High-Order Symplectic FDTD Scheme for Solving Time-Dependent Schrödinger Equation

Jing Shen[1,2], Wei E.I. Sha[3], Zhixiang Huang[1], Mingsheng Chen[2], Xianliang Wu[1,2]

1. Key Laboratory of Intelligent Computing & Signal Processing, Anhui University, Hefei 230039, China.
2. Department of Physics and Electronic Engineering, Hefei Normal University, Lianhua Road, Hefei 230601, China
3. Department of Electrical and Electronic Engineering, The University of Hong Kong, Pokfulam Road, Hong Kong.

Corresponding Authors: wsha@eee.hku.hk (W.E.I. Sha); xlwu@ahu.edu.cn (X.L. Wu)

Abstract—**Using the three-order symplectic integrators and fourth-order collocated spatial differences, a high-order symplectic finite-difference time-domain (SFDTD) scheme is proposed to solve the time-dependent Schrödinger equation. First, the high-order symplectic framework for discretizing Schrödinger equation is described. Then the numerical stability and dispersion analyses are provided for the FDTD(2,2), higher-order FDTD(2,4) and SFDTD(3,4) schemes. Next, to implement the Dirichlet boundary condition encountered in the quantum eigenvalue problem, the image theory and one-sided difference technique are manipulated particularly for high-order collocated differences. Finally, a detailed numerical study on 1D and 2D quantum eigenvalue problems is carried out. The simulation results of quantum wells and harmonic oscillators strongly confirm the advantages of the SFDTD scheme over the traditional FDTD method and other high-order approaches. The explicit SFDTD scheme, which is high-order-accurate and energy-conserving, is well suited for a long-term simulation and can save computer resources with large time step and coarse spatial grids.**

Keywords—**symplectic integrators; high-order collocated differences; Schrödinger equation; numerical stability and dispersion.**

Ⅰ. INTRODUCTION

Numerical solutions of Schrödinger equation have become increasingly important because of the tremendous demands for the design and optimization of nanodevices where quantum effects are significant or dominate [1]. The eigenvalue problem of Schrödinger equation is fundamentally important for the quantum transport and nanodevice modeling. The ballistic electron transport strongly depends on the transverse eigenstates of the conducting channel, which strongly relates to the nonequilbriium green's function [1]. Various intriguing phenomena related to the microscopic electron transport, such as the resonant tunneling effect [2], Fano-resonance [3], etc, are attributed to the excitation of eigenstates or interplay of different eigenstates. Thus, an accurate and efficient method to calculate the eigenstates and eigenfrequencies plays an important role in understanding the fundamental and device physics. Moreover, eigenstates and eigenfrequencies extraction tailored to industrial requirements is also indispensable in quantum computer aided design (CAD). One of commonly adopted methods to solve the time-dependent eigenvalue problem of Schrödinger equation is the FDTD method [4, 5], but it suffers from intolerable dispersion errors in a long term simulation.

A large quantity of physical phenomena can be modeled by Hamiltonian differential equations whose time evolution is the symplectic transform and flow conserves the symplectic structure [6-9]. The symplectic schemes include a variety of different temporal discretization strategies designed to preserve the global symplectic structure of the phase space for a Hamiltonian system. They have demonstrated their advantages in numerical computations for the Hamiltonian system, especially for a long-term simulation. The symplectic scheme has been successfully applied to solve Schrödinger equation with three different strategies. For the time-dependent Schrödinger equation, one scheme splits the complex wave function into real and imaginary parts [10-13], and another one decomposes the Hamiltonian into the kinetic and potential operators [13, 14]. For the time-independent Schrödinger equation, the symplectic scheme can also be employed if the generalized coordinate (complex wave function) and generalized velocity (spatial derivatives of complex wave function) are introduced [15, 16]. Moreover, the symplectic scheme has been extended to solve the nonlinear Schrödinger equation [17, 18]. Rigorously speaking, the symplectic scheme is particularly useful for solving Schrödinger equation satisfying the energy conservation law for the sum of kinetic and potential energy, which is different from Maxwell's equations dissipating the electromagnetic energy through the lossy materials [19].

In comparison with previous works, our work offers some new contributions described as follows: (1) a rigorous numerical stability and dispersion analyses of the high-order SFDTD scheme; (2) a comparative study of boundary treatments for infinite potential wells; (3) apply the high-order SFDTD scheme to the eigenvalue problem of Schrödinger equation and demonstrate its advantages comprehensively in terms of accuracy, convergence, and energy conservation.

The organization of the paper is given as follows. The general formulations of the high-order SFDTD scheme for discretizing the time-dependent Schrödinger equation are presented in Section 2. Analyses and comparisons of numerical stability and dispersion are described in Section 3. The boundary treatments for quantum eigenvalue problems are discussed in Section 4. The numerical results of 1D and 2D quantum wells and harmonic oscillators are shown in Section 5. Conclusion is summarized in Section 6.

II. Symplectic Framework of Schrödinger Equation

A. Traditional formulations

The time-dependent Schrödinger equation is given by [2]

$$i\hbar\frac{\partial\psi(\mathbf{r},t)}{\partial t}=-\frac{\hbar^2}{2m^*}\nabla^2\psi(\mathbf{r},t)+V(\mathbf{r})\psi(\mathbf{r},t) \tag{1}$$

where ψ is the wave function that is a probability amplitude in quantum mechanism describing the quantum state of a particle at position $\mathbf{r}$ and time t, m^* is the mass of the particle, $-\frac{\hbar^2}{2m^*}\nabla^2$ is the kinetic energy operator, $V(\mathbf{r})$ is the time-independent potential energy, and $-\frac{\hbar^2}{2m^*}\nabla^2+V$ is the Hamiltonian operator. To avoid using complex numbers, one can separate the variable $\psi(r,t)$ into its real and imaginary parts as

$$\psi(r,t)=\psi_R(r,t)+i\psi_I(r,t) \tag{2}$$

Inserting Eq. (2) into Eq. (1), we can get the following coupled set of equations [5]

$$\hbar\frac{\partial\psi_R(r,t)}{\partial t}=-\frac{\hbar^2}{2m^*}\left[\frac{\partial^2\psi_I(r,t)}{\partial x^2}+\frac{\partial^2\psi_I(r,t)}{\partial y^2}+\frac{\partial^2\psi_I(r,t)}{\partial z^2}\right]+V(r)\psi_I(r,t) \tag{3}$$

$$\hbar\frac{\partial\psi_I(r,t)}{\partial t}=\frac{\hbar^2}{2m^*}\left[\frac{\partial^2\psi_R(r,t)}{\partial x^2}+\frac{\partial^2\psi_R(r,t)}{\partial y^2}+\frac{\partial^2\psi_R(r,t)}{\partial z^2}\right]-V(r)\psi_R(r,t) \tag{4}$$

A mesh is defined in a discrete set of grid points that sample the wave function in space and time. The discretized real and imaginary parts of the wave function can be represented as

$$\psi_R(\mathbf{r},t)\approx\psi_R^n(i,j,k)=\psi_R(i\Delta_x,j\Delta_y,k\Delta_z,n\Delta_t) \tag{5}$$

$$\psi_I(\mathbf{r},t)\approx\psi_I^n(i,j,k)=\psi_I(i\Delta_x,j\Delta_y,k\Delta_z,n\Delta_t) \tag{6}$$

where Δ_x, Δ_y, and Δ_z are, respectively, the spatial steps in the x, y, and z coordinate directions, Δ_t is the time step, i, j, k and n are integers. The first-order time derivatives can be discretized by a second-order centered-difference scheme. The second-order Laplace operator in Eqs. (3) and (4) is discretized by using the second-order collocated difference, which distinguishes from the Yee (staggered) cell in the FDTD method of Maxwell's equations [20, 21]. As a result, the update equations of the real and imaginary parts of the wave function are of the forms

