Research Article
An Efficient Parallel Laplace Transform Spectral Element Method for the Two-Dimensional Schrödinger Equation

Wenting Shao
School of Science, Shanghai Second Polytechnic University, Shanghai 201209, China
Correspondence should be addressed to Wenting Shao; wtshao@sspu.edu.cn
Received 12 March 2022; Accepted 27 June 2022; Published 21 July 2022
Academic Editor: Luigi Rodino

Copyright © 2022 Wenting Shao. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this paper, a parallel Laplace transform spectral element method for the linear Schrödinger equation and the cubic nonlinear Schrödinger equation is introduced. The main advantage of this new scheme is that it has the high accuracy of spectral methods and the parallel efficiency of the Laplace transform method. The key idea is twofold. First, Laplace transform is utilized to eliminate time dependency and the subsequent boundary value problems are discretized by using spectral element method. Second, a parallel Laplace transform method is developed to improve the computational efficiency. For the cubic nonlinear Schrödinger equation, the increment linearization technique is employed to deal with the nonlinear terms. Numerical experiments are addressed to demonstrate the accuracy and effectiveness of the proposed method. Compared with the Crank–Nicolson scheme, the CPU time is greatly saved by the parallel Laplace transform for both linear and nonlinear problems.

1. Introduction

In physics, especially in quantum mechanics, the Schrödinger equation is used to describe how the quantum state of a physical system changes in time. Currently, this equation has been widely applied in many areas, for example, optics, underwater acoustics, material science, plasma physics, electromagnetic wave propagation, design of optoelectronic devices, and Bose–Einstein condensation [1, 2].

Since in general, finding the analytical solutions of the two-dimensional Schrödinger equation can be a hard task, one has to make use of numerical procedures and extensive numerical methods have been studied in recent years [3]. Concerning the time discretization, different directions can be investigated, for example, the finite difference scheme [4–6], the Crank–Nicolson (CN) scheme [7–9], the time-splitting scheme [10, 11], the Runge–Kutta (RK) scheme [1, 2, 12–14], and the integrating factor method [15]. Among these schemes, the CN scheme is implicit and unconditionally stable. The main disadvantage of the CN scheme is that it needs to solve a nonlinear system at each time step for the nonlinear problem. Although the nonlinearity can be dealt with using some fixed-point iteration method, it might be rather taxing and time consuming. A possible alternative way of handling the nonlinearity is the time-splitting scheme. Its basic idea is to split NLSE into a linear part and a nonlinear part. The linear subproblem can be solved by the CN method in time. The nonlinear subproblem in some situations where the potential and nonlinearity can be integrated analytically can be solved exactly. Thus, the scheme is explicit and unconditionally stable. These temporal schemes can be coupled with high-order spatial discretizations like spectral or pseudo-spectral methods. Nonetheless, most of the schemes are low-order accuracy in time which limits the use of large time steps in numerical implementation and the global accuracy. It is therefore necessary to develop high-order temporal discretizations to improve the accuracy and efficiency.

The Laplace transform (LT) represents a very effective tool for solving time-dependent problems in science and engineering [16]. It removes the time dependency and converts the original problem into a series of boundary value problems (BVPs). Chen and Wu et al. combined LT with the rational spectral collocation method to solve the advection-diffusion equation [17] and one-dimensional Burgers’
A coupled method of LT and Legendre wavelets was presented in [19] for one-dimensional Klein–Gordon equations. In [20], some different types of unsteady flows of the third grade fluid were studied by the use of LT in time. However, LT never really becomes popular in computational work, especially in conjunction with spectral element method. There are two reasons: one is the numerical inversion of LT that is an ill-posed problem when the transform is known only as a real-valued function, and the other one is the nonlinear term that makes the direct use of LT challenging. For the first issue, the idea proposed by Talbot [21] is adopted. The technique combines the trapezoidal rule or the midpoint rule with the contour integration by using the complex arithmetic. The convergence rate for the optimized quadrature formula could reach a geometric rate $O(e^{-CM})$ [22], where $M$ is the number of sample points. For the second issue, several techniques have been developed to overcome the limitation, such as the homotopy perturbation method [23] and Adomian decomposition method [24, 25]. The Adomian decomposition method is done by expanding the nonlinear term as an infinite series of the Adomian polynomials, which can be calculated by a recursive relationship. However, it causes a lot of extra cost for solving time-dependent problems. The increment linearization method was first proposed by Wu et al. [26] and then integrated with LT to solve Burgers’ equation (18). Its key idea is to define an increment function at each time step: $\Delta u(x,t) = u(x,t) - u(x,t_n)$, $t_n \leq t \leq t_{n+1}$, linearize the equation by omitting the nonlinear terms of the order $O(\tau^y)$, $y \geq 2$, and finally get the approximation by $u(x,t_{n+1}) = u(x,t_n) + \Delta u(x,t_{n+1})$. In this work, we will employ the increment linearization method to deal with the nonlinear terms.

Concerning the spatial discretization, many techniques have been developed including the finite difference method [11], finite element method [8, 27], Legendre spectral method [13], Fourier/Chebyshev pseudo-spectral methods [1, 7, 10, 12, 15, 28], Chebyshev wavelet method [29], radial basis meshless collocation method [4], radial basis function–finite difference method [2, 14], and Sinc collocation and Sinc Galerkin methods [5]. It is well known that the spectral method is a very useful tool for the solution of PDEs because of its spectral accuracy, when the geometry of the problem is smooth and regular. In [13, 28], Galerkin–Legendre spectral method and Chebyshev pseudo-spectral method were combined with the RK method to solve the two-dimensional linear Schrödinger equation. However, there have been few references about integrating spectral element method (SEM) with LT to solve time-dependent PDEs. This motivates our study to introduce SEM for solving two-dimensional Schrödinger equations. SEM was first proposed by Patera [30] for the numerical solution of the incompressible Navier–Stokes equations, and it combines the high accuracy of spectral methods and the flexibility of finite element method. The approximate result of SEM provides global spectral convergence, not at the expense of “uncontrollable” resolution. Compared with conventional spectral or pseudo-spectral methods, SEM requires weaker smoothness in operations and the fine grid can be obtained by increasing either the interpolation degree ($p$-convergence) in each element or the amount of elements ($h$-convergence). In this paper, the choice of SEM makes high-order accuracy in space and it is compatible with the time-discrete method LT.

In this paper, we consider the two-dimensional linear Schrödinger equation (LSE) [28].

$$\frac{i}{\tau} \frac{\partial \Phi(x,t)}{\partial t} + \nabla \cdot (K \nabla \Phi(x,t)) + w(x)\Phi(x,t) + f(x,t) = 0, \quad (1)$$

and the cubic nonlinear Schrödinger equation (NLSE).

$$\frac{i}{\tau} \frac{\partial \Phi(x,t)}{\partial t} + \nabla \cdot (K \nabla \Phi(x,t)) + \beta |\Phi(x,t)|^2 \Phi(x,t) + w(x)\Phi(x,t) = 0, \quad (2)$$

in the rectangle domain $\Omega$ and $0 < t \leq T$ with the initial condition

$$\Phi(x,0) = \Phi_0(x), \text{in} \Omega, \quad (3)$$

and the homogenous Dirichlet boundary conditions

$$\Phi(x,t) = 0, \text{on} \partial \Omega \times [0,T]. \quad (4)$$

Here, $i^2 = -1$, $x = (x, y)$, $\Phi(x,t)$ is the complex-valued function, $w(x)$ is a given real-valued power function, and $K$ and $\beta$ are given real constants. The cubic NLSE admits the mass conservation law [3]

$$M(t) = \int_{\Omega} |\Phi(x,t)|^2 d\sigma = \int_{\Omega} |\Phi_0(x)|^2 d\sigma = M(0). \quad (5)$$

and the energy conservation law

$$E(t) = \int_{\Omega} \left( K |\nabla \Phi(x,t)|^2 - w(x) |\Phi(x,t)|^2 - \frac{\beta}{2} |\Phi(x,t)|^4 \right) d\sigma$$

$$= \int_{\Omega} \left( K |\nabla \Phi_0(x)|^2 - w(x) |\Phi_0(x)|^2 - \frac{\beta}{2} |\Phi_0(x)|^4 \right) d\sigma = E(0). \quad (6)$$

for $t > 0$.

A coupled method of Laplace transform and spectral element method, named LTSEM, is presented for solving the LSE and the cubic NLSE. The application of LT converts the original time-dependent problem into a series of BVPs with different LT parameters. SEM is applied to solve the subsequent equations. For the cubic NLSE, the increment linearization technique is introduced to deal with the nonlinear terms. It removes the stumbling block of using LT to solve the nonlinear problem. Unlike the conventional time-marching method, the LT method has an advantage of excellent parallel efficiency, but it is not addressed in detail by numerical experiments in literature. Inspired by this, we develop a parallel LTSEM (pLTSEM) to improve the computational efficiency of LTSEM and it can be easily implemented with the MATLAB parallel computing toolbox. This is the main contribution of this work.

