

The One-Dimensional Finite-Difference Time-Domain (FDTD) Algorithm Applied to the Schrödinger Equation

James R. Nagel

MS, Electrical Engineering
nagel@mers.byu.edu

1. INTRODUCTION

For anyone who has ever studied quantum mechanics, it is well-known that the Schrödinger equation can be very difficult to solve analytically. Occasionally, certain situations allow for approximate solutions through the use of the WKB method or perturbation theory, but the vast majority of physical systems that occur in nature are far too complicated to ever solve by hand. Therefore, it is desirable to enlist the aid of computers when searching for solutions to quantum systems.

One of the more common methods for numerically solving a time-dependent partial differential equation (PDE) is the finite-difference time-domain algorithm, or FDTD. The basic idea behind FDTD is to discretize the PDE in space and time and then approximate the derivatives by using finite differences. Essentially, the PDE is allowed to play itself out by gradually incrementing the time variable in discrete steps. The power behind FDTD is its simplicity of implementation and its ability to visualize the solutions as they act out in both space and time.

This paper derives the numerical update equations for the one-dimensional Schrödinger equation and then solves for the stability conditions on their use. Several examples are also given to demonstrate the FDTD in action.

2. UPDATE EQUATIONS

The one-dimensional time-dependent Schrödinger equation is given as

$$j\hbar \frac{\partial \psi(x, t)}{\partial t} = -\frac{\hbar^2}{2m} \frac{\partial^2 \psi(x, t)}{\partial x^2} + V(x)\psi(x, t), \quad (1)$$

where $\psi(x, t)$ is the wave function in space and time, $V(x)$ is the potential function, m is the particle mass, and \hbar is Planck's constant. The first step in the FDTD algorithm is to establish a discrete grid in space and time by defining the points x_ℓ and t_n such that

$$x_\ell = \ell \Delta x, \quad \text{and} \quad (2)$$

$$t_n = n \Delta t, \quad (3)$$

where $0 \leq n \leq N$ and $0 \leq \ell \leq L$. The wave function at a specific grid point can now be defined in terms of a *stencil*, which is a short-hand notation that defines the wave function at specific points on the grid such that

$$\psi(x_k, t_n) = \psi^n(\ell). \quad (4)$$

In practice, the function $\psi^n(\ell)$ is often called the *present state* of the wave function at a given time step t_n . Similarly, the state $\psi^{n+1}(\ell)$ is called the *future state* and $\psi^{n-1}(\ell)$ is called the *past state*. The goal of the FDTD algorithm is to solve for an unknown future state $\psi^{n+1}(\ell)$ in terms of known past and present states. The time step is then incremented to t_{n+1} , thereby promoting the future state to present status, and the present state to the past. The iteration then repeats itself and solves for the next future state of the system.

Because complex arithmetic can be numerically costly, it is helpful to break up the wave function into real and imaginary components such that

$$\psi(x, t) = \psi_R(x, t) + j\psi_I(x, t) . \quad (5)$$

This step allows us to treat each component separately and perform only real-valued computations with each part. Each grid component also has its own point in the stencil written as

$$\psi^n(\ell) = \psi_R^n(\ell) + j\psi_I^n(\ell) . \quad (6)$$

Plugging back into the Schrödinger equation produces two coupled partial differential equations of the form

$$\hbar \frac{\partial \psi_R(x, t)}{\partial t} = -\frac{\hbar^2}{2m} \frac{\partial^2 \psi_I(x, t)}{\partial x^2} + V(x)\psi_I(x, t) , \quad (7)$$

$$\hbar \frac{\partial \psi_I(x, t)}{\partial t} = +\frac{\hbar^2}{2m} \frac{\partial^2 \psi_R(x, t)}{\partial x^2} - V(x)\psi_R(x, t) . \quad (8)$$

The next step is to approximate the partial derivative in time by using a finite central-difference on the wave function along the discretized grid. For the real component, this gives

$$\frac{\partial \psi_R(x, t)}{\partial t} \approx \frac{\psi_R^{n+1}(\ell) - \psi_R^{n-1}(\ell)}{2\Delta t} \quad (9)$$

with a similar expression for the imaginary component. Applying a central-difference on the spatial derivative also gives an approximation to the second-partial with the form

$$\frac{\partial^2 \psi_R(x, t)}{\partial x^2} \approx \frac{\psi_R^n(\ell + 1) - 2\psi_R^n(\ell) + \psi_R^n(\ell - 1)}{\Delta x^2} . \quad (10)$$