$$\begin{aligned}\psi_R^{n+1}\left(i,j,k\right)=\psi_R^{n}\left(i,j,k\right)-\frac{\Delta t}{\left(\Delta x\right)^2}\frac{\hbar}{2m^*}\left[\psi_I^{n+1/2}\left(i+1,j,k\right)-2\Psi_I^{n+1/2}\left(i,j,k\right)+\psi_I^{n+1/2}\left(i-1,j,k\right)\right]\\ -\frac{\Delta t}{\left(\Delta y\right)^2}\frac{\hbar}{2m^*}\left[\psi_I^{n+1/2}\left(i,j+1,k\right)-2\Psi_I^{n+1/2}\left(i,j,k\right)+\psi_I^{n+1/2}\left(i,j-1,k\right)\right]\\ -\frac{\Delta t}{\left(\Delta z\right)^2}\frac{\hbar}{2m^*}\left[\psi_I^{n+1/2}\left(i,j,k+1\right)-2\Psi_I^{n+1/2}\left(i,j,k\right)+\psi_I^{n+1/2}\left(i,j,k-1\right)\right]+\frac{V\left(i,j,k\right)\Delta t}{\hbar}\times\Psi_I^{n+1/2}\left(i,j,k\right)\end{aligned}\tag{7}$$

$$\begin{aligned}\psi_I^{n+1/2}\left(i,j,k\right)=\psi_I^{n-1/2}\left(i,j,k\right)+\frac{\Delta t}{\left(\Delta x\right)^2}\frac{\hbar}{2m^*}\left[\psi_R^{n}\left(i+1,j,k\right)-2\Psi_R^{n}\left(i,j,k\right)+\psi_R^{n}\left(i-1,j,k\right)\right]\\ +\frac{\Delta t}{\left(\Delta y\right)^2}\frac{\hbar}{2m^*}\left[\psi_R^{n}\left(i,j+1,k\right)-2\Psi_R^{n}\left(i,j,k\right)+\psi_R^{n}\left(i,j-1,k\right)\right]\\ +\frac{\Delta t}{\left(\Delta z\right)^2}\frac{\hbar}{2m^*}\left[\psi_R^{n}\left(i,j,k+1\right)-2\Psi_R^{n}\left(i,j,k\right)+\psi_R^{n}\left(i,j,k-1\right)\right]-\frac{V\left(i,j,k\right)\Delta t}{\hbar}\times\Psi_R^{n}\left(i,j,k\right)\end{aligned}\tag{8}$$

B. Symplectic scheme

A wave function of space and time evaluated at a discrete point in the collocated grid and at a discrete stage in the time step can be notated as

$$\psi(i,j,k)=\psi^{n+l/m}\left(i\Delta_x,j\Delta_y,k\Delta_z,(n+\tau_l)\Delta_t\right)\tag{9}$$

where $n+l/m$ denotes the lth stage after n time steps, m is the total stage number, and τ_l is the fixed time with respect to the lth stage.

For the spatial domain, the explicit qth-order-accurate collocated differences are used to discretize the second-order spatial derivatives, i.e.

$$\left(\frac{\partial^2\psi^{n+l/m}}{\partial\delta^2}\right)_h=\frac{1}{\Delta_\delta^2}\sum_{r=-q/2}^{q/2}W_r\psi^{n+l/m}\left(h+r\right)+O\left(\Delta_\delta^{q+1}\right)\tag{10}$$

where $\delta=x,y,z$, $h=i,j,k$, and W_r is the spatial difference coefficients as shown in Table I.

Table I. Coefficients of qth-order collocated differences

Order (q)	W_{-2}	W_{-1}	W_0	W_1	W_2
2		1	-2	1	
4	$-1/12$	$4/3$	$-5/2$	$4/3$	$-1/12$

With the help of Eqs. (3) and (4), the Schrödinger equation can be casted into a matrix form

$$\frac{\partial}{\partial t}\begin{pmatrix}\psi_R \\ \psi_I\end{pmatrix} = L\begin{pmatrix}\psi_R \\ \psi_I\end{pmatrix} = (A+B)\begin{pmatrix}\psi_R \\ \psi_I\end{pmatrix} \tag{11}$$

$$A = \begin{pmatrix}0 & K \\ 0 & 0\end{pmatrix} \qquad B = \begin{pmatrix}0 & 0 \\ -K & 0\end{pmatrix} \tag{12}$$

$$K = -\frac{\hbar}{2m^*}\left(\frac{\partial}{\partial x^2} + \frac{\partial}{\partial y^2} + \frac{\partial}{\partial z^2}\right) + \frac{V}{\hbar} \tag{13}$$

where $A^v = 0$ and $B^v = 0$ if $v \geq 2$. It is easy to proof that L in Eq. (11) is an asymmetric operator and therefore the exact solution of Schrödinger equation $\exp(Lt)$ is an orthogonal operator. In other words, the time evolution of Schrödinger equation essentially rotates the normalized wave function with a perfectly energy-conserving feature. Using the product of elementary symplectic mapping, the exact solution of (11) from $t = 0$ to $t = \Delta_t$ can be approximately reconstructed [22, 23]

$$\exp(\Delta_t(A+B)) = \prod_{l=1}^{m} \exp(d_l \Delta_t B)\exp(c_l \Delta_t A) + O(\Delta_t{}^{p+1}) = \prod_{l=1}^{m}(1 + d_l \Delta_t B)(1 + c_l \Delta_t A) + O(\Delta_t{}^{p+1}) \tag{14}$$

where c_l and d_l are the coefficients of symplectic integrators, and p is the order of the approximation. The symplectic integrators can satisfy the time-reversible condition or symmetric condition [11, 22, 24]. Here we use $m = 3$ and $p = 3$, thus a three-stage three-order explicit symplectic integrator is constructed [20]. The high-order SFDTD(3,4) scheme, which is third-order-accurate in time and fourth-order-accurate in space, can hold the energy-conserving property of Schrödinger equation without the amplitude error. The detailed update equation for the real part of the wave function at the lth stage can be written as

$$\begin{aligned}
\psi_R^{n+l/m}(i,j,k) &= \psi_R^{n+(l-1)/m}(i,j,k) + \frac{V(i,j,k)\Delta t}{\hbar} \times \Psi_I^{n+l/m}(i,j,k) \\
&- \alpha_{x1} \times \left[\psi_I^{n+l/m}(i+1,j,k) - 2\psi_I^{n+l/m}(i,j,k) + \psi_I^{n+l/m}(i-1,j,k)\right] \\
&- \alpha_{x2} \times \left[\psi_I^{n+l/m}(i+2,j,k) - 2\psi_I^{n+l/m}(i,j,k) + \psi_I^{n+l/m}(i-2j,k)\right] \\
&- \alpha_{y1} \times \left[\psi_I^{n+l/m}(i,j+1,k) - 2\psi_I^{n+l/m}(i,j,k) + \psi_I^{n+l/m}(i,j-1,k)\right] \\
&- \alpha_{y2} \times \left[\psi_I^{n+l/m}(i,j+2,k) - 2\psi_I^{n+l/m}(i,j,k) + \psi_I^{n+l/m}(i,j-2,k)\right] \\
&- \alpha_{z1} \times \left[\psi_I^{n+l/m}(i,j,k+1) - 2\psi_I^{n+l/m}(i,j,k) + \psi_I^{n+l/m}(i,j,k-1)\right] \\
&- \alpha_{z2} \times \left[\psi_I^{n+l/m}(i,j,k+2) - 2\psi_I^{n+l/m}(i,j,k) + \psi_I^{n+l/m}(i,j,k-2)\right]
\end{aligned} \tag{15}$$

$$\alpha_{x1} = \frac{4}{3}c_l \times S_x \qquad \alpha_{y1} = \frac{4}{3}c_l \times S_y \qquad \alpha_{z1} = \frac{4}{3}c_l \times S_z \tag{16}$$

$$\alpha_{x2} = \frac{-1}{12}c_l \times S_x \qquad \alpha_{y2} = \frac{-1}{12}c_l \times S_y \qquad \alpha_{z2} = \frac{-1}{12}c_l \times S_z \tag{17}$$

$$S_x = \frac{\hbar}{2m^*}\frac{\Delta_t}{\Delta^2{}_x} \qquad S_y = \frac{\hbar}{2m^*}\frac{\Delta_t}{\Delta^2{}_y} \qquad S_z = \frac{\hbar}{2m^*}\frac{\Delta_t}{\Delta^2{}_z} \tag{18}$$