The paper is organized as follows. In Section 2, we consider LSE and convert it into a series of BVPs by using
In Section 3, Talbot’s method is briefly reviewed for the numerical inversion of LT. In Section 4, SEM is utilized to solve the BVPs generated by using LT and a detailed description of pLTSEM is given for LSE. In Section 5, the cubic NLSE is considered. The increment linearization method is introduced to deal with the nonlinearity. The description of pLTSEM for this case is given. In Section 6, numerical examples are tested to show the performance of our proposed method. Finally, in Section 7, a brief conclusion is drawn and some comments are included.

2. Laplace Transform Method

In this section, we will introduce the LT method to convert the LSE into a series of BVPs.

In practice, we decompose the complex-valued function \( \Phi(x, t) \) into its real and imaginary parts by writing \( \Phi(x, t) = u(x, t) + iv(x, t) \), and set \( f(x, t) = f_u(x, t) + if_v(x, t) \), where \( u(x, t) \), \( v(x, t) \), \( f_u(x, t) \), and \( f_v(x, t) \) are real-valued functions. Problem (1) is transformed into the following coupled real-valued differential equations

\[
\begin{align*}
&u_t(x, t) + \nabla \cdot (K \nabla v(x, t)) + w(x)v(x, t) + f_v(x, t) = 0, \\
v_t(x, t) - \nabla \cdot (K \nabla u(x, t)) - w(x)u(x, t) - f_u(x, t) = 0,
\end{align*}
\]

with the boundary conditions

\[
\tilde{u}(x, s) = 0, \tilde{v}(x, s) = 0, \text{on } \partial \Omega,
\]

where \( \tilde{f}_u(x, s) \) and \( \tilde{f}_v(x, s) \) are the Laplace transforms of \( f_u(x, t) \) and \( f_v(x, t) \), respectively.

Next, we set the decompositions \( s = s^R + is^I \),

\[
\tilde{u}(x, s) = u^R(x) + u^I(x), \quad \tilde{v}(x, s) = v^R(x) + v^I(x),
\]

\[
\tilde{f}_u(x, s) = \tilde{f}^R_u(x) + i\tilde{f}^I_u(x), \quad \text{and} \quad \tilde{f}_v(x, s) = \tilde{f}^R_v(x) + i\tilde{f}^I_v(x),
\]

and then system (12) can be recast to the following linear system satisfied by \( u^R, v^R, \tilde{u}^I, \tilde{v}^I \)

\[
\begin{align*}
&\begin{align*}
&s^R u^R - s^I v^I + \nabla \cdot (K \nabla v^R) + w^R v^I = -\tilde{f}^R_u + u_0, \\
&\quad \quad s^I u^R + s^R v^I + \nabla \cdot (K \nabla v^I) + w^R v^I = -\tilde{f}^I_u,
\end{align*}
\end{align*}
\]

\[
\begin{align*}
-\nabla \cdot (K \nabla u^R) - w u^R + s^R v^R - s^I v^I = \tilde{f}^R_u + u_0, \\
-\nabla \cdot (K \nabla v^I) - w v^I + s^R v^I + s^I v^I = \tilde{f}^I_u,
\end{align*}
\]

with the boundary conditions

\[
\tilde{u}^R = \tilde{u}^I = \tilde{v}^R = \tilde{v}^I = 0, \text{on } \partial \Omega.
\]

Usually, one solves equations (14) and (15) with different LT parameters \( \{s_m\} \) and \( \{\tilde{u}(x, s_m)\} \) and \( \{\tilde{v}(x, s_m)\} \). Then, by doing a numerical inversion of LT, the numerical solutions in the original space could be obtained.

3. Numerical Inversion of Laplace Transform

In the past decade, several results [21, 22, 31] have been obtained for numerical inversion of LT, among which Talbot’s method [21] has received a lot of attention and constitutes the mainstream thanks to the high accuracy and relatively low computational complexity. In this section, we employ Talbot’s method to realize the numerical inversion of LT.

The inversion Laplace transform of \( \tilde{u}(x, s) \) is defined by

\[
u(x, t) = \frac{1}{2\pi i} \int_B e^{st} \tilde{u}(x, s) \, ds,
\]

which is called the Bromwich integral, with \( B \) as the Bromwich line \( \text{Re}(s) = \sigma > \sigma_0, \sigma_0 \) is the maximum value of the real part of all singularities of \( \tilde{u}(x, s) \).

In Talbot’s method, the Bromwich line is deformed into a curve \( \Gamma \) that begins and ends in the left half-plane, such that \( \text{Re}(s) \to -\infty \) on the contour. Owing to the exponential factor \( e^{st} \), the integrand decays rapidly on it. It has been proved [21] that Talbot’s method is applicable to both a wide
range of Laplace transform functions and values of $t$ where we need to compute $u(x, t)$.

Talbot’s contour is parameterized by

$$\Gamma: s(\theta) = \sigma + \mu (\theta \cot \theta + vi\theta), -\pi \leq \theta \leq \pi,$$  

where $\sigma$, $\mu$, and $v$ are real parameters that determine the geometry of the curve. Both $\mu$ and $v$ are positive. Assume that $\Gamma$ lies in the region of analyticity of $\widetilde{u}(x, s)$ and use the Cauchy integral theorem; the Bromwich integral (16) with Talbot’s contour becomes

$$u(x, t) = \frac{1}{2\pi i} \int_{-\pi}^{\pi} e^{i\theta t} \widetilde{u}(x, s(\theta)) s'(\theta) d\theta,$$  

which can be approximated by the midpoint formula. Define integral points as

$$\theta_m = (2m + 1) \frac{\pi}{2M},$$  

$$m = -M, \ldots, M - 1,$$

and allow $s_m = s(\theta_m)$, $s_m' = s'(\theta_m)$; the approximation to (18) is given by

$$u(x, t) \approx \frac{1}{2M} \sum_{m=-M}^{M-1} e^{i\theta t} s_m \widetilde{u}(x, s_m) = \frac{1}{M} \sum_{m=0}^{M-1} e^{i\theta t} s_m \widetilde{u}(x, s_m).$$  

where the second equation is deduced by symmetry.

**Remark 1.** It was analyzed in [21] that Talbot’s method with fixed parameters converges at a subgeometric rate of $\mathcal{O}(e^{-\gamma |t|})$. Further, Weideman [22] used the saddle point method to give the error estimate, that is,

$$E_M(t) = \mathcal{O}(e^{(\sigma + \mu)\gamma - 2\sqrt{\mu M}}), M \to \infty.$$  

This estimate suggests the strategy of choosing $\sigma$ and $\mu \propto M/t$ could lead to an improved convergence rate $\mathcal{O}(e^{-\gamma |t|})$. Besides, Weideman found optimal parameters for some parabolic problems: when the following Talbot’s contour

$$\Gamma: s(\theta) = \frac{M}{t} \left(-0.4814 + 0.4774\theta \cot(0.7409\theta) + 0.2698i\theta\right).$$  

is used as the Bromwich line, the convergence rate is given by $\mathcal{O}(e^{-1.90M})$.

Generally, we give a uniform partition of the time interval and denote discrete time $t_n = nr$, where $r$ is the time step. At each time step, LT with its numerical inversion is used for the temporal discretization. Accordingly, $t$ will be replaced by $r$ in (21).

Next, we plan to discretize equations (14) and (15) by using SEM in space and get the approximations of $\tilde{u}(x, s_m)$ and $\tilde{v}(x, s_m)$.

### 4. Spatial Discretization and Parallel Computation

In this section, the Legendre SEM is briefly recalled for the spatial discretization and a detailed description of pLTSEM is given for LSE.

#### 4.1. Notations and Discrete Schemes

We denote by $(\cdot, \cdot)$ and $\| \cdot \|$ the inner product and the norm of the space $L^2(\Omega)$. Denote $H^s(\Omega) (s \geq 0)$ to be the Sobolev space with the norm $\| \cdot \|_{H^s(\Omega)}$ and introduce the space: $H^1_0(\Omega) = \{ \phi(x) \in H^1(\Omega); \phi(x)|_{\partial \Omega} = 0 \}$.