Plugging these approximations back into Equations 7 and 8 then gives

$$\hbar \frac{\psi_R^{n+1}(\ell) - \psi_R^{n-1}(\ell)}{2\Delta t} = -\frac{\hbar^2}{2m} \frac{\psi_I^n(\ell + 1) - 2\psi_I^n(\ell) + \psi_I^n(\ell - 1)}{\Delta x^2} + V(\ell)\psi_I^n(\ell) \quad (11)$$

and

$$\hbar \frac{\psi_I^{n+1}(\ell) - \psi_I^{n-1}(\ell)}{2\Delta t} = +\frac{\hbar^2}{2m} \frac{\psi_R^n(\ell + 1) - 2\psi_R^n(\ell) + \psi_R^n(\ell - 1)}{\Delta x^2} - V(\ell)\psi_R^n(\ell) \quad (12)$$

Finally, solve for $\psi_R^{n+1}(\ell)$ and $\psi_I^{n+1}(\ell)$ to find

$$\psi_R^{n+1}(\ell) = -c_1 [\psi_I^n(\ell + 1) - 2\psi_I^n(\ell) + \psi_I^n(\ell - 1)] + c_2 V(\ell)\psi_I^n(\ell) + \psi_R^{n-1}(\ell) , \quad (13)$$

$$\psi_I^{n+1}(\ell) = +c_1 [\psi_R^n(\ell + 1) - 2\psi_R^n(\ell) + \psi_R^n(\ell - 1)] - c_2 V(\ell)\psi_R^n(\ell) + \psi_I^{n-1}(\ell) , \quad (14)$$

where the constants c_1 and c_2 are defined as

$$c_1 = \frac{\hbar\Delta t}{m\Delta x^2} , \quad \text{and} \quad (15)$$

$$c_2 = \frac{2\Delta t}{\hbar} . \quad (16)$$

Together, Equations 13 and 14 are called *update equations* because they give the future state $\psi^{n+1}(\ell)$ of the wave function at a point ℓ in terms of nearby points in space and time. The FDTD algorithm begins by looping over the spatial domain to solve for $\psi^{n+1}(\ell)$ at every value of ℓ . Once a future state has been found, the time step is incremented by one, thereby assigning the present state to the past and the future to the present:

$$\begin{aligned} \psi^n &\rightarrow \psi^{n-1} , \\ \psi^{n+1} &\rightarrow \psi^n . \end{aligned}$$

The loop then repeats itself by solving for the next future state of ψ at every value of ℓ , until finally terminating when $n = N$.

3. BOUNDARY CONDITIONS

In order to fully define a PDE, it is necessary to include special conditions at the boundaries of the spatial domain. The simplest condition to implement is known as the *Dirichlet* boundary condition, which simply defines a constant value at the boundaries of a PDE such that

$$\psi^n(0) = a, \quad \psi^n(L) = b . \quad (17)$$

An important aspect about quantum mechanical systems is that a nonzero value imposed at a boundary will act as a *source* of probability. Because it is generally desirable to conserve probability in a quantum simulation, the preferable choice of Dirichlet boundary conditions is to set $a = b = 0$ so that

$$\psi^n(0) = 0, \quad \psi^n(L) = 0 . \quad (18)$$

A boundary condition such as this is often called a *hard boundary* because it effectively imposes an infinite potential at the edges of the simulation. As a consequence, any wave incident at the boundary will be reflected back into the simulation domain. This is generally an undesirable behavior, but it is difficult to overcome. The simplest solution is to enlarge the simulation domain until nothing has time to reach the boundary, but this can be very wasteful of computer power. A far more effective solution is the *absorbing boundary condition* (ABC), often called the *Mur* condition, but the algorithm is very complex to implement.