Ⅲ. Numerical Stability and Dispersion Analyses

A. Stability analysis

According to the von Neumann stability method, the solution of the wave function can be represented as a superposition of plane-waves

$$\psi(x,y,z,t) = A_0 \exp\left(-j_0(i\Delta_x k_x + j\Delta_y k_y + k\Delta_z k_z)\right)$$

$$k_x = k_0 \sin\theta\cos\varphi\,,\ \ k_y = k_0 \sin\theta\sin\varphi\,,\ \ k_z = k_0 \cos\theta \tag{19}$$

where $k_0 = \frac{p_m}{\hbar}$ is the wave number, p_m is the momentum, and θ and φ are the spherical angles. The qth-order collocated differences are used to discretize the second-order spatial derivatives, i.e.

$$\frac{\partial^2\psi}{\partial z^2} \approx \sum_{r=-q/2}^{q/2} W_r \frac{\psi(i,j,k+r)}{\Delta_z^2} = \sum_{r=-q/2}^{q/2} W_r \frac{\exp(-j_0 r k_z \Delta_z)}{\Delta_z^2}\psi(i,j,k) = \eta_z \psi \tag{20}$$

where $\eta_z = \sum_{r=-q/2}^{q/2} W_r \frac{\exp(-j_0 r k_z \Delta_z)}{\Delta_z^2}$.

For simplicity, we consider a 1D Schrödinger equation with zero potential energy

$$\frac{\partial}{\partial t}\begin{pmatrix}\psi_R \\ \psi_I\end{pmatrix} = \begin{pmatrix} 0 & -\frac{\hbar}{2m^*}\frac{\partial^2}{\partial z^2} \\ \frac{\hbar}{2m^*}\frac{\partial^2}{\partial z^2} & 0 \end{pmatrix}\begin{pmatrix}\psi_R \\ \psi_I\end{pmatrix} \tag{21}$$

and corresponding spatial discretization form is given by

$$\frac{\partial}{\partial t}\begin{pmatrix}\psi_R \\ \psi_I\end{pmatrix} = \begin{pmatrix} 0 & -\frac{\hbar}{2m^*}\eta_z \\ \frac{\hbar}{2m^*}\eta_z & 0 \end{pmatrix}\begin{pmatrix}\psi_R \\ \psi_I\end{pmatrix} \tag{22}$$

It is trivial to access the discretized evolution matrix L^d with the high-order symplectic integration scheme

$$L^d = \begin{bmatrix} l_{11} & l_{12} \\ l_{21} & l_{22} \end{bmatrix} = \prod_{l=1}^{m}\begin{pmatrix} 1 & 0 \\ \frac{\hbar}{2m^*}\eta_z d_l \Delta_t & 1 \end{pmatrix}\begin{pmatrix} 1 & -\frac{\hbar}{2m^*}\eta_z c_l \Delta_t \\ 0 & 1 \end{pmatrix} \tag{23}$$

The eigenvalues λ of the evolution matrix satisfy the following eigen-equation

$$\lambda^2 - tr(L^d)\lambda + \det(L^d) = 0 \tag{24}$$

where $tr(L^d)$ and $\det(L^d)$ are the trace and determinant of the evolution matrix, respectively. Regarding that the discretized evolution matrix is a symplectic matrix with the determinant of 1. The eigen-equation then can be simplified as

$$\lambda^2 - tr(L^d)\lambda + 1 = 0 \tag{25}$$

and its solutions are $\lambda_{1,2} = \dfrac{tr(L^d) \pm j_0\sqrt{4-\left[tr(L^d)\right]^2}}{2}$. A stable algorithm requires $|\lambda_{1,2}|=1$, and thus $|tr(L^d)| \le 2$. Implementing terms of matrix multiplications, we can get

$$tr(L^d) = 2 + \sum_{l=1}^{m}(-1)^l g_l \left(\left(\frac{\hbar}{2m^*}\right)^2 \Delta_t^2 \eta_z^2\right)^l \tag{26}$$

$$g_l = \sum_{1\le i_1 \le j_1 < i_2 \le j_2 < \cdots < i_l \le j_l \le m} c_{i_1} d_{j_1} c_{i_2} d_{j_2} \cdots c_{i_l} d_{j_l} + \sum_{1\le i_1 < j_1 \le i_2 < j_2 \le \cdots \le i_l < j_l \le m} d_{i_1} c_{j_1} d_{i_2} c_{j_2} \cdots d_{i_l} c_{j_l} \tag{27}$$

The above results can be generalized to a 3D Schrödinger equation with zero potential energy, i.e.

$$tr(L^d) = 2 + \sum_{l=1}^{m}(-1)^l g_l \left(\left(\frac{\hbar}{2m^*}\right)^2 \Delta_t^2 (\eta_x + \eta_y + \eta_z)^2\right)^l \tag{28}$$

Finally we can get

$$\sqrt{\frac{\hbar}{m^*}\frac{\Delta_t}{\Delta_\delta{}^2}} \le CFL \tag{29}$$

where CFL is the Courant-Friedrichs-Levy (CFL) number. Table 2 lists the maximum stability (CFL number) of the traditional FDTD(2,2) method, FDTD(2,4) approach, and SFDTD(3,4) scheme. The symmetric symplectic integrators for the SFDTD(3,4) scheme is given as follows: $c_1 = 0.26833010$, $c_2 = -0.18799162$, $c_3 = 0.91966152$, and $d_l = c_{m-l+1}$ ($1 \le l \le m$).

Table II. The numerical stability for various algorithms. $d = 1,2,3$ is the dimension number.

Algorithm	**CFL Number**
FDTD(2,2)	$\mathbf{1/\sqrt{d}}$
FDTD(2,4)	$\mathbf{0.8660/\sqrt{d}}$
SFDTD(3,4)	$\mathbf{1.3019/\sqrt{d}}$

From the table, we can conclude that the spatial high-order collocated differences decrease the CFL number, which can be improved by the high-order symplectic integrators. Particularly, the stability of the SFDTD(3,4) scheme can go beyond that of the traditional FDTD(2,2) method through a careful optimization of symplectic integrators. An open question lies at what is the fundamental stability limit for the high-order symplectic scheme, which should be studied in future work.

B. Numerical dispersion analysis

The dispersion relation of free space photon described by Maxwell's equations is of form

$$\omega = c\,|\,\mathbf{k}_0\,| \tag{30}$$

where c is the speed of light and $\mathbf{k}_0 = (k_x, k_y, k_z)$ is the wave vector with the amplitude of k_0. Critically different from free space photon with the cone-shaped 3D dispersion relation, the dispersion relation of free electron is a paraboloid, i.e.

$$\omega = \left(\frac{\hbar}{2m^*}\right)|\,\mathbf{k}_0\,|^2 \tag{31}$$

Resembling to Maxwell's equations, we can define a dummy velocity of Schrödinger equation as $v_0 = \left(\frac{\hbar}{2m^*}\right)$, so Eq. (31) can be rewritten as

$$\omega = v_0\,|\,\mathbf{k}_0\,|^2 \tag{32}$$

According to the plane wave expansion and energy-conserving property of symplectic schemes, the dispersion relation of free electron can be written as

$$\omega\Delta_t = a\cos\left[tr\left(L^d\right)/2\right] \tag{33}$$

The relative error of phase velocity is given by

$$\eta = 20\log_{10}\left|\frac{v_p - v_0}{v_0}\right| \tag{34}$$

where $v_p = \frac{\omega}{k_0^2}$, and ω can be obtained by Eq. (33).

We set the stability criterion to be $S_\delta = \frac{\Delta t}{\Delta\delta^2}\frac{\hbar}{2m^*} = 0.125$ that is the maximum stability of the FDTD(2,4) approach. Fig. 1 shows the relative phase velocity error as a function of points per wavelength (PPW) for a plane wave traveling at $\theta = 0°$ and $\varphi = 0°$. The spatial resolution is set to be 7 points per wavelength. We redraw the relative error at $\theta = 30°$ versus the propagating angle φ as shown in Fig. 2. According to Figs. 1 and 2, the SFDTD(3,4) scheme and the FDTD(2,4) approach show lower numerical dispersion than the traditional FDTD(2,2) method. Accordingly, the high-order collocated differences allow coarser grids within a given error bound, which in turn results in shorter CPU time and less storage. Intriguingly, the high order symplectic integrators did not reduce the dispersion compared to the low-order symplectic integrators (second-order staggered time stepping strategy), which is quite different from the corresponding results of Maxwell's equations [19, 25].

