Here you should just specify $\Psi = 0$ for all time; this should introduce no problems as long as the wavepacket is well away from the edges. Alternatively, this is equivalent to putting infinitely high potential “walls” at these locations.

You will also want to adopt a “reasonable” system of units so that the numerical values of quantities like x , p_0 , etc. are neither too large nor too small. One convenient choice

is to use nanometers ($1 \text{ nm} = 10^{-9} \text{ m}$) for distance, femtoseconds ($1 \text{ fs} = 10^{-15} \text{ s}$) for time, and electron volts ($1 \text{ eV} = 1.6 \times 10^{-19} \text{ J}$) for energy. In these units we have

$$\hbar = 0.6582 \text{ eV} \cdot \text{fs} \quad (5.3)$$

$$c = 300 \text{ nm/fs} \quad (5.4)$$

A useful combination of these is $\hbar c = 197.5 \text{ eV} \cdot \text{nm}$. The mass of a particle is conveniently expressed in terms of its rest energy, e.g., for an electron $m_e c^2 = 0.511 \text{ MeV}$ or $511,000 \text{ eV}$. Hence its mass is $m_e = 511,000/300^2 = 5.68$ in these units.

A standard test of a program for time evolution is to evolve the solution forward for a while, then evolve backwards by the same amount. The solution should return to its initial value. Note that this does not directly test whether the time evolution is being simulated *accurately*, only that it is reversible. Compare your result also to the exact solution and see what sorts of values for Δx and Δt are useful. In general we expect the difference approximations to be valid when the change in Ψ from one grid point to the next, or one time step to the next, is small compared to Ψ itself.

Observe the spreading of the wavepacket with time. Try other forms for the initial wavefunction, for example the Lorentzian function

$$\Psi(x, 0) = \frac{A^2}{(x - x_0)^2 + \gamma^2} e^{ip_0 x} \quad (5.5)$$

where $A^4 = 2\gamma^3/\pi$, or or a step function or “tent” function multiplied by $\exp(ip_0 x)$.

Add code that will calculate the expectation value of the energy at any time, and verify that this quantity is conserved.

Verify that the evolution is unitary, that is, that the wavefunction remains properly normalized as it evolves in time.

Experiment with either of the unsuitable approaches to the discretization, and note what happens.

2. Scattering

An extremely instructive application of the simulation is to study the scattering of a wavepacket from a potential barrier or well. Modify your code to do this.

Examine the behavior for infinite and finite barriers. The infinite barrier can be simulated by the requirement that $\Psi = 0$ at the location of the barrier. For a finite barrier, observe the penetration of the wavefunction into the classically forbidden region. According to the semi-classical approximation, when a wavepacket with average energy E encounters a barrier of height $V > E$, it persists in the classically forbidden region for a time

$$T \sim \frac{\hbar}{\sqrt{E(V - E)}} \quad (5.6)$$

Check this result using your simulation. What happens as $V \rightarrow E$?

Next, consider scattering off of barriers and wells of finite width. Use your simulation to calculate the fraction of the initial probability that is transmitted and reflected, and compare to the textbook results derived using plane waves.

For the case of the finite well, note the formation of a metastable state that it trapped (for a while) in the well. The initial wavepacket separates into a “prompt” component that emerges immediately, and a delayed component that leaks out more slowly.

3. Energy Eigenstates

Study the behavior of a wavepacket in a harmonic oscillator potential,

$$V(x) = \frac{1}{2}m\omega_0^2x^2 \quad (5.7)$$

Begin by setting the initial wavefunction to be a harmonic oscillator stationary state [1], and confirm that the probability density $|\Psi|^2$ is time-independent for such states.

Next, study linear combinations of a few stationary states. In this case you can observe the oscillation of the wave packet in the potential well, with a frequency you can calculate. Compare the calculation to what your simulation shows.

Verify again that the expectation value of the energy is constant, and has the expected values.

You can also study the evolution of gaussian or other wavepackets in this potential.

4. Periodic Perturbations

Consider again the harmonic oscillator and add to the Hamiltonian a time-dependent interaction of the form

$$V(x, t) = Fx \cos \omega t \quad (5.8)$$

where F is a constant. In classical theory this would correspond to a sinusoidal (in time) external driving force. In quantum theory it can be thought of as inducing transitions between the stationary states of the unperturbed harmonic oscillator.