The weak form of equations (14) and (15) can be written as follows: for any given $s_m$, find $\{ \tilde{u}^R, \tilde{u}^I, \tilde{v}^I, \tilde{v}^R \} \in (H^1_0(\Omega))^4$, such that

$$\begin{aligned}
& \int_{-\pi}^{\pi} \left( \frac{d}{ds}\tilde{u}^R, \phi \right) ds - \int_{-\pi}^{\pi} \left( \frac{d}{ds}\tilde{u}^I, \phi \right) ds - \int_{-\pi}^{\pi} \left( \nabla \tilde{v}^R, \nabla \phi \right) ds + \left( w\tilde{v}^R, \phi \right) ds = \left( -\tilde{f}^R_v, \phi \right), \\
& \int_{-\pi}^{\pi} \left( \frac{d}{ds}\tilde{u}^I, \phi \right) ds + \int_{-\pi}^{\pi} \left( \frac{d}{ds}\tilde{v}^R, \phi \right) ds - \int_{-\pi}^{\pi} \left( \nabla \tilde{v}^I, \nabla \phi \right) ds + \left( w\tilde{v}^I, \phi \right) ds = \left( -\tilde{f}^I_v, \phi \right), \\
& \int_{-\pi}^{\pi} \left( \frac{d}{ds}\tilde{v}^I, \phi \right) ds - \int_{-\pi}^{\pi} \left( w\tilde{v}^I, \phi \right) ds + \int_{-\pi}^{\pi} \left( \tilde{v}^I, \phi \right) ds = \left( \tilde{f}^I_w, \phi \right), \\
& \int_{-\pi}^{\pi} \left( \frac{d}{ds}\tilde{v}^R, \phi \right) ds + \int_{-\pi}^{\pi} \left( \tilde{v}^R, \phi \right) ds = \left( \tilde{f}^R_w, \phi \right), \forall \phi \in H^1_0(\Omega).
\end{aligned}$$  

We take $\mathcal{F}_h$ be a uniform regular partition of quadrilateral element of domain $\Omega$: $\mathcal{T}_h = \bigcup_{l=1}^{N_h} \mathcal{T}_l = \bigcup_{l=1}^{N_h} \mathcal{T}_l \times \mathbb{R}^2 \subset \Omega$. Let $I = [-1, 1]$ be the reference interval and denote $\Omega_\alpha = [-1, 1]^2$. Define $P_N(I)$ to be the space of polynomials with degree at most $N$ on the interval $I$, and denote $Q_N(\Omega_\alpha) = \{ \phi \in P_N(I) \} \otimes P_N(I)$. The modal bases of $Q_N(\Omega_\alpha)$ are defined by

$$\phi_{j,n}(\xi, \eta) = \phi_j(\xi) \phi_n(\eta), 0 \leq j, n \leq N,$$  

with

$$\phi_0(\xi) = \frac{(1 - \xi)}{2},$$  

$$\phi_N(\xi) = \frac{(1 + \xi)}{2},$$  

$$\phi_j(\xi) = L_{j-1}(\xi) - L_{j+1}(\xi),$$  

$$1 \leq j \leq N - 1,$$
where \( L_j (\xi) \) is the \( j \)th Legendre polynomial and \( \varphi_i (\eta) \) has the same definition with \( \varphi_j (\xi) \).

For any element \( \Omega = [a, b] \times [c, d] \), we can use the following mapping functions to transfer \( \Omega_i \) to \( \Omega_N \) and its inverse

\[
\xi(x) = \frac{2(x - a)}{b - a} - 1, \\
\eta(y) = \frac{2(y - c)}{d - c} - 1, \quad (x, y) \in \Omega_i.
\]

(25)

The modal bases of \( Q_N (\Omega_i) \) can be easily defined by using (23) and (25).

Define the approximation space

\[
W^0_N (\Omega) = \{ \phi_N (x) \in H_0^1 (\Omega): \phi_N (x)|_{\Omega_i} \in Q_N (\Omega_i), \forall \Omega_i \in \Sigma_i \}.
\]

(26)

The discrete weak form of (22) is as follows: for any given \( s_m \), find \( \{ \tilde{u}_R^N, \tilde{u}_N^I, \tilde{v}_R^N, \tilde{v}_N^I \} \in (W^0_N (\Omega))^4 \), such that

\[
\begin{cases}
\mathbf{s}_m^R (u_R^N, \phi_N) - \mathbf{s}_m^I (\tilde{u}_N^I, \phi_N) - K (\nabla \tilde{v}_R^N, \nabla \phi_N) + (w \tilde{v}_N^R, \phi_N) = \left( -\tilde{f}_u^R + u_0, \phi_N \right), \\
\mathbf{s}_m^I (\tilde{u}_N^I, \phi_N) + \mathbf{s}_m^R (\tilde{u}_N^R, \phi_N) - K (\nabla \tilde{v}_I^R, \nabla \phi_N) + (w \tilde{v}_N^I, \phi_N) = \left( -\tilde{f}_u^I + v_0, \phi_N \right), \\
\mathbf{K} \nabla \tilde{u}_N^I, \nabla \phi_N \left( -w \tilde{u}_R^N, \phi_N \right) + \mathbf{s}_m^R (\tilde{v}_R^N, \phi_N) - \mathbf{s}_m^R (\tilde{v}_N^I, \phi_N) = \left( f_u^R + v_0, \phi_N \right), \\
\mathbf{K} \nabla \tilde{u}_N^R, \nabla \phi_N \left( -w \tilde{u}_I^I, \phi_N \right) + \mathbf{s}_m^R (\tilde{v}_R^I, \phi_N) + \mathbf{s}_m^R (\tilde{v}_N^I, \phi_N) = \left( f_u^I + v_0, \phi_N \right),
\end{cases}
\]

(27)

On any element \( \Omega_i \), we assume the approximations

\[
\tilde{u}_N^R (x) = \sum_{j=0}^{N} \sum_{m=0}^{N} \tilde{u}_N^{j,m} \bar{\psi}_{j,m} (x), \\
\tilde{u}_N^I (x) = \sum_{j=0}^{N} \sum_{m=0}^{N} \tilde{u}_N^{j,m} \bar{\psi}_{j,m} (x), \\
\tilde{v}_N^R (x) = \sum_{j=0}^{N} \sum_{m=0}^{N} \tilde{v}_N^{j,m} \bar{\psi}_{j,m} (x), \\
\tilde{v}_N^I (x) = \sum_{j=0}^{N} \sum_{m=0}^{N} \tilde{v}_N^{j,m} \bar{\psi}_{j,m} (x).
\]

(28)

\[
\begin{bmatrix}
\mathbf{s}_m^R \mathbf{M}_i & -\mathbf{s}_m^I \mathbf{M}_i & -\mathbf{K} \mathbf{A}_i - \mathbf{C}_i & \mathbf{O}_i \\
\mathbf{s}_m^I \mathbf{M}_i & \mathbf{s}_m^R \mathbf{M}_i & -\mathbf{K} \mathbf{A}_i + \mathbf{C}_i & \mathbf{O}_i \\
\mathbf{K} \mathbf{A}_i - \mathbf{C}_i & \mathbf{O}_i & \mathbf{s}_m^R \mathbf{M}_i & -\mathbf{s}_m^I \mathbf{M}_i \\
\mathbf{K} \mathbf{A}_i + \mathbf{C}_i & \mathbf{O}_i & \mathbf{s}_m^R \mathbf{M}_i & \mathbf{s}_m^I \mathbf{M}_i
\end{bmatrix}
\begin{bmatrix}
\tilde{U}_i \\
\tilde{b}_i
\end{bmatrix} = \begin{bmatrix}
\mathbf{U}_i \\
\tilde{b}_i
\end{bmatrix},
\]

(29)

where the element vector \( \tilde{U}_i \) is composed of expansion coefficients

\[
\mathbf{U}_i = \left[ \text{Vec} \left( \left[ \tilde{u}_N^{j,m} \right]_{j,m=0}^{N} \right) \right]^T, \left[ \text{Vec} \left( \left[ \tilde{u}_N^{j,m} \right]_{j,m=0}^{N} \right) \right]^T, \left[ \text{Vec} \left( \left[ \tilde{v}_N^{j,m} \right]_{j,m=0}^{N} \right) \right]^T, \left[ \text{Vec} \left( \left[ \tilde{v}_N^{j,m} \right]_{j,m=0}^{N} \right) \right]^T.
\]

(30)

and “Vec” is the matrix vectorization operator. The element matrices \( \mathbf{M}_i, \mathbf{A}_i, \) and \( \mathbf{C}_i \) are of dimension \((N + 1)^2 \times (N + 1)^2\) and could be calculated as follows:
\[
\begin{align*}
\mathbf{M}_i &= \left[ (\bar{\phi}_j, \bar{\phi}_{j'}) \right]_{j,n,j',n'}, \\
\mathbf{A}_i &= \left[ (\nabla \bar{\phi}_j, \nabla \bar{\phi}_{j'}) \right]_{j,n,j',n'}, \\
\mathbf{C}_i &= \left[ (u \bar{\phi}_j, \bar{\phi}_{j'}) \right]_{j,n,j',n'},
\end{align*}
\] (31)

and \( \mathbf{O}_i = [0]_{(N+1)^2 \times (N+1)^2} \). The element vectors
\[
\bar{\mathbf{b}}_i = \left[ \left( \bar{b}^1_i \right)^T, \left( \bar{b}^2_i \right)^T, \left( \bar{b}^3_i \right)^T, \left( \bar{b}^4_i \right)^T \right]^T.
\] (32)

with the components
\[
\begin{align*}
\bar{\tau}^1_i &= \text{Vec} \left( \left( -f_j + u_0 \bar{\phi}_j \right) \Omega_{j,n} \right), \\
\bar{\tau}^2_i &= \text{Vec} \left( \left( -f_j + u_0 \bar{\phi}_j \right) \Omega_{j,n} \right), \\
\bar{\tau}^3_i &= \text{Vec} \left( \left( f_j - v_0 \bar{\phi}_j \right) \Omega_{j,n} \right), \\
\bar{\tau}^4_i &= \text{Vec} \left( \left( f_j - v_0 \bar{\phi}_j \right) \Omega_{j,n} \right).
\end{align*}
\] (33)

Here, the Gauss integral formula is adopted in the quadratures (31) and (33).