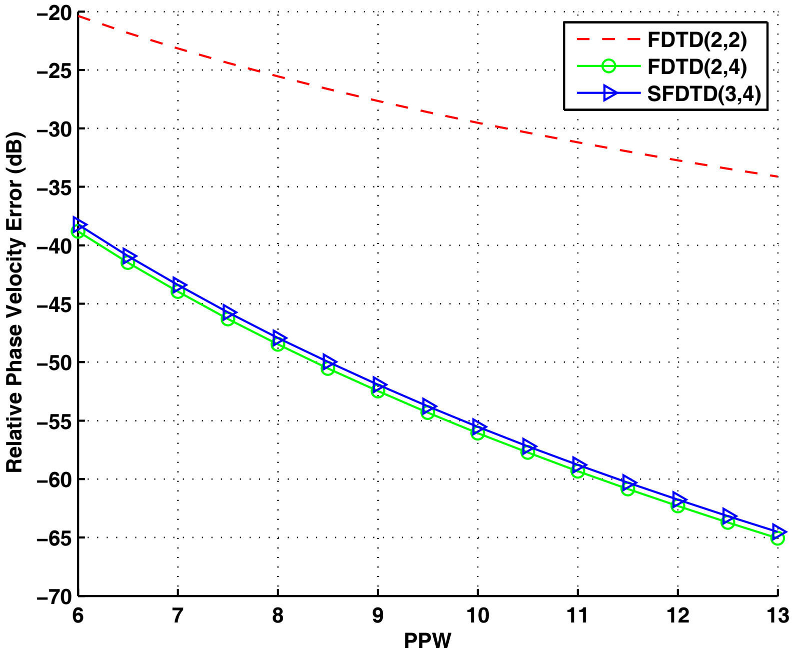

Fig. 1. The relative dispersion error as a function of the spatial resolution (points per wavelength). Here, the spherical angles are set to $\theta = 0^{\circ}$, $\varphi = 0^{\circ}$, and the stability constant is $S_{\delta} = \dfrac{\Delta t}{\Delta \delta^{2}} \dfrac{\hbar}{2m^{*}} = 0.125$.

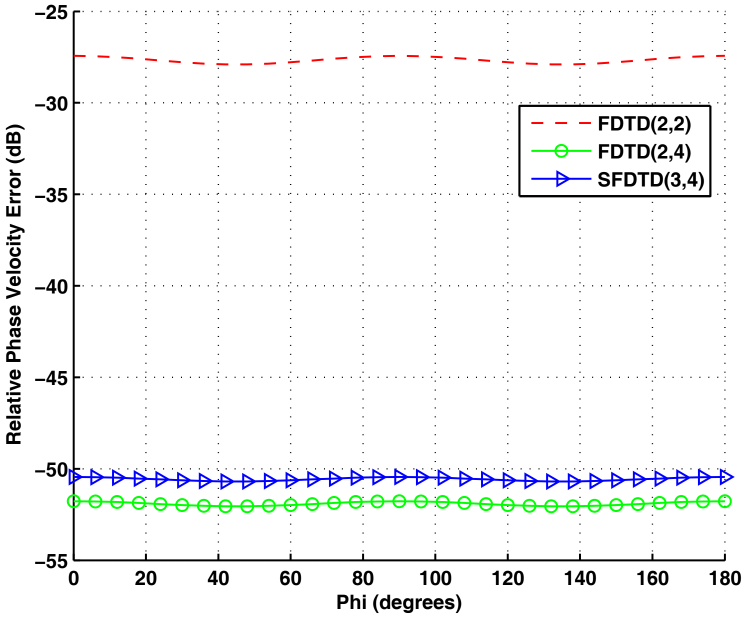

Fig. 2. The relative dispersion error as a function of the spherical angle φ. The spherical angle is $\theta = 30^\circ$, the spatial resolution is 7 points per wavelength, and the stability constant is $S_\delta = \frac{\Delta t}{\Delta\delta^2}\frac{\hbar}{2m^*} = 0.125$.

Ⅳ. Boundary Conditions

For solving the eigenvalue problem of Schrödinger equation, the Dirichlet boundary condition should be implemented properly to guarantee the simulation accuracy. Taking a one-dimensional quantum well as an example, we have $\psi(0) = 0$ and $\psi(L) = 0$ respectively for the left and right boundaries, where L is the length of the quantum well.

In view of high-order spatial differences, the image theory [26] and one-sided difference technique [27] adopted in Maxwell's equations can be naturally extended to Schrödinger equation. For the image theory, we impose the wave function at the left boundary as $\psi(0) = 0$, $\tilde{\psi}(1) = -\psi(1)$, and $\tilde{\psi}(2) = -\psi(2)$, where $\tilde{\psi}(1)$ and $\tilde{\psi}(2)$ correspond to the image points of $\psi(1)$ and $\psi(2)$ as illustrated in Fig. 3(a). For the right boundary, we have $\psi(L) = 0$, $\tilde{\psi}(L-1) = -\psi(L-1)$, and $\tilde{\psi}(L-2) = -\psi(L-2)$ as shown in Fig. 3(b).

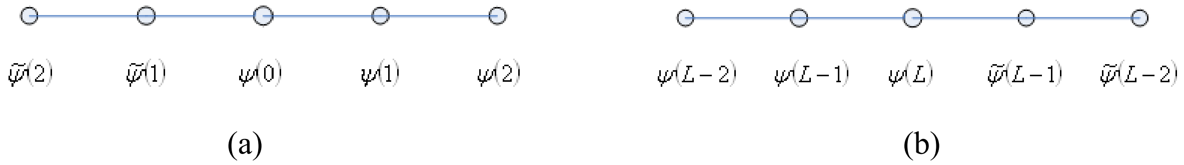

Fig. 3. A schematic pattern of the image theory method.

For the one-sided difference technique (Fig. 4), the high-order one-sided difference at the left boundary is given by

$$\left.\left(\frac{\partial^2\psi}{\partial\delta^2}\right)\right|_{i=1} = \frac{1}{12(\Delta\delta)^2}\left[10\psi(0) - 15\psi(1) - 4\psi(2) + 14\psi(3) - 6\psi(4) + \psi(5)\right] + O(\Delta_\delta^{q+1}) \tag{35}$$

Likewise, the one-sided difference at the right boundary is of form

$$\left.\left(\frac{\partial^2\psi}{\partial\delta^2}\right)\right|_{i=L-1} \approx \frac{-1}{12(\Delta\delta)^2}\left[10\psi(L) - 15\psi(L-1) - 4\psi(L-2) + 14\psi(L-3) - 6\psi(L-4) + \psi(L-5)\right] \tag{36}$$

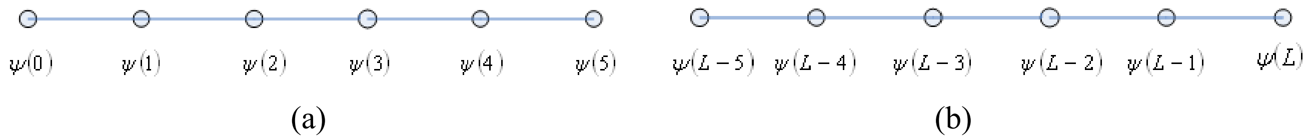

Fig. 4. A schematic pattern of the one-sided difference technique.

Ⅴ. Numerical Results

A. 1D Schrödinger equation

(a) A particle in a 1D box

The simplest form of a particle in a box model considers a one-dimensional system. Here, the particle only moves backward and forward along a straight line with impenetrable barriers at either end. The walls of a one-dimensional box may be visualized as regions of space with an infinitely large potential energy. The potential energy is

$$V(x)=\begin{cases}0, & 0<x<a\\ \infty, & otherwise\end{cases} \tag{37}$$

where a is the length of the box and x is the position of the particle within the box.

