

# Simulation of Time-Dependent Schrödinger Equation in the Position and Momentum Domains

**Michael Jennings**

Chicago, IL, USA

Email: [jenm.79@gmail.com](mailto:jenm.79@gmail.com)

Received 2 August 2015; accepted 4 September 2015; published 7 September 2015

Copyright © 2015 by author and Scientific Research Publishing Inc.  
This work is licensed under the Creative Commons Attribution International License (CC BY).  
<http://creativecommons.org/licenses/by/4.0/>



Open Access

---

## Abstract

The paper outlines the development of a new, spectral method of simulating the Schrödinger equation in the momentum domain. The kinetic energy operator is approximated in the momentum domain by exploiting the derivative property of the Fourier transform. These results are compared to a position-domain simulation generated by a fourth-order, centered-space, finite-difference formula. The time derivative is approximated by a four-step predictor-corrector in both domains. Free-particle and square-well simulations of the time-dependent Schrödinger equation are run in both domains to demonstrate agreement between the new, spectral methods and established, finite-difference methods. The spectral methods are shown to be accurate and precise.

## Keywords

Schrödinger Equation, Predictor-Corrector, Fourier Transform, Momentum Domain

---

## 1. Introduction

This paper outlines the development of simulations of the time-dependent Schrödinger equation produced in both position and momentum domains. In the position domain this is the x-y plane. In the momentum domain this is the  $k_x$ - $k_y$  plane, as it is the Fourier transform of the position domain [1]. The simulations demonstrate the accuracy of the spectral methods used in the momentum domain.

All simulations are advanced in time using a four-step predictor-corrector method. The predictor-corrector can be applied independently in both position and momentum domains to step the simulation forward in time. The predictor-corrector is generated using Lagrange polynomials, outlined by [2] and [3]. The predictor formula found here is shown to be consistent with established Adams-Bashforth formulas [3].

**How to cite this paper:** Jennings, M. (2015) Simulation of Time-Dependent Schrödinger Equation in the Position and Momentum Domains. *American Journal of Computational Mathematics*, 5, 291-303.  
<http://dx.doi.org/10.4236/ajcm.2015.53027>

The position-domain approximation of the kinetic energy operator is derived using Lagrange polynomials and consistent with results from [4]. In the position domain the approximation to the kinetic energy operator is fourth-order accurate. In the momentum domain, the kinetic energy operator approximation is global-order accurate because it relies on the derivative property of the Fourier transform [5]. The software written to generate these simulations uses the Fastest Fourier Transform in the West (FFTW) to transform between position and momentum domains [6]. Simulating the time-dependent Schrödinger equation in the momentum domain achieves higher orders of spatial accuracy. The performance and precision of momentum-domain simulations is comparable to position-domain simulations.

Given an initial state  $y_0$  at  $t = 0$ , the four-step predictor-corrector requires the creation of  $y_{n-j} = y_n - j\Delta t$  for  $j = 1, \dots, 4$  in order to compute the first predictor-corrector time-step. A simple backwards Euler method, outlined in [4], [7]-[12], is used to generate the wave function at these early time-steps. Each of these early states for  $j = 1, \dots, 4$  is re-normalized after their creation to ensure minimum initial error.

The first simulation is a free particle with no imposed boundary conditions, when the Hamiltonian consists only of the kinetic energy operator. This simulation demonstrates the difference in boundary conditions of each domain. In the position domain, this is equivalent to an infinite square-well potential, or particle-in-a-box. When the wave function reaches one boundary, it is reflected back. In the momentum domain, this is equivalent to periodic boundary conditions. When the wave function disappears into one boundary it will reappear in the opposite boundary, travelling in the same direction. This is to establish a relative performance benchmark when only the kinetic energy operator is applied.

Second, a finite square-well potential of 100 eV is imposed in both domains. This simulation demonstrates the computational burden associated with imposing the same initial and boundary conditions in both domains. Application of the potential function in the position domain is carried out by entry-wise multiplication of the wave function and potential function lattices. In the momentum domain, this operation is equivalent to convolution. Rather than carry out this time-consuming operation, the wave function in the momentum domain is transformed back to the position domain at every time-step in the simulation in order to apply the potential function. The kinetic energy operator is applied when the wave function has been transformed forward into the momentum domain.

Each of the simulations begins with an initial state of a two-dimensional wave packet with a Gaussian envelope. The simulations are stepped forward in time and the complex-valued wave function components and densities, as well as some expectation values, are captured incrementally. The wave function components and densities are converted to image format and animated [13].

## 2. Methods

The following subsections outline the numerical methods used to generate solutions to the Schrödinger equation.

$$i\hbar \frac{\partial}{\partial t} \Psi = \frac{-\hbar^2}{2m} \nabla^2 \Psi + V\Psi \quad (1)$$

### 2.1. Time Derivative

The same four-step predictor-corrector method is used to step the position and momentum domain simulations forward in time. The predictor and corrector start with the basic form of the differential equation  $y' = f(y)$ . The substitutions  $y(r, t_n) = y_n$  and  $f(y_n, t_n) = f_n$  are made for the following formulas. A four-step predictor-corrector requires five equally-spaced sample points in time. The leading, or current step is  $y_n$ . The sample points are distributed in time according to  $y_{n-j} = y_n - j\Delta t$  for  $j = 0, \dots, 4$  and a fixed  $\Delta t$ .

#### 2.1.1. Predictor

The predictor uses the integral form of the differential equation.

$$y' = f(y) \Rightarrow \tilde{y}_n - y_{n-1} = \int_{t_{n-1}}^{t_n} f_{n-1} dt \quad (2)$$

The function  $f$  is approximated by Lagrange polynomials [2]. Once the polynomial approximation to  $f$  has been substituted, the integral is straightforward to compute. This yields Adams-Bashforth coefficients which calculate the predicted value  $\tilde{y}_n$  [3].

$$\tilde{y}_n = y_{n-1} + \Delta t \left( \frac{55}{24} f_{n-1} - \frac{59}{24} f_{n-2} + \frac{37}{24} f_{n-3} - \frac{9}{24} f_{n-4} \right) \quad (3)$$

### 2.1.2. Corrector

The corrector uses the original form of the differential equation.

$$y' = f(y) \Rightarrow \frac{\partial}{\partial t} y_n = f_n \quad (4)$$

For the corrector,  $y$  is approximated by Lagrange polynomials [2]. Once the polynomial approximation to  $y$  has been substituted, the first derivative is straightforward to compute. This yields the following coefficients which calculate the value  $f_n$ .