4. STABILITY

Suppose the potential function is a constant such that $V(\ell) = V_0$. Solutions to the Schrödinger equation then take on the form of a free particle with a wave function given by

$$\psi(x, t) = e^{j(kx - \omega t)} , \quad (19)$$

where k is the particle wavenumber and ω is the angular frequency. In terms of the FDTD stencil, this can be written as

$$\psi^n(\ell) = e^{j(k\ell\Delta x - \omega n\Delta t)} . \quad (20)$$

If the time step is advanced by one unit, then the stencil becomes

$$\psi^{n+1}(\ell) = e^{j(k\ell\Delta x - \omega n\Delta t) - \omega\Delta t} , \quad (21)$$

$$= e^{j(A-B)} , \quad (22)$$

where the constants A and B are defined as

$$A = k\ell\Delta x - \omega n\Delta t , \quad \text{and} \quad (23)$$

$$B = \omega\Delta t . \quad (24)$$

Similarly, we can increment the grid in space to find

$$\psi^n(\ell + 1) = e^{j(A+C)} , \quad (25)$$

where $C = k\Delta x$. In terms of the grid stencil, the real and imaginary components of the future state of the wave function can now be expressed as

$$\psi_R^{n+1}(\ell) = \cos(A - B) = \cos(A)\cos(B) + \sin(A)\sin(B) , \quad (26)$$

$$\psi_I^{n+1}(\ell) = \sin(A - B) = \sin(A)\cos(B) - \cos(A)\sin(B) . \quad (27)$$

Next, plug Equations 26 and 27 back into Equation 13. After a little simplification, the result is found to be

$$\cos(A)\cos(B) + \sin(A)\sin(B) = -c_1 [\sin(A + C) - 2\sin(A) + \sin(A - C)] + c_2 V_0 \sin(A) + \cos(A + B) . \quad (28)$$

Note that a similar result would be obtained if we instead plugged into Equation 14.

The importance of Equation 28 is that places constraints on the available choices for c_1 and c_2 . No matter what values are chosen, the update equations will force this equality to be true. As a consequence, if c_1 and c_2 are not properly defined, it is possible to require A and B to be imaginary in order to maintain an equality. If either A or B become imaginary, then a real value is introduced into the exponent of Equation 21. Consequently, as time increments, the wave function increases exponentially without bound, and the simulation is said to *blow up*.

In order to maintain a stable simulation, it is necessary to choose the constants c_1 and c_2 such that Equation 28 is satisfied by only real values of A and B . The simplest way to do this is by choosing a time step Δt that prevents the right-hand side from exceeding the bounds of the left-hand side. Thus, for all real-valued A and B , we must ensure the condition

$$-1 \leq \cos(A) \cos(B) + \sin(A) \sin(B) \leq 1 . \quad (29)$$

The upper bound of this expression occurs when $\sin(A) = \sin(B) = 1$ and $\cos(A) = \cos(B) = 0$. Plugging into the right-hand side of Equation 28 and simplifying then gives

$$2 \geq -c_1 [\sin(A + C) - 2 + \sin(A - C)] + c_2 V_0 . \quad (30)$$

Note that if $\sin(A) = 1$, then $\sin(A + C) = \sin(A - C)$. The extreme value of the right-hand side then occurs when $\sin(A + C) = -1$, and the final result is

$$2 \geq 4c_1 + c_2 V_0 , \quad (31)$$

or equivalently

$$2 \geq \frac{4\hbar\Delta t}{m\Delta x^2} + \frac{2\Delta t V_0}{\hbar} . \quad (32)$$

Finally, solve for Δt to find

$$\Delta t \leq \frac{\hbar}{\frac{2\hbar^2}{m\Delta x^2} + V_0} . \quad (33)$$

The upper bound on Δt is called the *critical time step*, Δt_c , and represents the maximum allowable time increment that will maintain a stable simulation. If the potential function is not constant, then the critical time step is found by simply replacing V_0 with the maximum potential over the simulation interval.

5. EXAMPLES: FDTD IN ACTION

This next section demonstrates the one-dimensional FDTD algorithm in action using the Python programming language. For convenience, it helps to set all physical constants to unit values such that $\hbar = m = \Delta x = 1$. Solving for the critical time step therefore gives $\Delta t_c = 1/2$. The size of the simulation domain is determined by choosing an appropriate value for L . Although larger values produce finer spatial resolution, they also require more computational power per time step. For all of the following examples, the particle is initialized to that of a free particle enclosed in a Gaussian envelope with a variance of 40 grid points.

5.1. Constant Potential

The simplest scenario to simulate is that of a free particle traveling through a constant potential of $V(\ell) = 0$, shown in Figure 1. As can be seen, the particle travels to right and exhibits *dispersion* along the way. Because the wave packet is localized in space by the Gaussian envelope, its momentum is necessarily spread out among a continuous spectrum. Since different momentum components all travel at their own velocities, the wave packet spreads out as it travels. The only way to lessen the dispersive effects would be initialize the wave with a broader envelope, but only an infinitely long wave would exhibit zero dispersion.