Regarding the simulation domain and cell size, they depend on the length of the box to be simulated and the highest eigenenergy of the particle of interest, respectively. Without loss of generality, we choose the domain to be $x\in[0,34]$. The cell size $\Delta x=1$, the stability constant in Eq. (18) is $S_\delta=\frac{\Delta t}{\Delta x^2}\frac{\hbar}{2m^*}=0.1$, and the iteration step $N_{\max}=41000$. The eigenenergies of the quantum well are quantized as

$$E_n=\frac{\hbar^2\pi^2}{2m^*a^2}n^2,\quad n=1,2,3,.. \tag{38}$$

In order to excite all possible modes, the delta source is located at the center of the box with two grids offset. Table Ⅲ lists the calculated eigenfrequencies. Compared with the analytical solution, SFDTD(3,4) scheme can achieve best accuracy.

Table Ⅲ. The eigenfrequency comparisons for a particle in a 1D infinite well.

Algorithm	**FDTD(2,2)**	**FDTD(2,4)**	**SFDTD(3,4)**	**Analytical**
ω_1	240	242	247	247
ω_2	981	983	998	998

ω_3	2220	2222	2227	2227

To further confirm the advantages of the SFDTD(3,4) scheme, the errors between the calculated eigenstates and the analytical solutions are depicted in Fig. 5. Obviously, the error associated with the SFDTD(3,4) scheme is significantly suppressed.

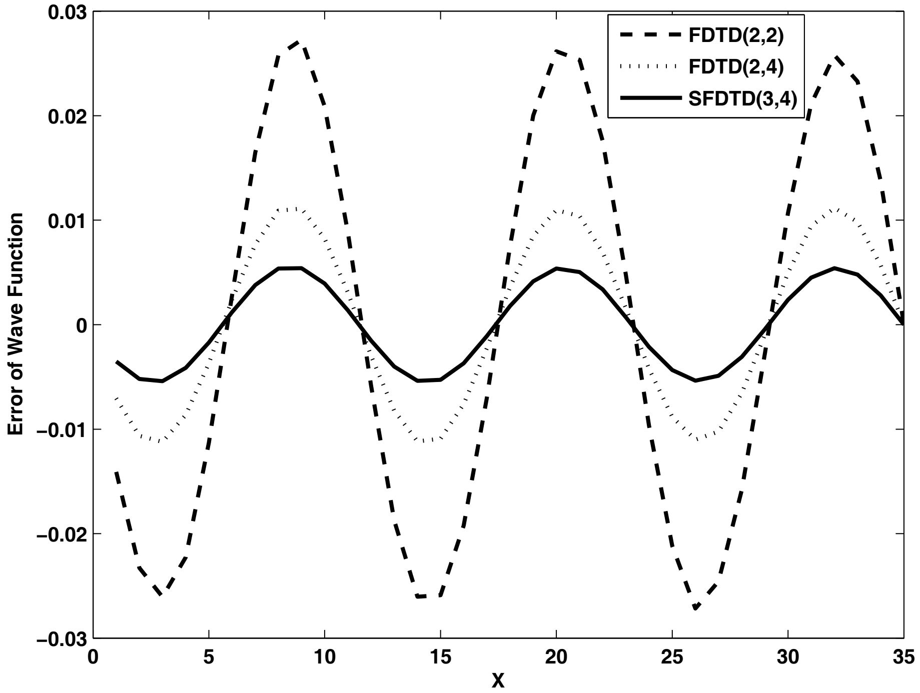

Fig. 5. The errors between the third eigenstate (the real part of the wave function) calculated and the analytical solution for a particle in a 1D infinite well.

Fig. 6 shows the L_2 norm error as a function of the grid size for the third eigenstate. Using the image theory to treat the impenetrable boundary, the high-order spatial differences indeed reduce the numerical error but achieve the same convergence rate compared with the low-order spatial difference method. Fig. 7 corresponds to the one-sided difference technique, where the high-order convergence rate can be clearly observed. By the aid of the high-order strategies applied both in space and time domains, the SFDTD(3,4) scheme achieves the best convergence rate.

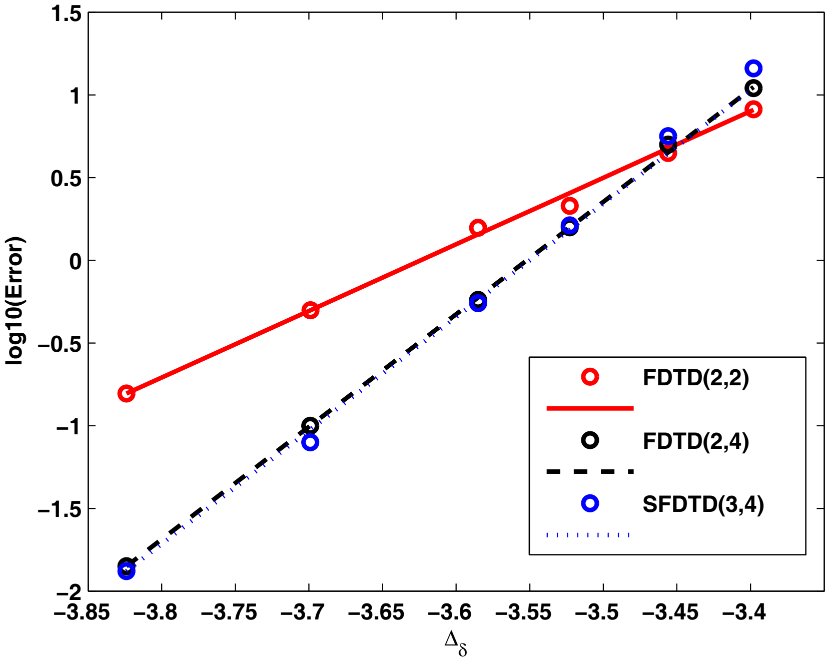

Fig. 6. The L_2 norm error (between the third eigenstate calculated and the analytical solution) as a function of the grid size. The image theory is adopted for boundary treatments.

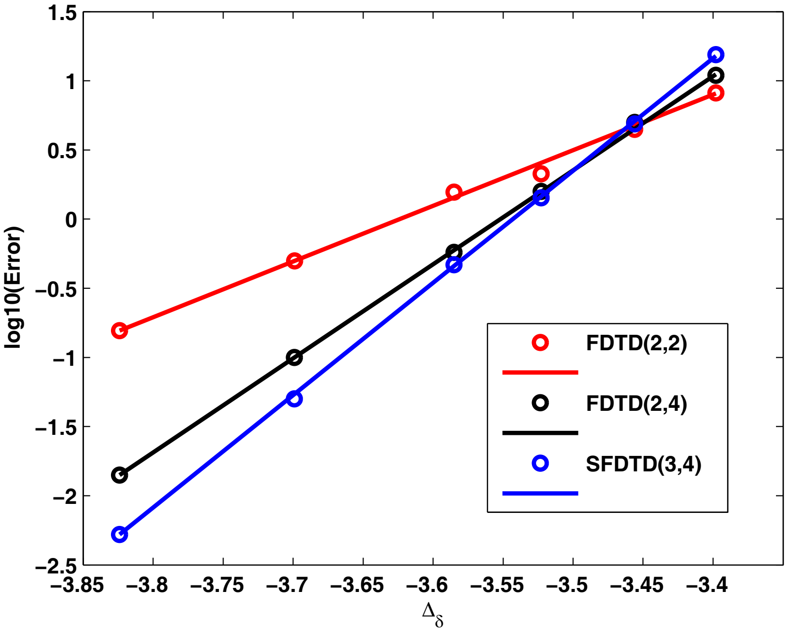

Fig. 7. The L_2 norm error (between the third eigenstate calculated and the analytical solution) as a function of the grid size. The one-sided difference technique is adopted for boundary treatments.

The energy-conserving property of Schrödinger equation determines the normalized condition of the wave function in long-term simulations. In order to testify the property, a Gaussian pulse is employed as the initial condition and the iteration step $N_{\max} = 10000$. Fig. 8 shows the wave function integral over the quantum well region by using various approaches. The SFDTD(3,4) scheme can hold the normalized condition of the wave function better.