$$\frac{25}{12\Delta t} y_n - \frac{48}{12\Delta t} y_{n-1} + \frac{36}{12\Delta t} y_{n-2} - \frac{16}{12\Delta t} y_{n-3} + \frac{3}{12\Delta t} y_{n-4} = f_n \quad (5)$$

The predicted value  $\tilde{y}_n$  is substituted for  $y_n$  in the function  $f$  to get  $\tilde{f}_n$ .

$$y_n = \frac{12\Delta t}{25} \tilde{f}_n + \frac{48}{25} y_{n-1} - \frac{36}{25} y_{n-2} + \frac{16}{25} y_{n-3} - \frac{3}{25} y_{n-4} \quad (6)$$

### 2.1.3. Application to the Schrödinger Equation

In the position domain, the Schrödinger equation is written as follows using operator notation as shorthand. The Hamiltonian operator  $\hat{H}$  is written with a  $P$  superscript to denote the position domain and with superscript  $M$  to denote the momentum domain.

$$\frac{\partial}{\partial t} \Psi = \frac{i\hbar}{2m} \nabla^2 \Psi + \frac{-i}{\hbar} V \Psi = \frac{-i}{\hbar} \hat{H}^P \Psi \quad (7)$$

At every time step, the predictor calculation is carried out.

$$\tilde{\Psi}_n = \Psi_{n-1} + \Delta t \left( \frac{55-i}{24} \frac{\hat{H}^P}{\hbar} \Psi_{n-1} - \frac{59-i}{24} \frac{\hat{H}^P}{\hbar} \Psi_{n-2} + \frac{37-i}{24} \frac{\hat{H}^P}{\hbar} \Psi_{n-3} - \frac{9-i}{24} \frac{\hat{H}^P}{\hbar} \Psi_{n-4} \right) \quad (8)$$

That result is plugged in to the corrector calculation to advance the simulation forward one time-step.

$$\Psi_n = \frac{12\Delta t}{25} \frac{-i}{\hbar} \hat{H}^P \tilde{\Psi}_n + \frac{48}{25} \Psi_{n-1} - \frac{36}{25} \Psi_{n-2} + \frac{16}{25} \Psi_{n-3} - \frac{3}{25} \Psi_{n-4} \quad (9)$$

The following sections outline the development of the Hamiltonian operator  $\hat{H}^P$  in the position domain and  $\hat{H}^M$  in the momentum domain.

## 2.2. Free Particle, Kinetic Energy Operator

The first simulations in the position and momentum domains assume free particle conditions. The particle is given the mass of an electron. Only the kinetic energy operator applies in the Hamiltonian.

$$i\hbar \frac{\partial}{\partial t} \Psi = \frac{-\hbar^2}{2m} \nabla^2 \Psi = \hat{H}^P \Psi \quad (10)$$

Before the simulations are started, the position- and momentum-domain lattices must be constructed. This requires fixing the position-domain step sizes  $\Delta x$  and  $\Delta y$  as well as the number of columns  $N_x$  and number of rows  $N_y$ . It is also helpful to define an origin point,  $x_c = \frac{N_x}{2}$  and  $y_c = \frac{N_y}{2}$ . The position domain is defined by  $X_j = (j - x_c) \Delta x$  for  $j = 0, \dots, N_x - 1$  and  $Y_j = (y_c - j) \Delta y$  for  $j = 0, \dots, N_y - 1$ . The reversal of the  $y$ -direction accounts for the fact that computer storage increments the row index as the row moves down.

The momentum domain lattices are constructed according to the relationship  $\Delta k_x = \frac{1}{N_x \Delta x}$ . The momentum

domain is defined by  $K_{x,j} = \frac{(j-x_c)}{N_x \Delta x}$  for  $j=0, \dots, N_x-1$  and  $K_{y,j} = \frac{(y_c-j)}{N_y \Delta y}$  for  $j=0, \dots, N_y-1$ . The

high frequencies have been shifted into the negative frequencies. Use of the FFTW library requires applying a phase-shift to the position domain before transforming into the momentum domain if negative frequencies are used instead of high frequencies [6].

### 2.2.1. Position Domain

The approximation to the kinetic energy operator in the position domain was generated using Lagrange polynomials. This was accomplished by approximating the second derivative in one dimension, as the same formula can be applied to all dimensions. This is a centered-space formula accurate to fourth order, requiring five sample points. In terms of the generalized coordinate  $q$ , the sample points  $q_j = q_2 + (j-2)\Delta q$  for  $j=0, \dots, 4$  and a fixed  $\Delta q$  describe the set of sample points centered around  $q_2$ . For a function  $f(q)$ , where  $f_j = f(q_j)$ , the polynomial approximation to  $f$  is substituted and the second derivative is calculated. This procedure yields the following approximation to the Laplacian operator:

$$\nabla^2 f_2 = \frac{-1}{12\Delta q^2} (f_0 - 16f_1 + 30f_2 - 16f_3 + f_4) \quad (11)$$

The real and imaginary parts of the wave function  $\Psi = R + iI$  are calculated independently, yielding a pair of coupled equations.

$$\frac{-i}{\hbar} \hat{H}^P I = \frac{\hbar}{2m} \nabla^2 R \quad (12)$$

$$\frac{-i}{\hbar} \hat{H}^P R = \frac{-\hbar}{2m} \nabla^2 I \quad (13)$$

Denote spatial sample points with subscripts:  $f(x_2, y_2) = f_{x_2, y_2}$  and substitute the approximation to the Laplacian. The simulation can be stepped forward in time once the approximation to the Hamiltonian has been substituted into the predictor corrector formula.

$$\begin{aligned} \frac{-i}{\hbar} \hat{H}^P I_{x_2, y_2} &= \frac{-\hbar}{2m} \left[ \left( \frac{1}{12\Delta x^2} \right) (R_{x_0, y_2} - 16R_{x_1, y_2} + 30R_{x_2, y_2} - 16R_{x_3, y_2} + R_{x_4, y_2}) \right. \\ &\quad \left. + \left( \frac{1}{12\Delta y^2} \right) (R_{x_2, y_0} - 16R_{x_2, y_1} + 30R_{x_2, y_2} - 16R_{x_2, y_3} + R_{x_2, y_4}) \right] \end{aligned} \quad (14)$$

$$\begin{aligned} \frac{-i}{\hbar} \hat{H}^P R_{x_2, y_2} &= \frac{\hbar}{2m} \left[ \left( \frac{1}{12\Delta x^2} \right) (I_{x_0, y_2} - 16I_{x_1, y_2} + 30I_{x_2, y_2} - 16I_{x_3, y_2} + I_{x_4, y_2}) \right. \\ &\quad \left. + \left( \frac{1}{12\Delta y^2} \right) (I_{x_2, y_0} - 16I_{x_2, y_1} + 30I_{x_2, y_2} - 16I_{x_2, y_3} + I_{x_2, y_4}) \right] \end{aligned} \quad (15)$$