Assembling the contributions from all elements leads to the global matrices \( \mathbf{M}, \mathbf{A}, \mathbf{C} \) and vector \( \bar{\mathbf{b}} \), and finally, arrive at the following matrix form
\[
\begin{pmatrix}
\mathbf{s}_m^R \mathbf{M} & -\mathbf{s}_m^i \mathbf{M} - \mathbf{K} \mathbf{A} + \mathbf{C} & \mathbf{O} \\
\mathbf{s}_m^i \mathbf{M} & \mathbf{s}_m^R \mathbf{M} & -\mathbf{K} \mathbf{A} + \mathbf{C} \\
\mathbf{K} \mathbf{A} - \mathbf{C} & \mathbf{O} & \mathbf{s}_m^R \mathbf{M} - \mathbf{s}_m^i \mathbf{M} \\
\mathbf{O} & \mathbf{K} \mathbf{A} - \mathbf{C} & \mathbf{s}_m^R \mathbf{M} - \mathbf{s}_m^i \mathbf{M}
\end{pmatrix}
= \bar{\mathbf{U}},
\] (34)

where the vector \( \bar{\mathbf{U}} \) contains the expansion coefficients of all elements. The right-hand side
\[
\bar{\mathbf{b}} = \left[ \left( \bar{b}^1 \right)^T, \left( \bar{b}^2 \right)^T, \left( \bar{b}^3 \right)^T, \left( \bar{b}^4 \right)^T \right]^T.
\] (35)

where \( \bar{b}^{1,2,3,4} \) could be calculated by (33) defined on the whole domain.

In the final of this subsection, we present the following theorem that will be used in the analysis of the existence and uniqueness of the solution of (34).

**Theorem 1** (see [32]). The matrix \( \mathbf{L} \in \mathbb{R}^{m \times n} \) is positive definite if and only if its symmetric part \( \mathbf{W} = (\mathbf{L} + \mathbf{L}^T)/2 \) is positive definite. Besides, \( \mathbf{W} \) is positive definite if and only if its eigenvalues are positive.

Working back from the matrix formulation to the functional form of the integral operator, it is easy to show that matrices \( \mathbf{A} \) and \( \mathbf{C} \) are symmetric, and the mass matrix \( \mathbf{M} \) is symmetric and positive definite [33]. Denote the coefficient matrix of (34) by \( \mathbf{L} \); then, we have
\[
\begin{pmatrix}
\mathbf{s}_m^R \mathbf{M} & \mathbf{O} & \mathbf{O} & \mathbf{O} \\
\mathbf{O} & \mathbf{s}_m^R \mathbf{M} & \mathbf{O} & \mathbf{O} \\
\mathbf{O} & \mathbf{O} & \mathbf{s}_m^R \mathbf{M} & \mathbf{O} \\
\mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{s}_m^R \mathbf{M}
\end{pmatrix}
= \frac{\mathbf{L} + \mathbf{L}^T}{2},
\] (36)

and it is a block diagonal matrix.

It is a practical choice that avoids sampling the integrand at \( \mathbf{s}_m^R = 0 \). If \( \mathbf{s}_m^R > 0 \), since the mass matrix \( \mathbf{M} \) is positive definite, we obtain that its eigenvalues are all positive, then the block diagonal matrix \( (\mathbf{L} + \mathbf{L}^T)/2 \) is also positive definite. From Theorem 1, it is direct to get that the coefficient matrix \( \mathbf{L} \) is positive definite; thus, the existence and uniqueness of the solution of (34) are guaranteed. If \( \mathbf{s}_m^R < 0 \), we consider the coefficient matrix \( \tilde{\mathbf{L}} = -\mathbf{L} \) and obtain the same conclusion.

Moreover, since the modal bases (23) are constructed by using the orthogonal Legendre polynomials, the matrices \( \mathbf{A} \) and \( \mathbf{M} \) are typically sparse although they may have a full bandwidth [33]. Thus, (34) is large but very sparse. In this paper, we adopt the built-in algorithm in MATLAB to solve it, which is already optimized for solving the system of linear equations.

**4.2. ParallelLTSEM for the Linear Problem.** In Algorithm 1, a description of the pLTSEM for solving LSE is given. The underlying idea of the pLTSEM is straightforward, and the calculation of (34) with different LT parameters \( \mathbf{s}_m \) is independent. This step can be handled more efficiently by the parallel procedure that means the computing procedure can be implemented simultaneously in different CPU cores. Algorithm 1 is divided into three stages of initialization, two-level iteration followed by a solution update step. In the initialization stage, since the matrices \( \mathbf{A}, \mathbf{M}, \mathbf{C} \) depend only on the spatial grid size and the order of polynomial expansion, they can be precomputed before entering into the time loop. The iterative procedure is composed of two loops: the outer loop and the inner loop. The inner loop is a parallel computing process. Its main purpose is to solve the discretized problems on the current time level. The parallel computation is carried out by using the MATLAB parallel computing toolbox, where the parallel loop is a tool for parallel computing single loop. The degree of parallelism is related to the number of CPU cores. The computational time can be saved significantly under the condition of the balance between the computation cost and the data communication.

The outer loop is used to forward the solution process to the next time step until the final time is reached.

**Step 0.** for time level \( n = 0 \) (i.e., \( t = 0 \)), generate the initial mesh, establish matrices \( \mathbf{M}, \mathbf{A}, \mathbf{C} \).
Step 1. for time level $n \geq 1$,
\begin{equation}
\% \text{ parallel computing:}
\end{equation}
\begin{equation}
\text{parfor} \ (m = 0; m < M; m + +) \ \text{(for each given } s_m, \text{)}
\end{equation}
\begin{equation}
\text{establish vectors } \bar{b}^{1,2,3,4},
\end{equation}
\begin{equation}
solve \ the \ global \ system \ equation \ (30) \ to \ get \ \text{coefficient } U,
\end{equation}
\begin{equation}
determine \ \alpha_N(x, s_m), \ \beta_N(x, s_m).
\end{equation}
\begin{equation}
make \ numerical \ inversion \ of \ LT \ to \ get \ \text{coefficients } p_x(x), v_N(x, t_{n+1}).
\end{equation}
\begin{equation}
\text{Step 2. set } n = n + 1,
\end{equation}
\begin{equation}
\text{update \ the \ right-hand \ side \ (33),}
\end{equation}
\begin{equation}
goto \ \text{Step 1}
\end{equation}
\end{equation}
\begin{equation}
end \ for
\end{equation}

5. The Increment Linearization Method for the Nonlinear Problem

The LT method proves its virtues by transforming any differential or integral equation into the algebraic equation; however, it has the limitation to deal with the nonlinearity. In this context, we follow the idea of the increment linearization method [26] and incorporate it into pLTSEM to solve the cubic NLSE (2). The numerical results in [18, 26] demonstrate its advantages of easy formulation and relatively little computational cost.

First, by the decomposition of real and imaginary parts, (2) can be rewritten as

\begin{equation}
\begin{aligned}
\text{with the initial condition } 8 \text{ and the homogenous boundary condition (9).}
\end{aligned}
\end{equation}

At each time step, we decompose $u(x,t)$ and $v(x,t)$ into two parts

\begin{equation}
u(x,t) = u(x,t_n) + p(x,t), \ v(x,t) = v(x,t_n) + q(x,t), \ t \in [t_n, t_{n+1}],
\end{equation}

where $u(x,t_n)$ and $v(x,t_n)$ are the solutions at $t_n = nt$. $p(x,t)$ and $q(x,t)$ are defined as the increment functions and obviously $p(x,t_n) = 0$, $q(x,t_n) = 0$.