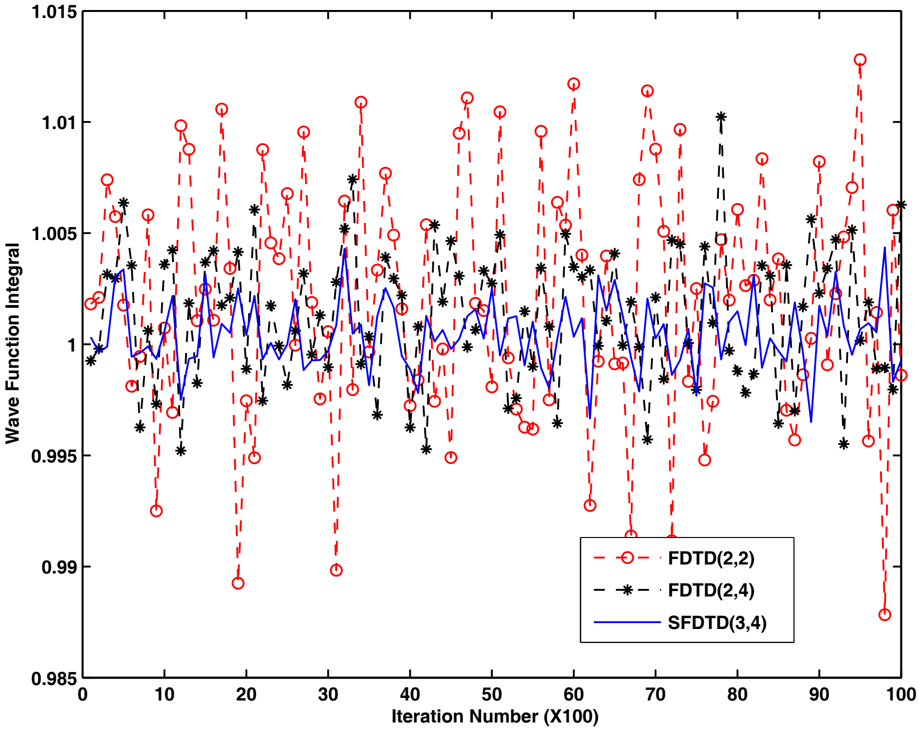

Fig. 8. The wave function integral over the quantum well region after 10000 iterations.

(b) 1D harmonic oscillator

The simulation domain is confined to be $x \in [0,34]$, $\Delta x = 1$, $S_\delta = \dfrac{\Delta t}{\Delta x^2}\dfrac{\hbar}{2m^*} = 0.1$, and $N_{\max} = 41000$. The one-sided difference technique is used for boundary treatments. The harmonic oscillator model is fundamentally important for the quantum simulation of nanodevices and its potential energy is given by

$$V(x) = \frac{1}{2}kx^2 \tag{39}$$

where $k = m\omega^2$. The eigenenergies of the harmonic oscillator are

$$E_n = \left(n + \frac{1}{2}\right)\hbar\omega \quad n = 0,1,2,3,... \tag{40}$$

The analytical wave function $\psi(x,t)$ can be written as

$$\psi_n(x,t) = \sqrt{\frac{\beta}{\sqrt{\pi}2^n n!}} e^{-x^2\beta^2/2} H_n(\beta x) \tag{41}$$

where $\beta = \sqrt{\frac{m^* \omega}{\hbar}}$ and H_n is the Hermite polynomial**.** Table IV and Fig. 9 respectively show the eigenfrequencies and the eigenstate errors of the harmonic oscillator. Likewise, the SFDFD(3,4) scheme shows good numerical performances for the eigenvalue problem of Schrödinger equation with a parabolic potential.

Table IV The eigenfrequency comparisons for a 1D harmonic oscillator model.

Algorithm	**FDTD(2,2)**	**FDTD(2,4)**	**SFDTD(3,4)**	**Analytical**
ω_1	1510	1513	1518	1518
ω_2	4545	4550	4554	4554
ω_3	7573	7582	7590	7590

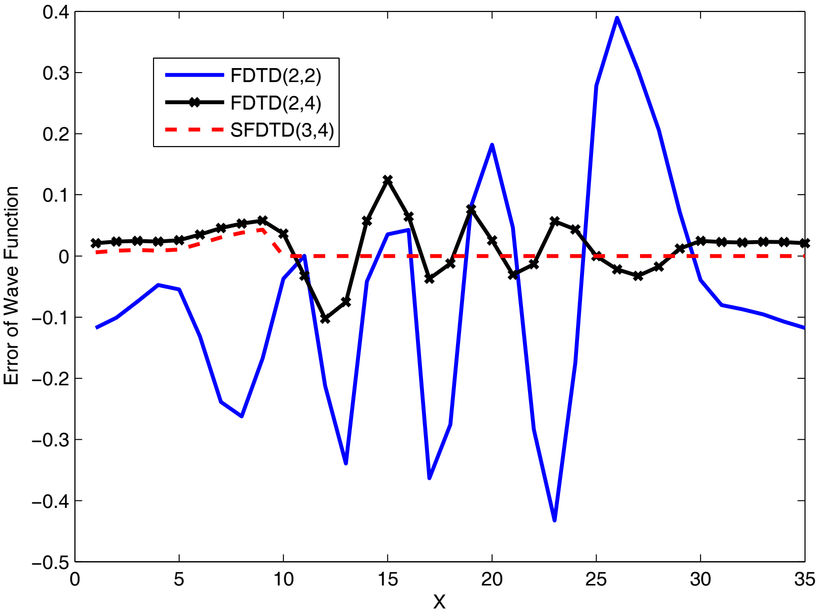

Fig. 9. The errors between the sixth eigenstate (the real part of the wave function) calculated and the analytical solution for a 1D harmonic oscillator.

B. 2D Schrödinger equation

(a) 2D quantum well

The simulation domain is set to $D = [0, \mathrm{L}_x] \times [0, \mathrm{L}_y] = [0,29] \times [0,29]$, $\Delta_\delta = 1$, $S_\delta = \frac{\Delta t}{\Delta \delta^2} \frac{\hbar}{2m^*} = 0.1$, and the iteration step $N_{\max} = 40000$. The one-sided difference technique is adopted to handle the boundaries of the infinite potential well. The eigenenergies can be trivially obtained

$$E_{n_x,n_y} = \frac{\hbar^2 \pi^2}{2m^*}\left(\frac{n_x^{\,2}}{L_x^2} + \frac{n_y^{\,2}}{L_y^2}\right) \tag{42}$$

where (n_x, n_y) is the mode number. Table V lists the eigenfrequencies and Fig. 10 shows the obatined eigenstate $\psi_{3,3}$. In contrast to the FDTD(2,2) and FDTD(2,4) approaches, the SFDTD(3,4) scheme could obtain more accurate results.

Table V The eigenfrequency comparisons for a 2D quantum well.

Algorithm	**FDTD(2,2)**	**FDTD(2,4)**	**SFDTD(3,4)**	**Analytical**
$\omega_{1,1}$	254	267	272	272
$\omega_{1,2}$	662	670	679	679
$\omega_{2,2}$	1072	1080	1088	1088
$\omega_{1,3}$	1366	1380	1388	1388

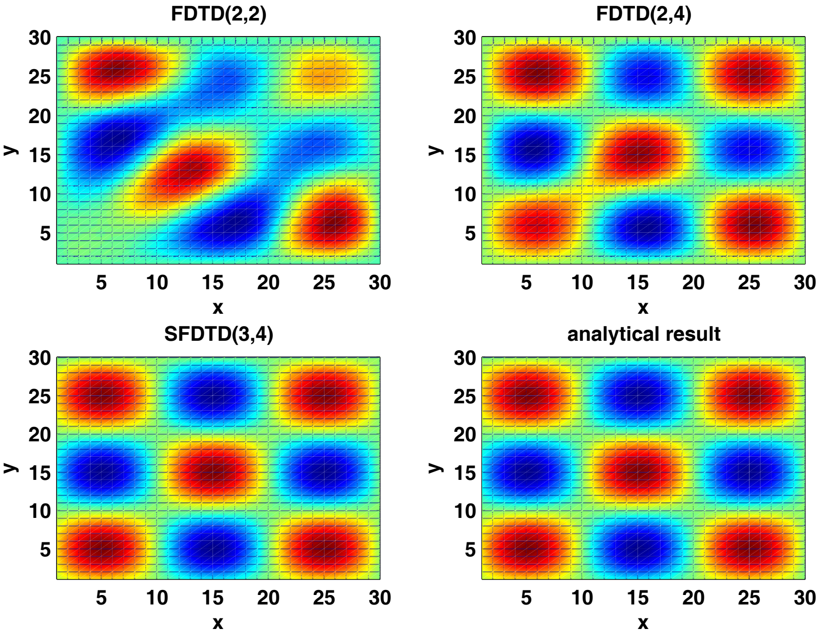

Fig. 10. The eigenstate (the real part of the wave function) corresponding to $n_x = 3$ and $n_y = 3$ for a 2D quantum well.