### 2.2.2. Momentum Domain

The approximation to the kinetic energy operator in the momentum domain was generated using the transform of the derivative operator. Because the momentum domain is the Fourier transform of the position domain, the derivative operator is transformed as well. In terms of the generalized position-domain coordinate  $q$ , let the function  $F(s)$  be the Fourier transform of the function  $f(q)$ , where  $\mathcal{FT}\{f(q)\} = F(s)$  and  $s$  is the generalized momentum-domain coordinate.

$$\mathcal{FT} \left\{ \frac{d}{dq} f(q) \right\} = i2\pi s F(s) \Rightarrow \mathcal{FT} \left\{ \frac{d^2}{dq^2} f(q) \right\} = -4\pi^2 s^2 F(s) \quad (16)$$

The initial position-domain state  $\Psi$  is transformed forward into the momentum-domain state  $\Phi$ , where  $\mathcal{FT}\{\Psi(x, y)\} = \Phi(k_x, k_y)$ . The real and imaginary parts of  $\Phi = \mathcal{R} + i\mathcal{I}$  are calculated independently,

yielding a pair of coupled equations. The simulation can be stepped forward in time once the approximation to the Hamiltonian has been substituted into the predictor corrector formula.

$$\frac{-i}{\hbar} \hat{H}^M \mathcal{I}_{k_x, k_y} = -\left(\frac{\hbar}{2m}\right) (4\pi^2) (k_x^2 + k_y^2) \mathcal{R}_{k_x, k_y} \quad (17)$$

$$\frac{-i}{\hbar} \hat{H}^M \mathcal{R}_{k_x, k_y} = \left(\frac{\hbar}{2m}\right) (4\pi^2) (k_x^2 + k_y^2) \mathcal{I}_{k_x, k_y} \quad (18)$$

### 2.3. Square Well, Potential Energy Operator

A square-well potential was chosen to test the effectiveness and relative performance of the simulations of both domains, when the particle interacts with an electrostatic potential. The particle is again given the mass of an electron. For the purposes of this demonstration, the chosen potential must be high enough to reflect most of the particle off the potential step, back into the region where the potential is zero.

$$i\hbar \frac{\partial}{\partial t} \Psi = \frac{-\hbar^2}{2m} \nabla^2 \Psi + V \Psi = \hat{H}^P \Psi \quad (19)$$

The lattice describing the potential must have the number of columns  $N_x$  and number of rows  $N_y$ . Before the simulations are started, it is helpful to choose a well boundary index constant  $j_{step}$  and potential step size  $V_{step}$ . Construction of the potential well will simply set the value of  $V_{x,y} = V_{step}$  if the row or column index is less than  $j_{step}$ ,  $V_{x,y} = V_{step}$  if the row index is greater than  $N_x - j_{step}$  or column index is greater than  $N_y - j_{step}$  and  $V_{x,y} = 0$  elsewhere.

#### 2.3.1. Position Domain

The application of the potential operator is straightforward in the position domain. The lattices representing the real and imaginary parts of the wave-function  $\Psi$  are multiplied entry-wise by the lattice representing the potential function. The simulation can be stepped forward in time once the approximation to the Hamiltonian has been substituted into the predictor corrector formula.

$$\frac{-i}{\hbar} \hat{H}^P I_{x_2, y_2} = \frac{\hbar}{2m} \nabla^2 R_{x_2, y_2} - \frac{V_{x_2, y_2}}{\hbar} R_{x_2, y_2} \quad (20)$$

$$\frac{-i}{\hbar} \hat{H}^P R_{x_2, y_2} = \frac{-\hbar}{2m} \nabla^2 I_{x_2, y_2} + \frac{V_{x_2, y_2}}{\hbar} I_{x_2, y_2} \quad (21)$$

#### 2.3.2. Momentum Domain

In the momentum domain, application of the potential operator transforms to the convolution operation,  $\mathcal{FT}\{V\Psi\} = \mathcal{V} * \Phi$ , where  $\mathcal{FT}\{V\} = \mathcal{V}$  and  $*$  represents convolution [5]. Rather than carry out this computationally expensive operation, the operator is applied in the position domain. This requires transforming back and forth between the position and momentum domains at every time-step. Use of the predictor-corrector complicates this procedure somewhat, because the potential operator must be applied at the predictor step and the corrector step.

For simplicity and readability the procedure below does not write the state functions  $\Psi$  and  $\Phi$  decomposed into their real and imaginary components.

First, the potential operator is applied in the position domain and transformed to the momentum domain.

$$\mathcal{FT}\{V\Psi\} = \mathcal{V} * \Phi \quad (22)$$

The kinetic energy operator is applied to  $\Phi$  and added to  $\mathcal{V} * \Phi$  to produce the predictor Hamiltonian  $\hat{H}_{pred}^M$ .

$$\frac{-i}{\hbar} \hat{H}_{pred}^M \Phi = -\left(\frac{\hbar}{2m}\right) (4\pi^2) (k_x^2 + k_y^2) \Phi + \frac{-i}{\hbar} \mathcal{V} * \Phi \quad (23)$$

Following the example of Equation (8) in Section 2.1.3, the predictor formula is applied to produce the predicted value  $\tilde{\Phi}$ .

$$\tilde{\Phi}_n = \Phi_{n-1} + \Delta t \left( \frac{55-i}{24} \frac{\hat{H}_{pred}^M}{\hbar} \Phi_{n-1} - \frac{59-i}{24} \frac{\hat{H}_{pred}^M}{\hbar} \Phi_{n-2} + \frac{37-i}{24} \frac{\hat{H}_{pred}^M}{\hbar} \Phi_{n-3} - \frac{9-i}{24} \frac{\hat{H}_{pred}^M}{\hbar} \Phi_{n-4} \right) \quad (24)$$