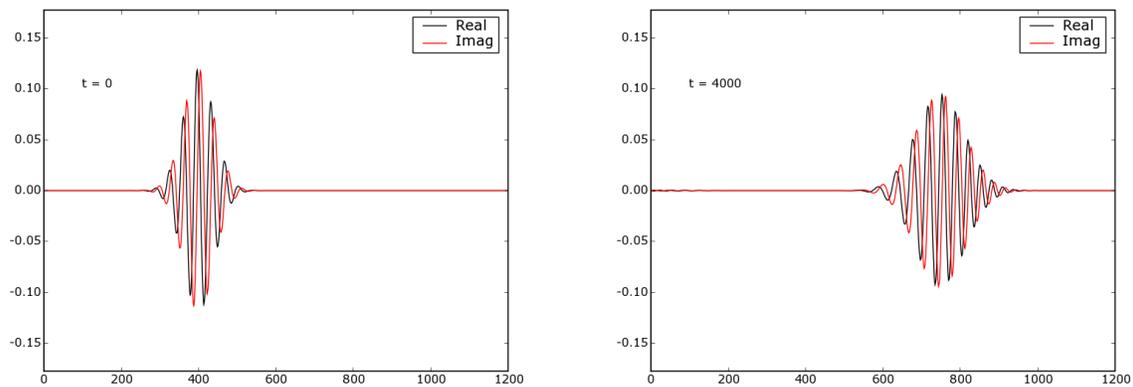


Figure 1. A wave packet traveling to the right.

5.2. Step Potential

The next example is a step potential defined by $V(\ell) = V_0$ for all ℓ greater than $L/2$ and zero otherwise. Shown in figure 2, the wave packet collides with the step potential and splits into two components. Because the kinetic energy of the particle is greater than the potential of the step, some of the wave is able to transmit through while the rest is reflected away.

5.3. Tunneling

One of the more interesting predictions of quantum mechanics is that of *tunneling* through a potential barrier where V_0 is greater than the kinetic energy of the particle. Demonstrated in Figure 3, the wave packet collides with a barrier that is 20 grid points thick. Note how most of the wave reflects away, but some of the wave is still able to penetrate through and emerge from the other side of the barrier.

REFERENCES

1. A. Soriano, *Analysis of the finite difference time domain technique to solve the Schrodinger equation for quantum devices*, Journal of Applied Physics, Vol 95, N12, June 2004.
2. E. Merzbacher, *Quantum Mechanics*, Third Edition, John Wiley & Sons Inc, 1998.

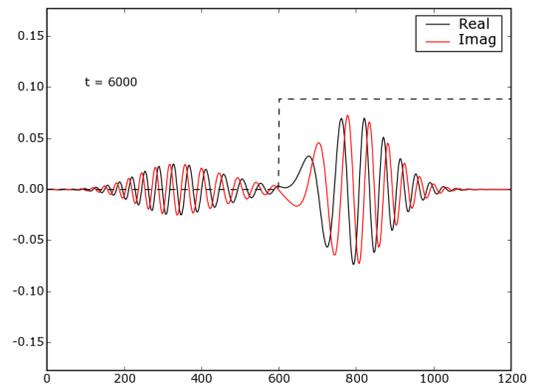
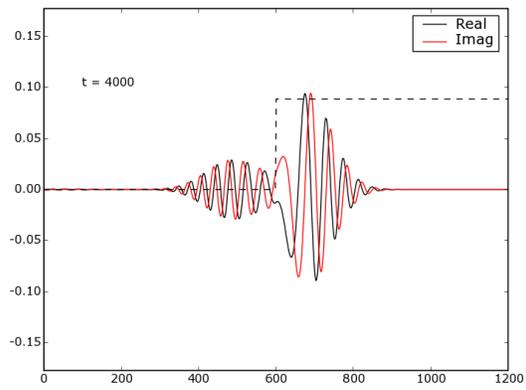
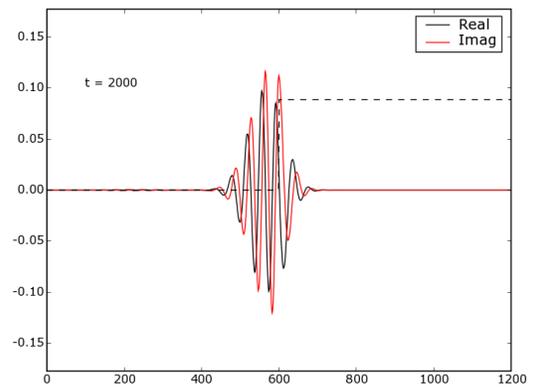
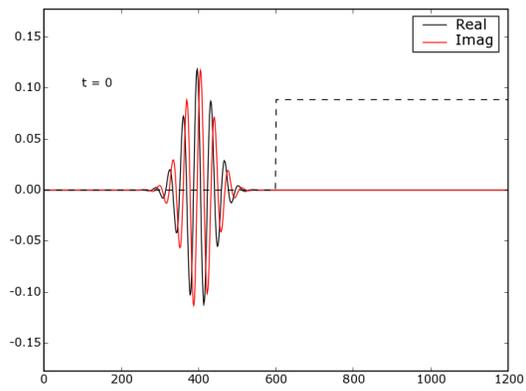