Substituting (38) into (37) and omitting the nonlinear term, which is $O(t^\gamma)$, $\gamma \geq 2$, we get the following linear system which is satisfied by $p(x,t)$ and $q(x,t)$

\begin{equation}
\begin{aligned}
p_t(x,t) + \nabla \cdot (K \nabla q(x,t)) + g_{11}^p(x)p(x,t) + g_{12}^p(x)q(x,t) + g_{13}^p(x) = 0,
p_t(x,t) - \nabla \cdot (K \nabla p(x,t)) + g_{21}^q(x)p(x,t) + g_{22}^q(x)q(x,t) + g_{23}^q(x) = 0.
\end{aligned}
\end{equation}

in $\Omega$ and $t_n < t \leq t_{n+1}$ with the initial condition

\begin{equation}
p(x,t_n) = q(x,t_n) = 0, \ \text{in } \Omega.
\end{equation}

and the boundary condition

\begin{equation}
\begin{aligned}
g_{11}^p(x) &= 2\beta u(x,t_n)v(x,t_n), 
g_{12}^p(x) &= w(x) + 3\beta v^2(x,t_n) + \beta u^2(x,t_n), 
g_{13}^p(x) &= K\Delta v(x,t_n) + \{u(x) + \beta u^2(x,t_n) + \beta v^2(x,t_n)\}v(x,t_n), 
g_{21}^q(x) &= -w(x) - 3\beta u^2(x,t_n) - \beta v^2(x,t_n), 
g_{22}^q(x) &= -2\beta u(x,t_n)v(x,t_n), 
g_{23}^q(x) &= -K\Delta u(x,t_n) - \{u(x) + \beta u^2(x,t_n) + \beta v^2(x,t_n)\}u(x,t_n).
\end{aligned}
\end{equation}
In order not to render this paper unduly long, we avoid presenting in detail the derivation of above equations. The interested readers may consult reference [26]. In particular, we mention that the approximation problem (39) is obtained by neglecting the nonlinear terms of the order $O(r^2), \gamma \geq 2$. Therefore, it is expected that the approximations can reach the second-order accuracy in the temporal direction as reported in [18, 26], which will be verified in the numerical experiments.

Second, LT is used for the temporal discretization at each time step. Denote the Laplace transform of $p(x,t), q(x,t)$ by $\hat{p}(x,s), \hat{q}(x,s)$, subject to the initial conditions (40), on the system (39) yields: for any given $s_m$

\[
\begin{align*}
&\left\{ s_m \hat{p}(x,s_m) + \nabla \cdot (K \nabla \hat{q}(x,s_m)) + \hat{g}_{11}^s(x) \hat{p}(x,s_m) + \hat{g}_{12}^s(x) \hat{q}(x,s_m) + \frac{\hat{g}_{13}^s(x)}{s_m} = 0, \\
&\left\{ s_m \hat{q}(x,s_m) - \nabla \cdot (K \nabla \hat{p}(x,s_m)) + \hat{g}_{21}^s(x) \hat{p}(x,s_m) + \hat{g}_{22}^s(x) \hat{q}(x,s_m) + \frac{\hat{g}_{23}^s(x)}{s_m} = 0.
\end{align*}
\]

with homogenous Dirichlet boundary conditions.

Set the decomposition $s_m^R = s_m^R + i s_m^I$, $\hat{p}(x,s_m) = \hat{p}^R(x) + i \hat{p}^I(x)$, and $\hat{q}(x,s_m) = \hat{q}^R(x) + i \hat{q}^I(x)$, and (43) becomes

\[
\begin{align*}
&\left\{ (s_m^R + \hat{g}_{11}^s)^{\hat{p}^R} - s_m^R \hat{p}^I + \nabla \cdot (K \nabla \hat{q}^R) + \hat{g}_{12}^s \hat{q}^R = \frac{-s_m^R \hat{g}_{13}^s}{|s_m^R|^2}, \\
&s_m^R \hat{p}^R + (s_m^R + \hat{g}_{11}^s)^{\hat{p}^I} + \nabla \cdot (K \nabla \hat{q}^I) + \hat{g}_{12}^s \hat{q}^I = \frac{s_m^R \hat{g}_{13}^s}{|s_m^R|^2}, \\
&-\nabla \cdot (K \nabla \hat{p}^R) + \hat{g}_{21}^s \hat{p}^R + (s_m^R + \hat{g}_{22}^s)^{\hat{q}^R} - s_m^R \hat{q}^I = \frac{-s_m^R \hat{g}_{23}^s}{|s_m^R|^2}, \\
&-\nabla \cdot (K \nabla \hat{p}^I) + \hat{g}_{21}^s \hat{p}^I + s_m^R \hat{q}^R + (s_m^R + \hat{g}_{22}^s)^{\hat{q}^I} = \frac{s_m^R \hat{g}_{23}^s}{|s_m^R|^2}.
\end{align*}
\]

with homogenous Dirichlet boundary conditions.

The discrete weak form is as follows: for any given $s_m$, find $\{\hat{P}_N^R, \hat{P}_N^I, \hat{q}_N^R, \hat{q}_N^I\} \in (W_N^R(\Omega))^4$, such that

\[
\begin{align*}
&(s_m^R + \hat{g}_{11}^s)^{\hat{P}_N^R} - s_m^R \hat{P}_N^I + \nabla \cdot (\nabla \hat{q}_N^R) + \hat{g}_{12}^s \hat{q}_N^R = \frac{-s_m^R \hat{g}_{13}^s}{|s_m^R|^2}, \\
&s_m^R \hat{P}_N^R + (s_m^R + \hat{g}_{11}^s)^{\hat{P}_N^I} + \nabla \cdot (\nabla \hat{q}_N^I) + \hat{g}_{12}^s \hat{q}_N^I = \frac{s_m^R \hat{g}_{13}^s}{|s_m^R|^2}, \\
&K(\nabla \hat{P}_N^R, \nabla \hat{q}_N^R) + \hat{g}_{21}^s \hat{P}_N^R + (s_m^R + \hat{g}_{22}^s)^{\hat{q}_N^R} - s_m^R \hat{q}_N^I = \frac{-s_m^R \hat{g}_{23}^s}{|s_m^R|^2}, \\
&K(\nabla \hat{P}_N^I, \nabla \hat{q}_N^R) + \hat{g}_{21}^s \hat{P}_N^I + s_m^R \hat{q}_N^R + (s_m^R + \hat{g}_{22}^s)^{\hat{q}_N^I} = \frac{s_m^R \hat{g}_{23}^s}{|s_m^R|^2}, \\
\forall \hat{q}_N \in W_N^0(\Omega).
\end{align*}
\]
Assume the approximations on any element $\Omega$ as follows:

$$
\begin{align*}
\vec{p}_j^R (x) &= \sum_{j=0}^{N} \sum_{m=0}^{N} \vec{p}_{j,m} \vec{q}_{j,m} (x), \\
\vec{q}_j^R (x) &= \sum_{j=0}^{N} \sum_{m=0}^{N} \vec{q}_{j,m} \vec{q}_{j,m} (x), \\
\vec{q}_j^I (x) &= \sum_{j=0}^{N} \sum_{m=0}^{N} \vec{q}_{j,m} \vec{q}_{j,m} (x), \\
\vec{q}_j^I (x) &= \sum_{j=0}^{N} \sum_{m=0}^{N} \vec{q}_{j,m} \vec{q}_{j,m} (x).
\end{align*}
$$

(46)

with the expansion coefficients $[\vec{p}_{j,m}^R]_{j,m=0}^N$, $[\vec{q}_{j,m}^R]_{j,m=0}^N$ and $[\vec{q}_{j,m}^I]_{j,m=0}^N$.

The local linear system arising from (45) has the following partitioned matrix form

$$
\begin{pmatrix}
\vec{r}_{j,m}^R - \vec{s}_{m,j}^R M_i - K_{A,j} + E_{i,j}^p & -O_i \\
\vec{s}_{m,j}^R M_i & \vec{s}_{m,j}^R M_i + D_{i,j}^p & -\vec{s}_{m,j}^l M_i \\
K_{A,j} + E_{i,j}^p & O_i & \vec{s}_{m,j}^R M_i + D_{i,j}^p - \vec{s}_{m,j}^l M_i
\end{pmatrix} U_j = \vec{b}_i,
$$

(47)

where the element vector $\vec{U}_j$ is composed of the expansion coefficients

$$
U_j = \left[ (\text{Vec}(\vec{p}_{j,m}^R)_{j,m=0}^N), (\text{Vec}(\vec{q}_{j,m}^R)_{j,m=0}^N), (\text{Vec}(\vec{q}_{j,m}^I)_{j,m=0}^N) \right]^T.
$$

(48)

The local element matrices $D_i^p$, $E_i^p$, $D_i^q$, and $E_i^q$ are computed by

$$
\begin{align*}
D_i^p &= \left[ (g^{R}_{j,j'} \vec{q}_{j,j'} x_{i}, \vec{q}_{j,j'} x_{i})_{\Omega} \right]_{j,n,j',n'}, \\
E_i^p &= \left[ (g^{I}_{j,j'} \vec{q}_{j,j'} x_{i}, \vec{q}_{j,j'} x_{i})_{\Omega} \right]_{j,n,j',n'}, \\
D_i^q &= \left[ (g^{I}_{j,j'} \vec{q}_{j,j'} x_{i}, \vec{q}_{j,j'} x_{i})_{\Omega} \right]_{j,n,j',n'}, \\
E_i^q &= \left[ (g^{I}_{j,j'} \vec{q}_{j,j'} x_{i}, \vec{q}_{j,j'} x_{i})_{\Omega} \right]_{j,n,j',n'}.
\end{align*}
$$