At the corrector step, apply only the kinetic energy operator to the predicted value  $\tilde{\Phi}$  to produce the corrector Hamiltonian  $\hat{H}_{corr}$ .

$$\frac{-i}{\hbar} \hat{H}_{corr}^M \tilde{\Phi} = - \left( \frac{\hbar}{2m} \right) (4\pi^2) (k_x^2 + k_y^2) \tilde{\Phi} \quad (25)$$

Following the example of Equation (9) in Section 2.1.3, the corrector formula is applied to produce the corrected value  $\Phi$ .

$$\Phi_n = \frac{12\Delta t}{25} \frac{-i}{\hbar} \hat{H}_{corr}^M \tilde{\Phi}_n + \frac{48}{25} \Phi_{n-1} - \frac{36}{25} \Phi_{n-2} + \frac{16}{25} \Phi_{n-3} - \frac{3}{25} \Phi_{n-4} \quad (26)$$

Both the corrected value  $\Phi_n$  and the predicted value  $\tilde{\Phi}_n$  are transformed back to  $\Psi_n$  and  $\tilde{\Psi}_n$ . The potential energy operator is applied to the predicted value  $\tilde{\Psi}_n$  and added to  $\Psi_n$ .

$$\Psi_n = \Psi_n + \frac{-i}{\hbar} V \tilde{\Psi}_n \quad (27)$$

### 3. Results

All simulations are run under the same constraints and initial conditions to illustrate similarities and differences in the particle's position over time. The free particle simulations show differences in position due to the differences in boundary conditions between the two domains, despite having the same initial conditions. The square well simulations will show strong agreement in position due to the boundary conditions and initial conditions being the same. Regarding precision, all simulations were shown to be accurate to 8 decimal places. This value is stable for the duration of the simulations and was measured by finding the difference between the current state's density function and 1.

The physical constants of electron mass  $m$ , electron charge  $q$  and Planck's constant  $\hbar$  are given in standard units. This helps scale the simulations to real-world dimensions, although the simulated particles are much larger than real electrons to show detail. For the square-well simulations, the electrostatic potential is set to +100.0 eV, although this is also expressed in standard units. The well boundary is established at 20 index units inside the lattice boundaries.

The row and column sizes are set to  $N_x = N_y = 256$  for both position- and momentum-domain simulations. This sets the lattice sizes at a power of two for the FFTW. The geometric origin is set to  $(x_c, y_c) = (128, 128)$ , in index units. The lattice step sizes are set to  $\Delta x = \Delta y = 2.0 \times 10^{-10}$  meters. The time step is set to  $\Delta t = 1.0 \times 10^{-18}$  seconds. Each simulation is run for 100,000 time steps. The lifetime of each simulation is 0.1 picoseconds. The components, densities and expectation values are measured and recorded every 500 time steps. This increment is referred to as a "frame" in the graphs below and each frame is equal to  $5.0 \times 10^{-16}$  seconds.

All initial condition parameters are given in index units and the actual parameters are multiplied by the appropriate lattice step size. The initial conditions set the particle on the +x-axis at index +70.0, relative to the origin point and close to the vertical boundary at the end of the +x-axis. The particle is given positive momentum, which will propel the particle into the boundary. The particle's spatial wavelength  $\lambda$  is set to 10.0 index units. The particle's predicted velocity can be found by the conservation of momentum.

$$\hbar k = mv \Rightarrow v = \frac{2\pi\hbar}{m\lambda} \quad (28)$$

The initial velocity of the particle in all simulations is  $3.64 \times 10^5$  meters per second in the +x-direction. The initial momentum  $k_x$  is therefore  $\pi \times 10^9$  kilogram-meters per second. In units normalized by  $2\pi$ , as shown in the momentum domain simulations, the initial momentum is  $5.0 \times 10^8$ .

### 3.1. Free Particle

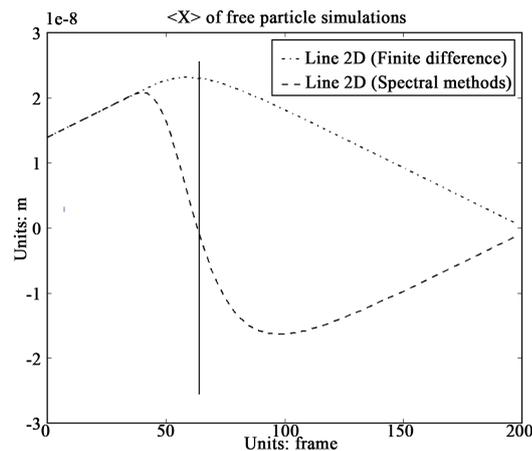
Based on the initial conditions and predicted velocity, the particle is expected to reach the boundary at time  $t = 3.19 \times 10^{-14}$  seconds, corresponding to the 64<sup>th</sup> frame of the simulation. The position-domain particle reflects off the boundary while the momentum-domain particle travels through the boundary and reappears in the opposite boundary, travelling in the same direction at the same velocity. **Figure 1** charts the relative position along the x-axis of the position- and momentum-domain particles. The position is given by the expectation value  $\langle X \rangle$ . A marker has been placed at the 64<sup>th</sup> frame to show where the paths are expected to diverge.

Four free-particle animations are produced for the position domain. One overhead view and one cross-sectional view are produced for the finite-difference free-particle simulation and one overhead view and one cross-sectional view are produced for the spectral free-particle simulation.

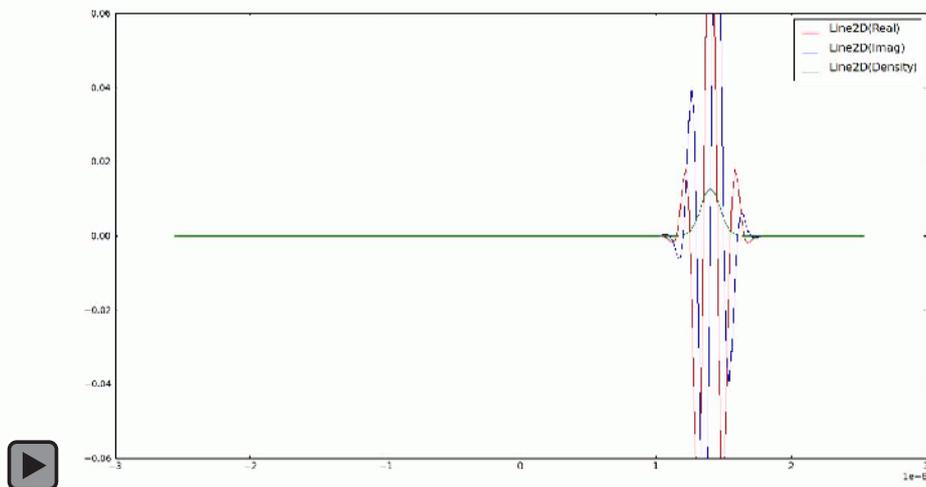
**Figure 2** shows the cross-sectional view of the free particle's components and density produced by finite-difference methods. The cross-section is along the x-axis in the position domain.