Fig. 11 shows the L_2 norm error of the eigenstate $\psi_{2,2}$ plotted as a function of the grid size. The error by the SFDTD(3,4) scheme decreases fastest with a reduced grid size.

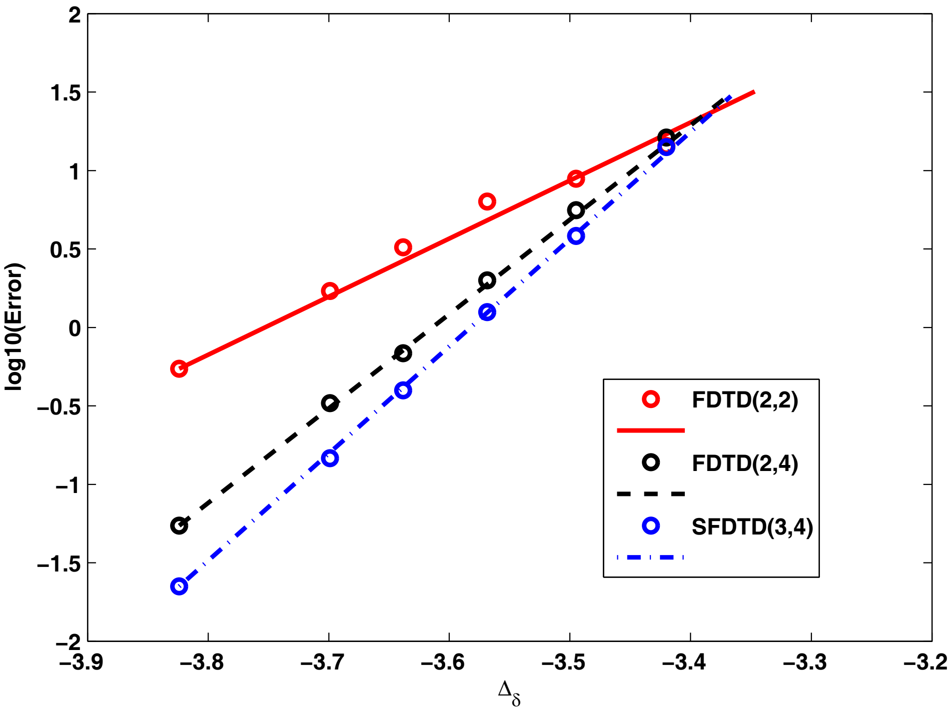

Fig. 11. The L_2 norm error (between the eigenstate $\psi_{2,2}$ calculated and the analytical solution) as a function of the grid size. The one-sided difference technique is adopted for boundary treatments.

(b) 2D harmonic oscillator

For a 2D harmonic oscillator, the potential is taken as $V(x,y)=\frac{1}{2}k(x^2+y^2)$. The other parameters are set to be the same as those in the 2D quantum well. The time-dependent solution of the wave function is

$$\psi(x,y,t)=\sqrt{\frac{\beta^2}{\pi 2^{n_x} n_x! 2^{n_y} n_y!}}e^{-(x^2+y^2)\beta^2/2}H_{n_x}(\beta x)H_{n_y}(\beta y) \tag{43}$$

$$E_{n_x,n_y}=\left(n_x+n_y+1\right)\hbar\omega \tag{44}$$

The eigenfrequencies of the 2D harmonic oscillator are calculated by the FDTD(2,2), FDTD(2,4), and SFDTD(3,4) methods (See Table VI). The SFDTD(3,4) scheme shows very significant advantages under the complicated potential energy condition, which is always met in the quantum device simulation.

Table Ⅵ The eigenfrequency comparisons for a 2D harmonic oscillator model.

Algorithm	FDTD(2,2)	FDTD(2,4)	SFDTD(3,4)	Analytical
$\omega_{1,1}$	352	363	360	360
$\omega_{1,2}$	773	788	785	785
$\omega_{2,2}$	1210	1218	1189	1189

VI. CONCLUSION

The high-order SFDTD(3,4) scheme, which is fourth-order accurate in space and third-order accurate in time, is energy-conserving and conditionally stable. On one hand, the scheme can achieve high-order accuracy by using both the fourth-order spatial collocated difference and the one-sided difference technique. On the other hand, incorporating the symplectic integrators, the scheme demonstrates desirable numerical performances under a long-term simulation. In this work, we studied the numerical stability and dispersion of the SFDTD(3,4) scheme and successfully applied it to the analyses of the eigenvalue problem of Schrödinger equation. Our numerical results validate significant advantages of the SFDTD(3,4) scheme in terms of accuracy, converge, and energy conservation. The work is fundamentally important for the quantum device simulation. The simple source code for implementing the SFDTD(3,4) scheme can be found in the Appendix.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China (No. 60931002，No.61101064, No.61001033, No. 61201122). This work was also funded by Provincial Natural Science Research Project of Anhui Colleges (No. KJ2011A242, No. KJ2011A240) and Provincial Natural Science Foundation of Anhui (No. 10040606Q51, No. 2012SQRL159).

REFERENCES

[1] S. Datta, Quantum Transport: Atom to Transistor, (Cambridge University Press, New York, 2005).

[2] D.J. Griffiths, Introduction to Quantum Mechanics, Second Edition (Addison-Wesley, Boston, 2004).

[3] Y.S. Joe, A.M. Satanin, C.S. Kim, Phys. Scr., 74 (2006) 259-266.

[4] A. Soriano, E.A. Navarro, J.A. Porti, V. Such, J. Appl. Phys., 95 (2004) 8011-8018.

[5] D.M. Sullivan, D.S. Citrin, J. Appl. Phys., 97 (2005).

[6] J.M. Sanz-Serna, M.P. Calvo, Numerical Hamiltonian Problems, (Chapman & Hall, London, 1994).

[7] Y. Sun, P.S.P. Tse, J. Comput. Phys., 230 (2011) 2076-2094.

[8] T. Monovasilis, Z. Kalogiratou, T.E. Simos, Comput. Phys. Commun., 181 (2010) 1251-1254.

[9] Z.X. Chen, X. You, W. Shi, Z.L. Liu, Comput. Phys. Commun., 183 (2012) 86-98.

[10] W.S. Zhu, X.S. Zhao, Y.Q. Tang, J. Chem. Phys., 104 (1996) 2275-2286.

[11] S.K. Gray, D.E. Manolopoulos, J. Chem. Phys., 104 (1996) 7099-7112.

[12] X.Y. Liu, P.Z. Ding, J.L. Hong, L.J. Wang, Comput. Math. Appl., 50 (2005) 637-644.

[13] S. Blanes, F. Casas, A. Murua, J. Chem. Phys., 124 (2006).

[14] S.A. Chin, C.R. Chen, J. Chem. Phys., 117 (2002) 1409-1415.

[15] X.S. Liu, X.Y. Liu, Z.Y. Zhou, P.Z. Ding, S.F. Pan, Int. J. Quantum Chem., 79 (2000) 343-349.

[16] T. Monovasilis, Z. Kalogiratou, T.E. Simos, Phys. Lett. A, 372 (2008) 569-573.

[17] A.L. Islas, D.A. Karpeev, C.M. Schober, J. Comput. Phys., 173 (2001) 116-148.

[18] T.C. Wang, T. Nie, L.M. Zhang, J. Comput. Appl. Math., 231 (2009) 745-759.