Figure 2. A wave packet incident on a step potential. Some of the wave is transmitted while the rest is reflected.

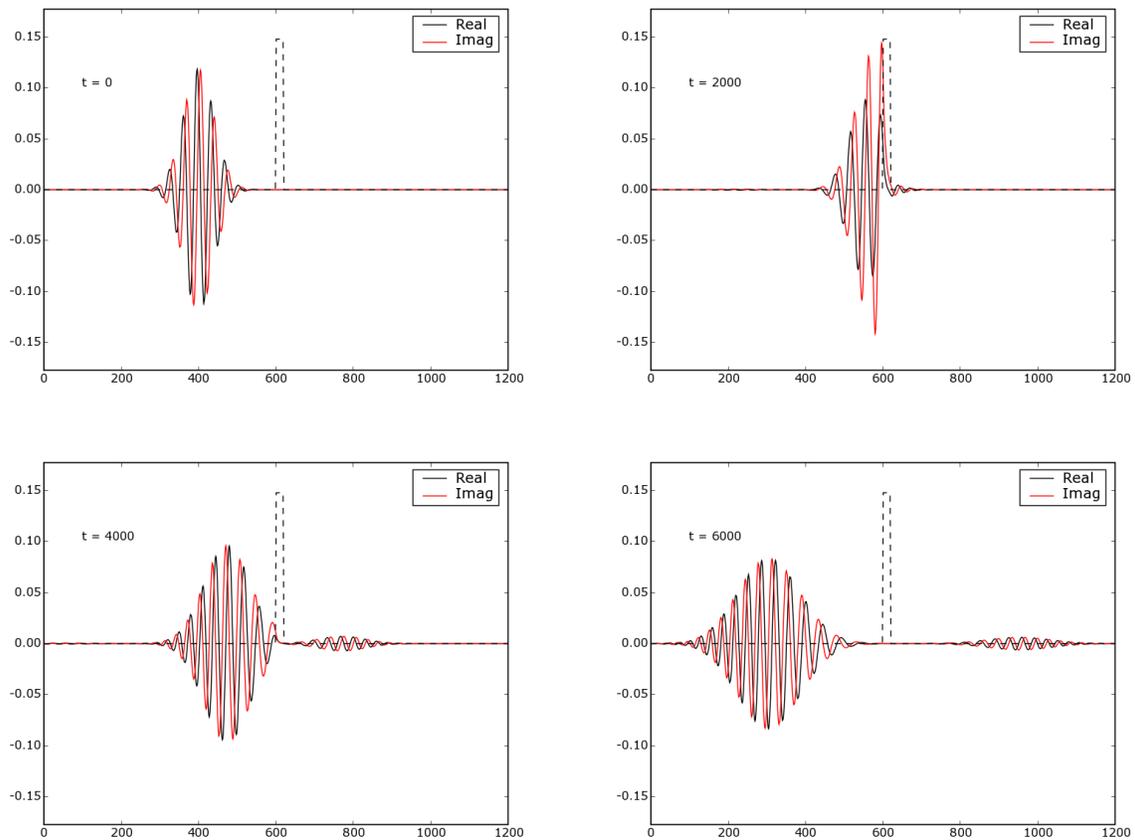


Figure 3. A wave packet incident on a potential barrier that is greater than the kinetic energy of the particle. Some of the wave tunnels through the barrier while the rest is reflected.