**Figure 3** shows the overhead view of the free particle's position-domain density produced by finite-difference methods. The particle diffracts as it reflects off the boundary.

**Figure 4** shows the cross-sectional view of the free particle's components and density produced by spectral methods. The cross-section is along the x-axis in the position domain.



**Figure 1.** Comparing  $\langle X \rangle$  of free particle produced by finite-difference and spectral methods.



**Figure 2.** Cross-sectional animation still at 64<sup>th</sup> frame of free particle simulation produced by finite-difference methods.

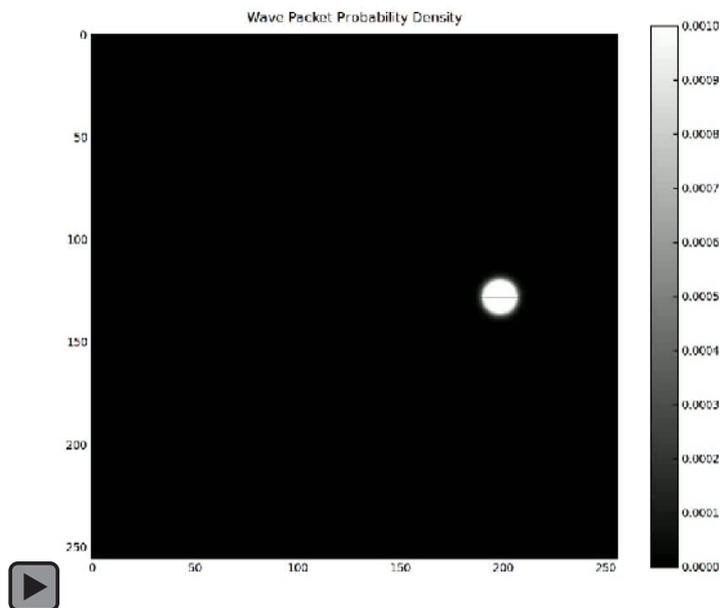


Figure 3. Overhead animation still at 64<sup>th</sup> frame of free particle simulation produced by finite-difference methods.

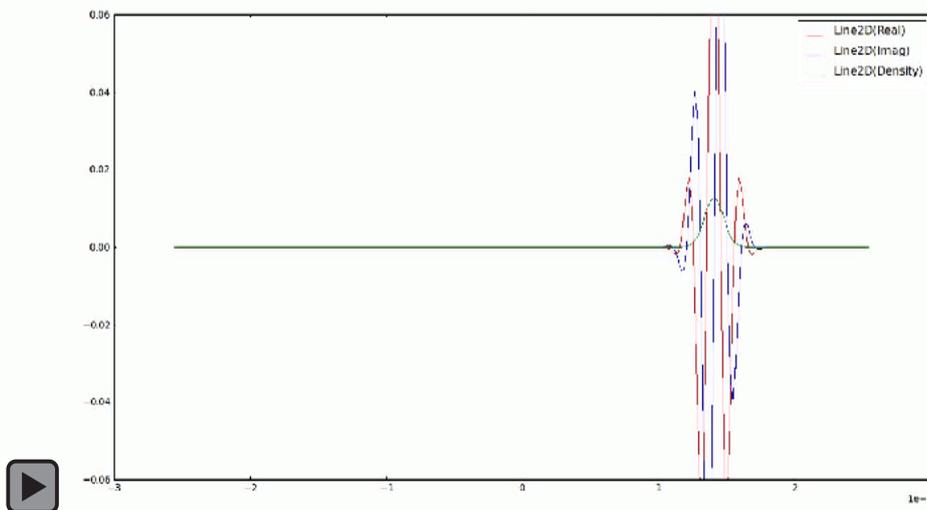


Figure 4. Cross-sectional animation still at 64<sup>th</sup> frame of free particle simulation produced by spectral methods.

Figure 5 shows the overhead view of the free particle’s position-domain density produced by spectral methods. The particle continues to diffuse in space as it pass through the boundary.

### 3.2. Square Well

Based on the initial conditions and predicted velocity, the particle is expected to reach the boundary at time  $t = 2.09 \times 10^{-14}$  seconds, corresponding to the 42<sup>nd</sup> frame of the simulation. The particles in both domains reflect off the boundary imposed by the electrostatic potential. Figure 6 charts the positions of the particles, given by the expectation value  $\langle X \rangle$ .

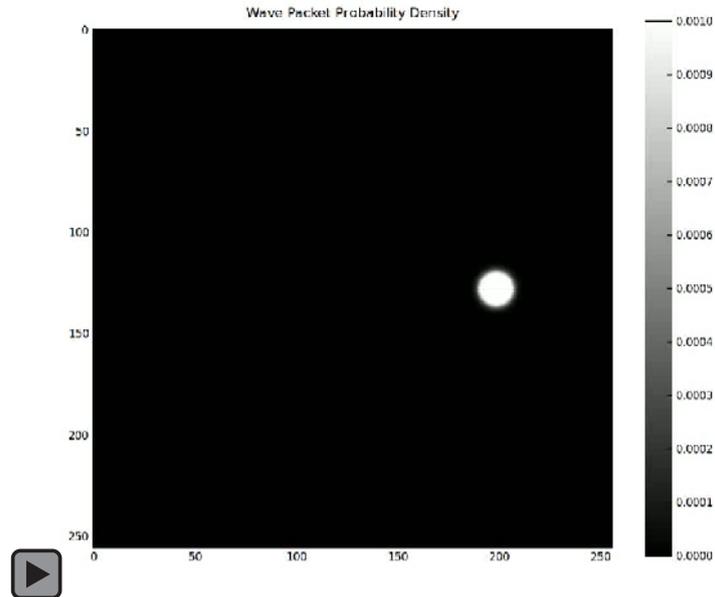


Figure 5. Overhead animation still at 64<sup>th</sup> frame of free particle simulation produced by spectral methods.