[19] W.E.I. Sha, Z.X. Huang, X.H. Wu, M.S. Chen, J. Comput. Phys., 225 (2007) 33-50.

[20] D.M. Sullivan, Electromagnetic Simulation Using the FDTD Method, (IEEE Press, New York, 2000).

[21] A. Taflove, S.C. Hagness, Computational Electrodynamics: the Finite-Difference Time-Domain Method, Third Edition, (Artech House, Boston, 2005).

[22] H. Yoshida, Phys. Lett. A, 150 (1990) 262-268.

[23] T. Hirono, W. Lui, S. Seki, Y. Yoshikuni, IEEE Trans. Microw. Theory Tech., 49 (2001) 1640-1648.

[24] W.E.I. Sha, Z.X. Huang, M.S. Chen, X.L. Wu, IEEE Trans. Antennas Propag., 56 (2008) 493-500.

[25] X.G. Ren, Z.X. Huang, X.L. Wu, S.L. Lu, H. Wang, L. Wu, S. Li, Comput. Phys. Commun., 183 (2012) 1192-1200.

[26] Q.S. Cao, Y.C. Chen, R. Mittra, IEEE Trans. Microw. Theory Tech., 50 (2002) 1578-1589.

[27] A. Yefet, P.G. Petropoulos, J. Comput. Phys., 168 (2001) 286-315.

APPENDIX

A simple C++ code for implementing the SFDTD(3,4) scheme is given as follows:

```
/*1D SFDTD program for Schrodinger equation simulation*/
#include <math.h>
#include <stdlib.h>
#include <stdio.h>
#define KE 14
const int CountFreq=1200;
double SweepFreq[CountFreq];
double Real_Point[CountFreq][KE];
double Imag_Point[CountFreq][KE];
double Real_Ref[CountFreq];
double Imag_Ref[CountFreq];
int main()
{
double psi_rl[KE],psi_im[KE];
int n,k,kc,ke,kstart,kcenter,NSTEPS,n_pml;
double P;
double pi,melec,hbar;
double ddx,dt,ra;
FILE *fp;
double lap_rl,lap_im,ke_rl,ke_im,kine,PE,s1,s2;
double RealTime=0;
double Fourier_Frq=0;
int TIMESTAGE=3; //number of stages
for(k=0;k<KE;k++)
{
psi_rl[k]=0.;
psi_im[k]=0.;
}
pi=3.141592653589793;
melec=9.10938188e-31;
hbar=1.05457148e-34;
double const1=8637.993775443278; //m_elec/hbar
ddx=2.5*pow(10.0,-4.0);
double stab=0.5;
dt=stab*(const1)*pow(ddx,2.0);
ra=(dt/(ddx*ddx))*(hbar/(2*melec));
int Pos=int(KE/2+3);
psi_rl[Pos]=1;
psi_im[Pos]=0;
double ra1[3]={0,0,0};
double ra2[3]={0,0,0};
//Symplectic integrator
double
D[3]={0.91966152301739985705089763815343,-0.18799161879915978200785286807 89,0.26833009578175992495695522992 55};
double
C[3]={0.26833009578175992495695522992 55,-0.18799161879915978200785286807 89,0.91966152301739985705089763815343};
double TimeTableH[3]={C[0],C[0]+C[1],C[0]+C[1]+C[2]}; //time table of H-field for SFDTD
const double TimeTableE[3]={0,D[0],D[0]+D[1]}; //time table of E-field for SFDTD
int i,w;
//Frequency
for (w=0;w<CountFreq; w++)
{
SweepFreq[w]=1+0.5*w;
}
// Fourier transform initial
for (w=0; w<CountFreq; w++)
{
Real_Ref[w]=0.;
Imag_Ref[w]=0.;
for ( int j=0; j<KE; j++)
{
Real_Point[w][j]=0.;
Imag_Point[w][j]=0.;
```

```
}
}
NSTEPS=41000; //time steps
for (n=0;n<=NSTEPS;n++)
{
for (int i=0; i<TIMESTAGE; i++)
{
printf("%d\n",n);
s1=C[i]*dt; //time coef
//boundary treatment (one-sided difference)
psi_rl[1]=exp(-s1)*psi_rl[1]-1.0/12*ra*(1-exp(-s1))/(s1)*(psi_im[5]-6*psi_im[4]+14*psi_im[3]-4*psi_im[2]-15*psi_im[1]);
psi_rl[KE-2]=exp(-s1)*psi_rl[KE-2]-1.0/12*ra*(1-exp(-s1))/(s1)*(psi_im[KE-6]-6*psi_im[KE-5]+14*psi_im[KE-4]-4*psi_im[KE-3]-15*psi_im[KE-2]);
//update equation for real part of wave function
for (k=2;k<KE-2;k++)
{
psi_rl[k]=exp(-s1)*psi_rl[k]-1.0/12*ra*(1-exp(-s1))/(s1)*((-1)*psi_im[k-2]+16*psi_im[k-1]-\
30*psi_im[k]+16*psi_im[k+1]-1*psi_im[k+2]);
}
//boundary condition
psi_rl[0]=0.;
psi_rl[KE-1]=0.;
s2=D[i]*dt; //time coef
//boundary treatment (one-sided difference)
psi_im[1]=exp(-s2)*psi_im[1]+1.0/12*ra*(1-exp(-s2))/(s2)*(psi_rl[5]-6*psi_rl[4]+14*psi_rl[3]-4*psi_rl[2]-15*psi_rl[1]);
psi_im[KE-2]=exp(-s2)*psi_im[KE-2]+1.0/12*ra*(1-exp(-s2))/(s2)*(psi_rl[KE-6]-6*psi_rl[KE-5]+14*psi_rl[KE-4]-4*psi_rl[KE-3]-15*psi_rl[KE-2]);
//update equation for imaginary part of wave function
for (k=2;k<KE-2;k++)
{
psi_im[k]=exp(-s2)*psi_im[k]+1.0/12*ra*(1-exp(-s2))/(s2)*((-1)*psi_rl[k-2]+16*psi_rl[k-1]-\
30*psi_rl[k]+16*psi_rl[k+1]-1*psi_rl[k+2]);
}
//boundary condition
psi_im[0]=0.;
psi_im[KE-1]=0.;
RealTime=(n+TimeTableH[i])*dt; //real time
//Fourier transform
for (w=0; w<CountFreq; w++)
{
Fourier_Frq=SweepFreq[w];
Real_Ref[w]=Real_Ref[w]+cos(Fourier_Frq*RealTime)*psi_rl[Pos];
Imag_Ref[w]=Imag_Ref[w]-sin(Fourier_Frq*RealTime)*psi_rl[Pos];
for (i=1; i<KE-1; i++)
{
```

```
Real_Point[w][i]=Real_Point[w][i]+cos(Fourier_Frq*RealTime)*psi_rl[i];
Imag_Point[w][i]=Imag_Point[w][i]-sin(Fourier_Frq*RealTime)*psi_rl[i];
}
}
}
}
// save results
fp=fopen("power_SFDTD.txt","w");
for (w=0; w<CountFreq; w++)
{
P=0.;
for (i=1; i<KE-1; i++)
{
P=P+pow(Real_Point[w][i],2.0)+\
pow(Imag_Point[w][i],2.0);
}
fprintf(fp,"%lf %lf\n",SweepFreq[w],P);
}
fclose(fp);
for (k=0;k<KE;k++)
{
printf("%d %lf %lf\n",k,psi_rl[k],psi_im[k]);
}
fp=fopen("prl_SFDTD.txt","w");
for (k=0;k<KE;k++)
{
fprintf(fp,"%lf\n",psi_rl[k]);
}
fclose(fp);
fp=fopen("pim_SFDTD.txt","w");
for (k=0;k<KE;k++)
{
fprintf(fp,"%lf\n",psi_im[k]);
}
fclose(fp);
fp=fopen("eigenfunction_SFDTD.txt","w");
w=937;
for (i=0; i<KE; i++)
{
fprintf(fp,"%lf %lf\n",Real_Point[w][i],Imag_Point[w][i]);
}
fclose(fp);
return 0;
}
```