(49)

and the right-hand side

$$
\vec{b}_i = \begin{bmatrix}
\vec{s}_{m,j}^R M_i & \vec{s}_{m,j}^R M_i + D_{i,j}^p & -\vec{s}_{m,j}^l M_i \\
\vec{s}_{m,j}^R M_i & \vec{s}_{m,j}^R M_i + D_{i,j}^p & -\vec{s}_{m,j}^l M_i \\
\vec{s}_{m,j}^R M_i & \vec{s}_{m,j}^R M_i + D_{i,j}^p & -\vec{s}_{m,j}^l M_i
\end{bmatrix}
$$

(50)

with the components

$$
\begin{align*}
\vec{b}_i^1 &= \text{Vec}\left[ (g^{R}_{j,j'} \vec{q}_{j,j'} x_{i})_{\Omega} \right]_{j,n,j',n'}. \\
\vec{b}_i^2 &= \text{Vec}\left[ (g^{I}_{j,j'} \vec{q}_{j,j'} x_{i})_{\Omega} \right]_{j,n,j',n'}.
\end{align*}
$$

(51)

Assemble the contributions from all elements, and we obtain the following global linear system with the partitioned matrix form

$$
\begin{pmatrix}
\vec{r}_{j,m}^R - \vec{s}_{m,j}^R M_i - K_{A,j} + E_{i,j}^p & -O_i \\
\vec{s}_{m,j}^R M_i & \vec{s}_{m,j}^R M_i + D_{i,j}^p & -\vec{s}_{m,j}^l M_i \\
K_{A,j} + E_{i,j}^p & O_i & \vec{s}_{m,j}^R M_i + D_{i,j}^p - \vec{s}_{m,j}^l M_i
\end{pmatrix} \vec{U}_j = \vec{b}_i,
$$

(52)

where the vector $\vec{U}_j$ is composed of the expansion coefficients of all elements. The coefficient matrix is sparse as can be appreciated from the fact that $M$ is sparse and $O$ is the zero matrix. The right-hand side

$$
\vec{b} = \begin{bmatrix}
\frac{s_{m,j}^R}{|s_{m}|^2} \vec{b}^1 - \frac{s_{m,j}^I}{|s_{m}|^2} \vec{b}^1 - \frac{s_{m,j}^R}{|s_{m}|^2} \vec{b}^2 + \frac{s_{m,j}^I}{|s_{m}|^2} \vec{b}^2
\end{bmatrix}^T,
$$

(53)

where $\vec{b}^1$ and $\vec{b}^2$ could be calculated by (51) defined on the whole domain.

Algorithm 2 describes the process of the pLTSEM for solving the cubic NLSE. The procedure is similar to Algorithm 1, which also has two loops. The outer loop is to forward the time step to the final level, and the inner loop is conducted by the parallel computing for each time step. The difference between two algorithms is that the main purpose of the inner loop in Algorithm 2 is to solve the linear system which is satisfied by the increment functions $p(x,t)$, $q(x,t)$ rather than $u(x,t)$, $v(x,t)$. After that, the computation on the current time level is complete by the addition operation (38) to obtain the approximations $u_N(x,t_{n+1})$ and $v_N(x,t_{n+1})$. Notice that matrices $D_i^p$, $E_i^p$, $D_i^q$, and $E_i^q$ and vectors $\vec{b}', \vec{b}''$ are independent of $s_m$, and they only need to be computed once in the outer loop.

Step 0. for time level $n = 0$ (i.e., $t = 0$),
generate the initial mesh, establish matrices

\[ M, A. \]

**Step 1.** for time level \( n \geq 1, \)

\[
\begin{align*}
g_{11}^n(x), \\
g_{12}^n(x), \\
g_{13}^n(x), \\
g_{21}^n(x), \\
g_{22}^n(x) \text{ and } g_{23}^n(x),
\end{align*}
\]

establish matrices \( D^b, E^b, D^q, E^q \) and vectors \( b^1, b^2. \)

\% parallel computing:

\[
\text{parfor } (m = 0; m < M; m + + ) 
\text{[for each given } s_m, \text{ solve the global system equation (44) to get coefficient } \mathcal{U}, \]

\[
\text{determine } p_N(x, s_m), q_N(x, s_m). \}
\]

make numerical inversion of LT to get \( p_N(x, t_{m+1}), q_N(x, t_{m+1}), \)

\[
\text{compute } u_N(x, t_{m+1}) \text{ and } v_N(x, t_{m+1}) \text{ by using (38).}
\]

**Step 2.** set \( n = n + 1, \) goto Step 1

end for

---

### 6. Numerical Examples

In this section, numerical examples are presented to witness the performance of our proposed method in terms of the accuracy and the parallel efficiency. In the first linear numerical example, the convergence rate of LTSEM is investigated. The nonlinear numerical example demonstrates the validity and feasibility of applying the increment linearization method to the cubic NLSE. Numerical results show the pLTSEM provides much more efficiency than LTSEM, and the pLTSEM with 4 cores is the best. It confirms the superiority of parallel computation of the LT method. All the calculations are conducted on the computer of Intel(R) Core(TM)i5-8500T CPU 2.10 GHz, 16.0 GB, Windows 10(x64).

**Example 1.** In this test case, we study the LSE (1) defined on \([0,8]^2,\) where the parameters adopted are \( K = 1/2, \) \( w(x) = -1, \) and \( f(x, t) \) which are chosen such that the exact solution is

\[
\Phi(x, t) = e^{t/2}(8 - x)(8 - y) \sin(xy) + i(8 - x)yt \sin(x(8 - y)).
\]

(54)

The initial condition could be obtained easily from setting the exact solution at \( t = 0. \)

First, we investigate the order of convergence of LTSEM in the spatial direction and the temporal direction. For the spatial direction, the approximation can converge either algebraically by increasing the number of elements with fixed degree of the polynomials or exponentially by increasing the degree of the polynomials with fixed number of elements [30]. Here, we choose the latter path and employ a uniform quadrilateral element mesh with the size of 2. The discrete \( L^2 \) norm error \( \| e(t) \|_L^2 \) and the discrete \( H^1 \) norm error \( \| e(t) \|_H^1 \) of the modulus \( |\Phi(t)| \) at \( t = 1 \) are computed.

In Figure 1(a), we plot the errors with various \( N \) (the degree of the polynomial) and set \( M = 80 \) (the number of integral points) at each time step to ignore the error in the temporal direction. For this linear problem, we choose the time step \( \tau = 1 \). One can see that the convergence rate in the spatial direction is \( O(e^{-1.67N}) \) for both two norms. Table 1 lists the errors with various \( M \), where we set \( N = 26 \) to ignore the error in the spatial direction. The convergence rate is shown in Figure 1(b), and it is found that the order is \( O(e^{-0.75M}) \) for both two norms.

Second, we show the excellent performance of the pLTSEM toward LTSEM. Table 1 presents the comparison of CPU time and errors of LTSEM and pLTSEM, where \( T_{LTm} \) denotes the CPU time required to get the approximations for different values of \( s_m \), \( T_{LT} \) denotes the CPU time required for the numerical inversion of LT, and \( T_{all} \) denotes the total CPU time required for all the operations.

Both tables clearly observed that for LTSEM, the numerical inversion of LT part accounts a small amount of calculation and the solving system (30) part is much more time consuming. Besides, the errors for LTSEM and pLTSEM are always the same, independently of the number of processors that are used. Hence, the parallel algorithm for the solving of equation (30) with different values of \( s_m \) does improve the computational efficiency.

For the sake of comparison, CNSEM is also considered, and it is using the CN scheme for the time derivative and SEM for the space derivative. The CN scheme is a kind of time-marching scheme with second-order convergence rate. The numerical results of CNSEM with various time steps \( \tau \) are listed in Table 2.

For convenience, we also plot errors versus the total CPU time in Figure 2, where \( x \)-axis represents the total CPU time \( T_{all} \) and \( y \)-axis represents the errors. It is easy to find that pLTSEM works much better than LTSEM and CNSEM; that is, when the comparable magnitude of the error is obtained, pLTSEM requires much less CPU time and CNSEM is the most expensive. More precisely, from the last three columns in Table 1, we could conclude that pLTSEM with 2 cores reduces nearly 38 % of the total CPU time of LTSEM, and pLTSEM with 4 cores reduces nearly 54 %.

In addition, Table 3 provides the errors and the total CPU time at different times with \( M = 80 \) and \( N = 26 \). The time interval \([0, 5]\) is discretized by the time step \( \tau = 1 \) and LT is used at each time step.