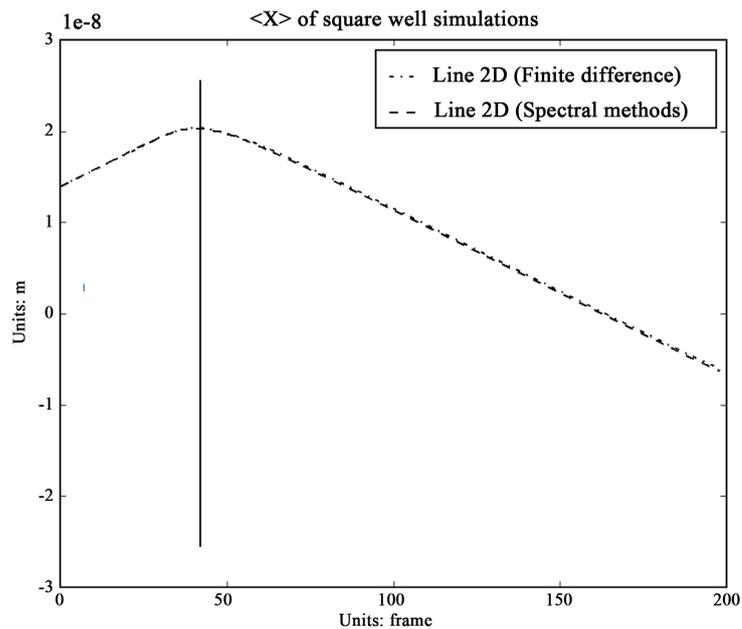


Figure 6. Comparing  $\langle X \rangle$  of bound particle in square well produced by finite-difference and spectral methods.

### 3.2.1. Position Domain

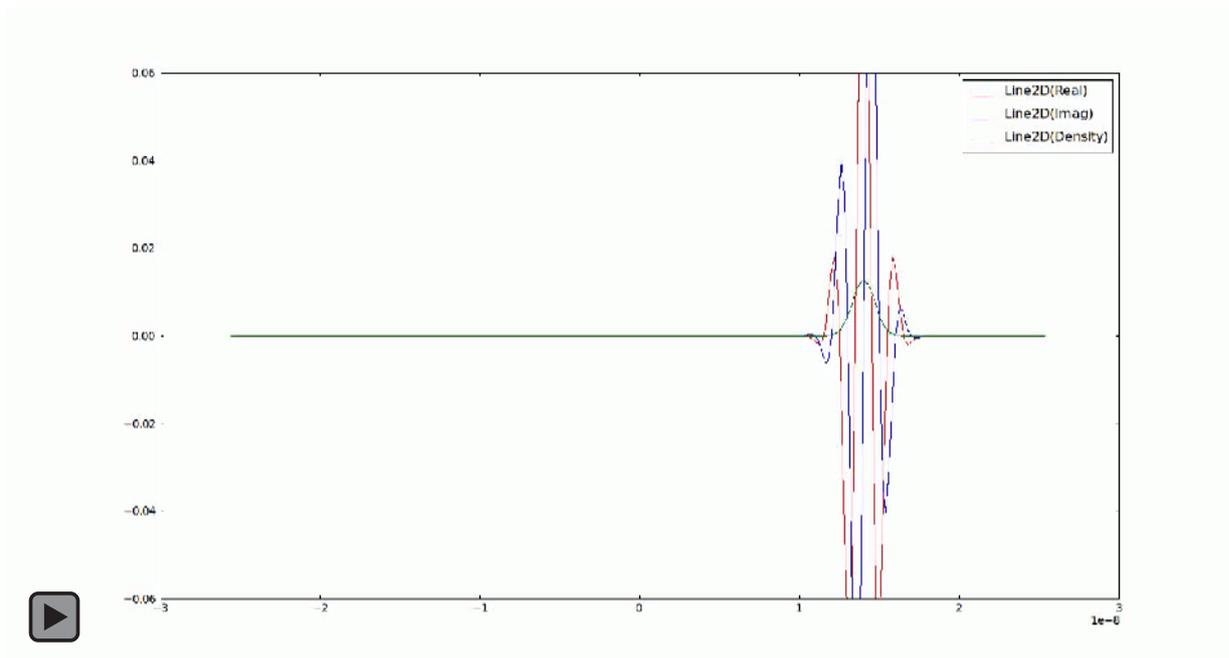
Four square-well animations are produced for the position domain. One overhead view and one cross-sectional view are produced for the finite-difference square-well simulation and one overhead view and one cross-sectional view are produced for the spectral square-well simulation.

Figure 7 shows the cross-sectional view of the bound particle's components and density produced by finite-difference methods. The cross-section is along the x-axis in the position domain.

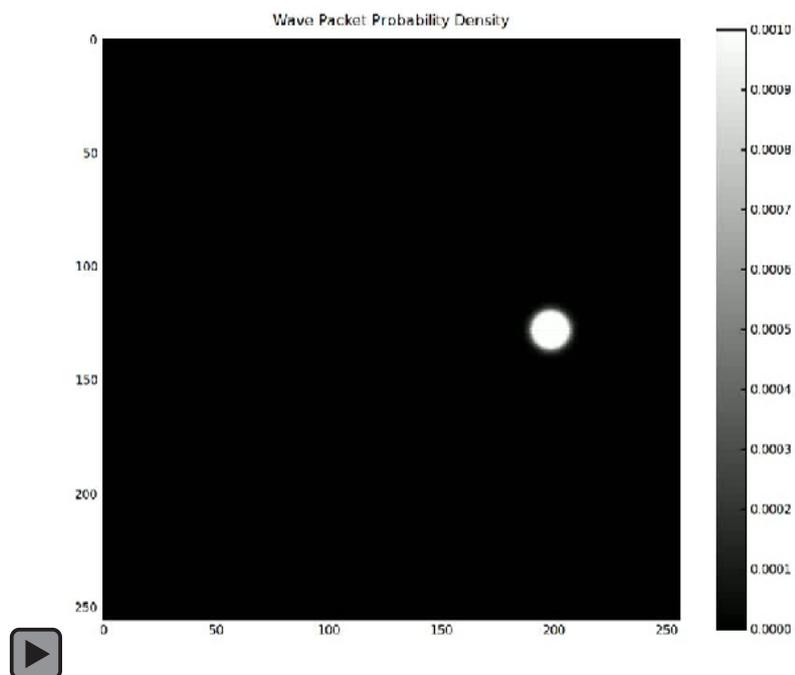
**Figure 8** shows the overhead view of the bound particle's position-domain density produced by finite-difference methods. The particle diffracts as it reflects off the potential wall.