Let's assume the particle starts out at $t = 0$ in its ground state ($n = 0$). The according to lowest-order perturbation theory it will be found in the first excited state ($n = 1$) at time t with probability

$$P(t) = \frac{F^2}{2\hbar m \omega_0} \left(\frac{\sin[(\omega - \omega_0)t/2]}{\omega - \omega_0} \right)^2 \quad (5.9)$$

where $\omega_0 \equiv \sqrt{k/m}$ is the natural frequency of the oscillator [1]. (The probability for transitions to other states is zero in lowest order.) This is actually an approximation to the full perturbation theory result, valid when we are “near resonance,” i.e., when

$$|\omega_0 - \omega| \ll \omega_0 + \omega \quad (5.10)$$

The full expression may be found in the standard texts.

Note that the transition probability itself oscillates in time with a period $2\pi/|\omega - \omega_0|$, and has its maximum value at $t = \pi/|\omega - \omega_0|$, where

$$P_{max} = \frac{F^2}{2\hbar m \omega_0 (\omega - \omega_0)^2} \quad (5.11)$$

This should be significantly less than one or perturbation theory is invalid.

Modify your harmonic oscillator code from Project 3 to include such a contribution and study the transitions. Compare the “exact” transition probability (i.e., computed using your solution of the full Schrödinger equation) to the result of perturbation theory. Examine the transition probability as a function of time and observe its periodic behavior.

[1] For additional details on background, see a standard introductory text on quantum mechanics, e.g., D.J. Griffiths, *Introduction to Quantum Mechanics*, 2nd ed. (Prentice Hall, 2005); R.L. Liboff, *Introductory Quantum Mechanics*, 4th ed. (Addison Wesley, 2003).

- [2] For example, S.E. Koonin, *Computational Physics* (Benjamin/Cummings, 1985); T. Pang, *An Introduction to Computational Physics* (Cambridge University Press, 2006); L.D. Fosdick, E.R. Jessup, C.J.C. Schauble, and G. Domik, *Introduction to High-Performance Scientific Computing* (MIT Press, 1996).
- [3] It also encodes the probabilities for any other measurement, for example of the momentum.
- [4] There is an important case where non-normalizable wavefunctions are useful despite their unphysicality: plane wave solutions for a free particle ($V = 0$), corresponding to a particle with a definite momentum. In this case the lack of normalizability is connected to the failure of such states to properly respect the uncertainty principle: a particle with a definite p would have $\Delta p = 0$ and hence $\Delta x = \infty$. However, normalizable states *can* be constructed as linear combinations of these plane waves, and, indeed, if sufficient care is exercised, the plane waves themselves can often be used directly to obtain physical results (e.g., transition and reflection probabilities for potential scattering).
- [5] E can be shown to be a real number, assuming V is real.

GLOSSARY

classically forbidden region	Region of space in which the total energy is less than the potential energy; forbidden in classical theory, but quantum wavefunctions can extend into such regions.
Crank-Nicholson method	Discretization of the Schrödinger equation that is unitary and stable.
energy eigenstate	Eigenstate of the Hamiltonian operator for a system. A stationary state.
expectation value	Average value of many measurements on identically prepared systems.
Hamiltonian	In quantum theory, the operator corresponding to the total energy of a system.
harmonic oscillator	Classical or quantum system in which the potential energy is proportional to x^2 .
normalization	Requirement applied to wavefunctions; its physical content is that the probability of finding the particle somewhere should be unity.
perturbation theory	Approximation scheme based on considering small perturbations of a solvable problem.
resonance	Condition when an oscillatory system is driven at close to its natural frequency.
round-off error	Error in floating-point computations on a computer introduced due to the discrete representation of real numbers.
row reduction	Matrix technique for solving systems of linear equations.
Schrödinger equation	The central equation of quantum theory, which determines the time development of the wavefunction.

separation of variables	Technique for separating a partial differential equation into ordinary differential equations. The basic assumption is a product form for the solutions.
stationary state	Eigenstate of the Hamiltonian operator for a system. An energy eigenstate.
transition probability	The probability that a quantum system initially in some state will be found in another state at some later time.
unitarity	Property of quantum systems that insures the wavefunction normalization remains constant in time; reflects conservation of probability.
wavefunction	The entity that describes the quantum state of a system. Its modulus squared gives the probability density for position.
wavepacket	A superposition of plane waves to produce a wavefunction that is localized in space.