**Example 2.** [9] Consider the cubic NLSE (2) defined on \( \Omega = [0, \pi]^2, \) where the parameters adopted are \( K = 1, \beta = 1, \) \( w(x) = - \sin^2 x \sin^2 y, \) and the exact solution is given by

\[
\Phi(x, t) = e^{-2it} \sin x \sin y.
\]

(55)

The discrete \( L^2 \) and \( H^1 \) norm errors at \( t = 1 \) with various time steps \( \tau \) are reported in Table 4, where the quadrilateral element mesh with the size of \( \pi / 8 \) is employed and \( M = 16, N = 6. \) These results show that the time discretization scheme (43) is second-order accurate for this two-dimensional nonlinear problem. This is consistent with the numerical results reported in [18, 26] and also verifies that the
Figure 1: Numerical results of LTSEM at $t = 1$: (a) errors with various $N$ and $M = 80$; (b) errors with various $M$ and $N = 26$, where $N$ is the degree of polynomial and $M$ is the number of integral points at each time step, Example 1.

Table 1: Results of LTSEM and pLTSEM with various $M$ and $N = 26$ at $t = 1$, Example 1.

<table>
<thead>
<tr>
<th>$M$</th>
<th>$|e(1)|_2$</th>
<th>$|e(1)|_{H_1}$</th>
<th>$T_{LT}$ (s)</th>
<th>$T_{all}$ (s)</th>
<th>$T_{all}$ (s)</th>
<th>$T_{all}$ (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>4.10e−01</td>
<td>1.66e+00</td>
<td>34.14</td>
<td>1.59</td>
<td>37.73</td>
<td>23.71</td>
</tr>
<tr>
<td>52</td>
<td>1.00e−01</td>
<td>4.10e−01</td>
<td>36.13</td>
<td>1.58</td>
<td>39.71</td>
<td>25.02</td>
</tr>
<tr>
<td>54</td>
<td>2.43e−02</td>
<td>9.89e−02</td>
<td>38.17</td>
<td>1.59</td>
<td>41.76</td>
<td>26.30</td>
</tr>
<tr>
<td>56</td>
<td>5.74e−03</td>
<td>2.33e−02</td>
<td>39.99</td>
<td>1.58</td>
<td>43.57</td>
<td>27.41</td>
</tr>
<tr>
<td>58</td>
<td>1.33e−03</td>
<td>5.41e−03</td>
<td>41.93</td>
<td>1.59</td>
<td>45.43</td>
<td>28.55</td>
</tr>
<tr>
<td>60</td>
<td>3.03e−04</td>
<td>1.23e−03</td>
<td>44.01</td>
<td>1.59</td>
<td>47.50</td>
<td>29.57</td>
</tr>
<tr>
<td>62</td>
<td>6.77e−05</td>
<td>2.75e−04</td>
<td>45.95</td>
<td>1.60</td>
<td>49.54</td>
<td>31.16</td>
</tr>
<tr>
<td>64</td>
<td>1.49e−05</td>
<td>6.06e−05</td>
<td>47.99</td>
<td>1.59</td>
<td>51.49</td>
<td>32.41</td>
</tr>
<tr>
<td>66</td>
<td>3.23e−06</td>
<td>1.31e−05</td>
<td>50.04</td>
<td>1.59</td>
<td>53.54</td>
<td>33.70</td>
</tr>
<tr>
<td>68</td>
<td>6.89e−07</td>
<td>2.80e−06</td>
<td>51.99</td>
<td>1.58</td>
<td>55.57</td>
<td>34.92</td>
</tr>
<tr>
<td>70</td>
<td>1.45e−07</td>
<td>5.90e−07</td>
<td>53.90</td>
<td>1.59</td>
<td>57.47</td>
<td>36.18</td>
</tr>
<tr>
<td>72</td>
<td>3.00e−08</td>
<td>1.22e−07</td>
<td>55.85</td>
<td>1.59</td>
<td>59.43</td>
<td>37.41</td>
</tr>
<tr>
<td>74</td>
<td>6.18e−09</td>
<td>2.52e−08</td>
<td>57.99</td>
<td>1.60</td>
<td>61.57</td>
<td>38.77</td>
</tr>
<tr>
<td>76</td>
<td>1.23e−09</td>
<td>5.25e−09</td>
<td>59.94</td>
<td>1.59</td>
<td>63.52</td>
<td>39.93</td>
</tr>
<tr>
<td>78</td>
<td>2.38e−10</td>
<td>4.19e−09</td>
<td>62.07</td>
<td>1.58</td>
<td>65.65</td>
<td>41.29</td>
</tr>
<tr>
<td>80</td>
<td>4.18e−11</td>
<td>3.63e−09</td>
<td>64.13</td>
<td>1.59</td>
<td>67.70</td>
<td>42.64</td>
</tr>
</tbody>
</table>

Table 2: Results of CNSEM with various $\tau$ and $N = 26$ at $t = 1$, Example 1.

<table>
<thead>
<tr>
<th>$\tau$</th>
<th>$|e(1)|_2$</th>
<th>order</th>
<th>$|e(1)|_{H_1}$</th>
<th>order</th>
<th>$T_{all}$ (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/2(^5)</td>
<td>3.40e+00</td>
<td>—</td>
<td>9.48e+00</td>
<td>—</td>
<td>9.94</td>
</tr>
<tr>
<td>1/2(^6)</td>
<td>8.51e−01</td>
<td>1.99</td>
<td>2.37e+00</td>
<td>1.99</td>
<td>19.89</td>
</tr>
<tr>
<td>1/2(^7)</td>
<td>2.12e−01</td>
<td>1.99</td>
<td>5.94e−01</td>
<td>1.99</td>
<td>39.70</td>
</tr>
<tr>
<td>1/2(^8)</td>
<td>5.32e−02</td>
<td>1.99</td>
<td>1.48e−01</td>
<td>1.99</td>
<td>79.47</td>
</tr>
<tr>
<td>1/2(^9)</td>
<td>1.33e−02</td>
<td>1.99</td>
<td>3.71e−02</td>
<td>2.00</td>
<td>156.99</td>
</tr>
<tr>
<td>1/2(^{10})</td>
<td>3.32e−03</td>
<td>2.00</td>
<td>9.29e−03</td>
<td>2.00</td>
<td>310.47</td>
</tr>
<tr>
<td>1/2(^{11})</td>
<td>8.31e−04</td>
<td>2.00</td>
<td>2.32e−03</td>
<td>2.00</td>
<td>625.83</td>
</tr>
<tr>
<td>1/2(^{12})</td>
<td>2.07e−04</td>
<td>2.00</td>
<td>5.80e−04</td>
<td>2.00</td>
<td>1269.93</td>
</tr>
<tr>
<td>1/2(^{13})</td>
<td>5.19e−05</td>
<td>2.00</td>
<td>1.45e−04</td>
<td>1.99</td>
<td>2529.30</td>
</tr>
<tr>
<td>1/2(^{14})</td>
<td>1.30e−05</td>
<td>1.99</td>
<td>4.74e−05</td>
<td>1.61</td>
<td>5078.36</td>
</tr>
<tr>
<td>1/2(^{15})</td>
<td>3.25e−06</td>
<td>1.99</td>
<td>5.81e−05</td>
<td>—</td>
<td>10060.28</td>
</tr>
<tr>
<td>1/2(^{16})</td>
<td>9.03e−07</td>
<td>1.84</td>
<td>1.43e−04</td>
<td>—</td>
<td>20294.98</td>
</tr>
<tr>
<td>1/2(^{17})</td>
<td>6.96e−07</td>
<td>0.37</td>
<td>2.59e−04</td>
<td>—</td>
<td>39789.69</td>
</tr>
<tr>
<td>1/2(^{18})</td>
<td>1.30e−06</td>
<td>—</td>
<td>5.23e−04</td>
<td>—</td>
<td>80955.92</td>
</tr>
</tbody>
</table>
combination of the increment linearization method and LTSEM/pLTSEM is a feasible approach for solving the cubic NLSE.