**Figure 9** shows the cross-sectional view of the bound particle's components and density produced by spectral methods. The cross-section is along the x-axis in the position domain.

**Figure 10** shows the overhead view of the bound particle's position-domain density produced by spectral methods. The particle diffracts as it reflects off the potential wall.



**Figure 7.** Cross-sectional animation still at 42<sup>nd</sup> frame of square well simulation produced by finite-difference methods.



**Figure 8.** Overhead animation still at 42<sup>nd</sup> frame of square well simulation produced by finite-difference methods.

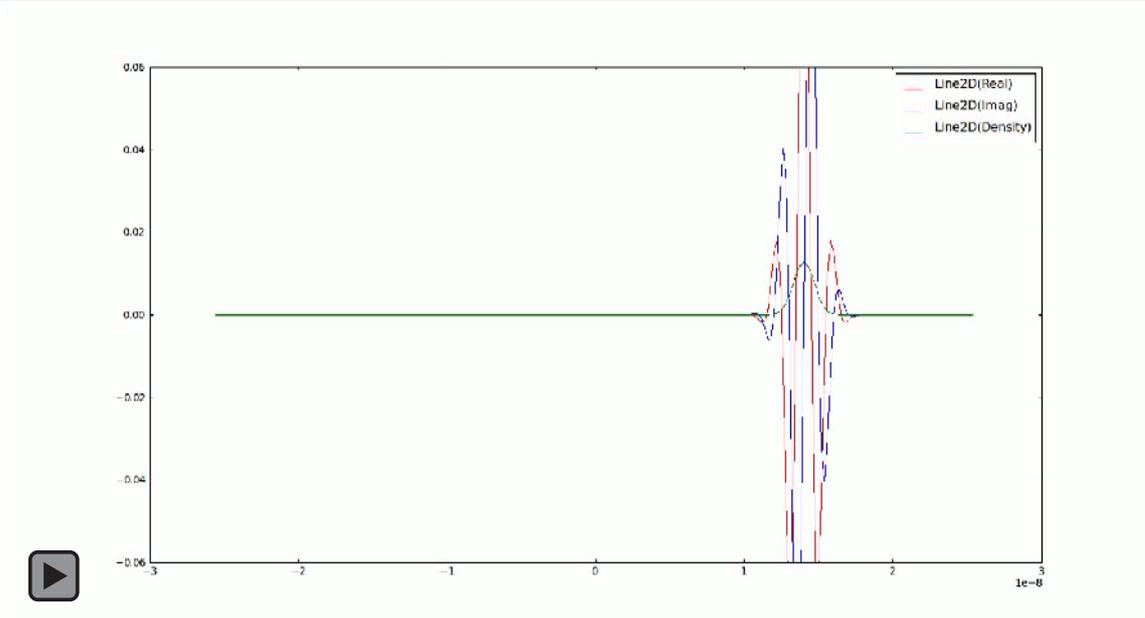


Figure 9. Cross-sectional animation still at 42<sup>nd</sup> frame of square well simulation produced by spectral methods.

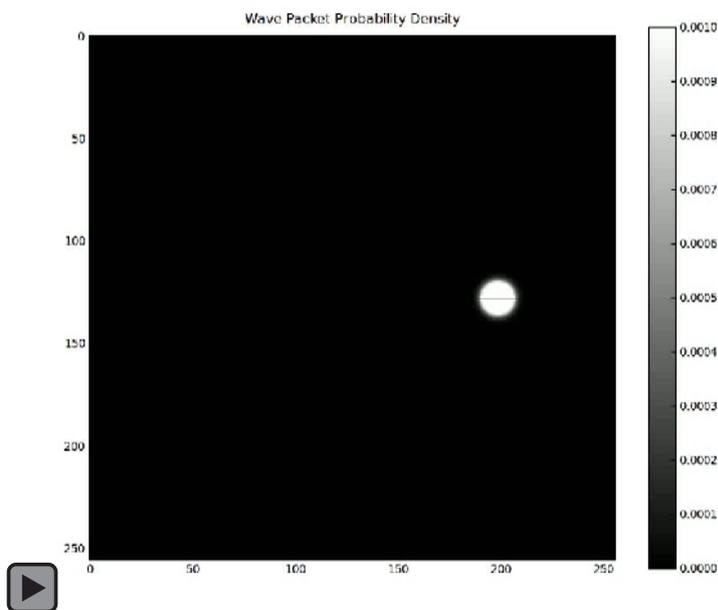


Figure 10. Overhead animation still at 42<sup>nd</sup> frame of square well simulation produced by spectral methods.

### 3.2.2. Momentum Domain

One additional animation is produced for the momentum-domain, square-well simulation that shows the cross-sectional view of the momentum density function. The cross-section is taken along the  $k_x$  axis. Figure 11 shows a cross-sectional view of the density function in the momentum domain at the 42<sup>nd</sup> frame. As the particle interacts with the electrostatic potential, the particle is reflected and reverses direction. In the momentum domain, this is indicated by the density function disappearing from the positive axis and reappearing on the negative axis.

Figure 12 illustrates this reversal of direction over time by measuring the expectation value  $\langle K_x \rangle$ . A marker has been placed at the 42<sup>nd</sup> frame indicating when the particle interacts with the boundary and reverses direction.

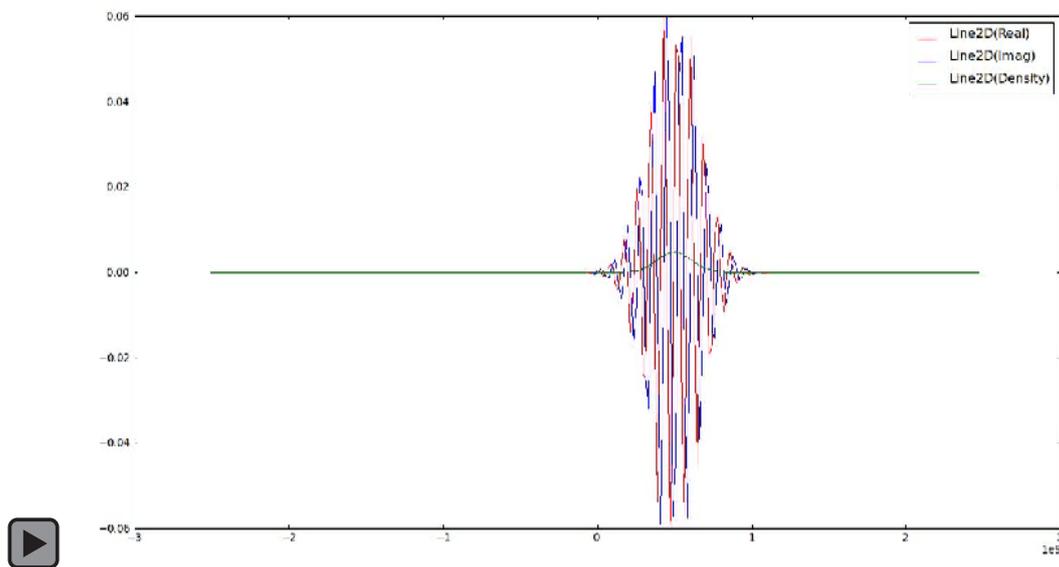


Figure 11. Cross-sectional, momentum-domain animation still at 42<sup>nd</sup> frame of square well simulation produced by spectral methods.

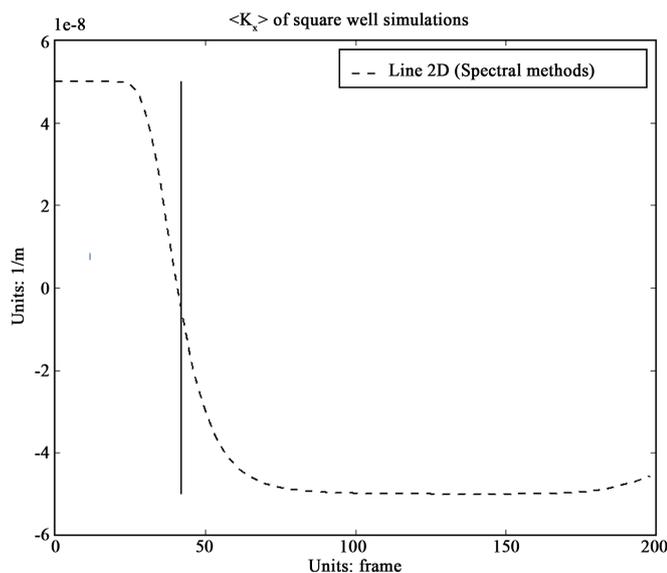


Figure 12.  $\langle K_x \rangle$  of bound particle in square well produced by spectral methods.

#### 4. Discussion

Momentum-domain simulations of the time-dependent Schrödinger equation provide precise and accurate results; however, the application of these techniques is not limited to the Schrödinger equation. The methods described in this paper are also suitable to simulate the heat equation  $\frac{\partial}{\partial t} f = \alpha \nabla^2 f$ , where  $f$  describes temperature in space and time and  $\alpha$  is the thermal diffusivity. The spectral methods described here may be applied to any parabolic differential equation. The spectral methods also transform the Hartree-Fock operation in a many-body problem from convolution in the position domain to entry-wise multiplication in the momentum domain, although this application is not explored here due to resource constraints.

These simulations were produced on single-core, AMD Athlon X2 processor. The spectral methods demonstrated faster performance for the free-particle simulation, while the finite difference methods demonstrated faster performance for the square-well simulation. None of the simulations employed parallel computing techniques due to the limitations of the hardware. Multiple cores would allow multiple Fourier transforms to be calculated at the same time. Because the spectral-method, square-well simulation requires multiple Fourier transforms at every time step, introducing parallel computing techniques would increase performance substantially.

## Acknowledgements

Thanks to Craig Unrein for proofreading assistance.

## References

- [1] Griffiths, D.J. (2005) Introduction to Quantum Mechanics. 2nd Edition, Prentice Hall, New Jersey.
- [2] Hamming, R.W. (1973) Numerical Methods for Scientists and Engineers. 2nd Edition, Dover, New York.
- [3] Kahaner, D., Moler, C. and Nash, S. (1989) Numerical Methods and Software. Prentice Hall, Upper Saddle River.
- [4] Moxley III, F.I., Zhu, F. and Dai, W. (2012) A Generalized FDTD Method with Absorbing Boundary Condition for Solving a Time-Dependent Linear Schrödinger Equation. *American Journal of Computational Mathematics*, **2**, 163-172. <http://dx.doi.org/10.4236/ajcm.2012.23022>
- [5] Bracewell, R.N. (2000) The Fourier Transform and Its Applications. 3rd Edition, McGraw-Hill, Boston.
- [6] Frigo, M. and Johnson, S.G. (2005) The Design and Implementation of FFTW3. *Proceedings of the IEEE*, **93**, 216-231. <http://dx.doi.org/10.1109/JPROC.2004.840301>
- [7] Askar, A. and Cakmak, A. (1978) Explicit Integration Method for the Time-Dependent Schrödinger Equation for Collision Problems. *Journal of Chemical Physics*, **68**, 2794-2798. <http://dx.doi.org/10.1063/1.436072>
- [8] Maestri, J.J., Landau, R.H. and Páez, M.J. (2000) Two-Particle Schrödinger Equation Animations of Wavepacket Wave-Packet Scattering. *American Journal of Physics*, **68**, 1113-1119. <http://dx.doi.org/10.1119/1.1286310>
- [9] Soriano, A., Navarro, E.A., Porti, J.A. and Such, V. (2004) Analysis of the Finite Difference Time Domain Technique to Solve the Schrödinger Equation for Quantum Devices. *Journal of Applied Physics*, **95**, 8011-8018. <http://dx.doi.org/10.1063/1.1753661>
- [10] Sullivan, D.M. (2012) Quantum Mechanics for Electrical Engineers. John Wiley and Sons, Hoboken. <http://dx.doi.org/10.1002/9781118169780>
- [11] Visscher, P. (1991) A Fast Explicit Algorithm for the Time-Dependent Schrödinger Equation. *Computers in Physics*, **5**, 596-598. <http://dx.doi.org/10.1063/1.168415>
- [12] Chen, Z.D., Zhang, J.Y. and Yu, Z.P. (2009) Solution of the Time-Dependent Schrödinger Equation with Absorbing Boundary Conditions. *Journal of Semiconductors*, **30**, 012001-1-012001-6.
- [13] Hunter, J.D. (2007) Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering*, **9**, 90-95. <http://dx.doi.org/10.1109/MCSE.2007.55>