In order to illustrate the superiority of the parallel LT, we consider the CN scheme in the temporal direction. Using the central difference approximation for \( u_t \) and \( v_t \) in (37), the temporal discretized form becomes

\[
\begin{align*}
\frac{u^{n+1} - u^n}{\tau} &+ \frac{1}{2} \left( \nabla \cdot (K\nabla u^{n+1}) + \nabla \cdot (K\nabla u^n) \right) + \frac{1}{2} u(u^{n+1} + u^n), \\
\frac{1}{2} \left[ v^{n+1} \left( (u^{n+1})^2 + (v^{n+1})^2 \right) + v'(u^n)^2 + (v^n)^2 \right] &= 0, \\
\frac{v^{n+1} - v^n}{\tau} &+ \frac{1}{2} \left( \nabla \cdot (K\nabla v^{n+1}) + \nabla \cdot (K\nabla v^n) \right) - \frac{1}{2} u(u^{n+1} + u^n), \\
\frac{1}{2} \left[ u^{n+1} \left( (u^{n+1})^2 + (v^{n+1})^2 \right) + u'(u^n)^2 + (v^n)^2 \right] &= 0,
\end{align*}
\]

(56)

where \( u^n \) and \( v^n \) evaluate \( u(x, t_n) \) and \( v(x, t_n) \), respectively. Obviously, this scheme is nonlinearly implicit and coupled in the computation, so it requires heavy iterative calculations to solve the resulting implicit algebraic equations at each time step and is not suitable for the parallel computation. Newton’s method of quasilinearization [34] is used to deal with the nonlinearities, and the scheme becomes

\[
\begin{align*}
\frac{u^{n+1,k+1} - u^n}{\tau} &+ \frac{1}{2} \left( \nabla \cdot (K\nabla u^{n+1,k+1}) + \nabla \cdot (K\nabla v^n) \right) + \frac{1}{2} u(u^{n+1,k+1} + u^n) \\
+ \frac{\beta}{2} \left( g_1^{n+1,k} + g_2^{n+1,k} (u^{n+1,k+1} - u^{n+1,k}) + g_3^{n+1,k} (v^{n+1,k+1} - v^{n+1,k}) \right) \\
+ \frac{\beta}{2} \left( (u^n)^2 + (v^n)^2 \right) &= 0, \\
\frac{v^{n+1,k+1} - v^n}{\tau} &+ \frac{1}{2} \left( \nabla \cdot (K\nabla v^{n+1,k+1}) + \nabla \cdot (K\nabla u^n) \right) - \frac{1}{2} u(u^{n+1,k+1} + u^n) \\
+ \frac{\beta}{2} \left( g_4^{n+1,k} + g_5^{n+1,k} (u^{n+1,k+1} - u^{n+1,k}) + g_6^{n+1,k} (v^{n+1,k+1} - v^{n+1,k}) \right) \\
+ \frac{\beta}{2} \left( (u^n)^2 + (v^n)^2 \right) &= 0.
\end{align*}
\]

(57)

with

Table 3: Results of LTSEM and pLTSEM at different times with \( M = 80, \tau = 1 \), and \( N = 26 \), Example 1.

<table>
<thead>
<tr>
<th>LTSEM/ pLTSEM</th>
<th>LTSEM (2/4 cores)</th>
<th>pLTSEM (2 cores)</th>
<th>pLTSEM (4 cores)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( |e (1)|_{L^2} )</td>
<td>( |e (1)|_{H^1} )</td>
<td>( T_{all} ) (s)</td>
<td>( T_{all} ) (s)</td>
</tr>
<tr>
<td>1 4.18e-11</td>
<td>3.63e-09</td>
<td>67.70</td>
<td>42.64</td>
</tr>
<tr>
<td>2 7.37e-10</td>
<td>5.68e-08</td>
<td>136.69</td>
<td>87.09</td>
</tr>
<tr>
<td>3 1.77e-08</td>
<td>1.82e-06</td>
<td>206.68</td>
<td>131.95</td>
</tr>
<tr>
<td>4 2.67e-07</td>
<td>2.02e-05</td>
<td>275.58</td>
<td>175.41</td>
</tr>
<tr>
<td>5 7.16e-06</td>
<td>7.22e-04</td>
<td>343.49</td>
<td>218.89</td>
</tr>
</tbody>
</table>

Figure 2: Errors at \( t = 1 \) versus the total CPU time \( T_{all} \): (a) LTSEM and pLTSEM with \( M = 50, 52, \ldots, 80 \) and \( N = 26 \); (b) CNSEM with \( \tau = (1/2)^i, (1/2)^i, \ldots, (1/2)^i \) and \( N = 26 \), Example 1.
Mathematical Problems in Engineering

Table 4: Results of LTSEM and pLTSEM with various \( \tau \) and \( M = 16, N = 6 \) at \( t = 1 \), Example 2.

<table>
<thead>
<tr>
<th>( \tau )</th>
<th>LTSEM/pLTSEM (2/4 cores)</th>
<th>Order</th>
<th>LTSEM</th>
<th>pLTSEM (2 cores)</th>
<th>pLTSEM (4 cores)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/2</td>
<td>2.63e-01</td>
<td>—</td>
<td>4.73e-01</td>
<td>—</td>
<td>18.64</td>
</tr>
<tr>
<td>1/2(^2)</td>
<td>6.44e-02</td>
<td>2.02</td>
<td>1.13e-01</td>
<td>2.06</td>
<td>37.42</td>
</tr>
<tr>
<td>1/2(^3)</td>
<td>1.72e-02</td>
<td>1.90</td>
<td>3.01e-02</td>
<td>1.91</td>
<td>74.77</td>
</tr>
<tr>
<td>1/2(^4)</td>
<td>4.51e-03</td>
<td>1.93</td>
<td>7.88e-03</td>
<td>1.93</td>
<td>149.80</td>
</tr>
<tr>
<td>1/2(^5)</td>
<td>1.15e-03</td>
<td>1.97</td>
<td>2.01e-03</td>
<td>1.97</td>
<td>301.37</td>
</tr>
<tr>
<td>1/2(^6)</td>
<td>2.91e-04</td>
<td>1.98</td>
<td>5.09e-04</td>
<td>1.98</td>
<td>603.43</td>
</tr>
<tr>
<td>1/2(^7)</td>
<td>7.21e-05</td>
<td>2.01</td>
<td>1.25e-04</td>
<td>2.02</td>
<td>1208.17</td>
</tr>
<tr>
<td>1/2(^8)</td>
<td>1.70e-05</td>
<td>2.08</td>
<td>2.98e-05</td>
<td>2.07</td>
<td>2426.80</td>
</tr>
</tbody>
</table>

Table 5: Results of CNSEM with various \( \tau \) and \( N = 6 \) at \( t = 1 \), Example 2.

<table>
<thead>
<tr>
<th>( \tau )</th>
<th>( |e(1)|_{L_2} )</th>
<th>order</th>
<th>( |e(1)|_{H^1} )</th>
<th>order</th>
<th>( T_{all} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/2</td>
<td>2.74e-01</td>
<td>—</td>
<td>4.74e-01</td>
<td>—</td>
<td>105.16</td>
</tr>
<tr>
<td>1/2(^2)</td>
<td>7.58e-02</td>
<td>1.85</td>
<td>1.31e-01</td>
<td>1.85</td>
<td>167.57</td>
</tr>
<tr>
<td>1/2(^3)</td>
<td>1.94e-02</td>
<td>1.96</td>
<td>3.37e-02</td>
<td>1.95</td>
<td>330.04</td>
</tr>
<tr>
<td>1/2(^4)</td>
<td>4.90e-03</td>
<td>1.98</td>
<td>8.49e-03</td>
<td>1.98</td>
<td>500.05</td>
</tr>
<tr>
<td>1/2(^5)</td>
<td>1.22e-03</td>
<td>2.00</td>
<td>2.12e-03</td>
<td>2.00</td>
<td>992.90</td>
</tr>
<tr>
<td>1/2(^6)</td>
<td>3.07e-04</td>
<td>1.99</td>
<td>5.32e-04</td>
<td>1.99</td>
<td>1980.63</td>
</tr>
<tr>
<td>1/2(^7)</td>
<td>7.67e-05</td>
<td>2.00</td>
<td>1.33e-04</td>
<td>2.00</td>
<td>4030.34</td>
</tr>
<tr>
<td>1/2(^8)</td>
<td>1.92e-05</td>
<td>1.99</td>
<td>3.32e-05</td>
<td>2.00</td>
<td>7896.50</td>
</tr>
</tbody>
</table>

Example 3. Consider the cubic NLSE (2) with a rotational term as follows [35]:

\[
\begin{align*}
\Phi(x, t) & = \frac{i}{2} \frac{\partial \Phi(x, t)}{\partial t} + \frac{1}{2} \frac{\partial^2 \Phi(x, t)}{\partial x^2} + \frac{1}{2} \frac{\partial^2 \Phi(x, t)}{\partial y^2} - |\Phi(x, t)|^2 \Phi(x, t) + w(x) \Phi(x, t) \\
& + i \Theta \left( y \frac{\partial \Phi(x, t)}{\partial x} - x \frac{\partial \Phi(x, t)}{\partial y} \right) = 0.
\end{align*}
\]
(a) \( \| \mathbf{e}(1) \|_{2} \). (b) \( \| \mathbf{e}(1) \|_{H} \).

### Table 6: Results of LTSEM and pLTSEM at different times with \( \tau = 1/2^k, M = 16, \) and \( N = 6, \) Example 2.

| \( t \) | \( \| \mathbf{e}(t) \|_2 \) | \( \| \mathbf{e}(t) \|_{H} \) | \( |M(t) - M(0)| \) | \( |E(t) - E(0)| \) | LTSEM \( T_{\text{all}} (s) \) | pLTSEM (2 cores) \( T_{\text{all}} (s) \) | pLTSEM (4 cores) \( T_{\text{all}} (s) \) |
|------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| 1    | 2.91e-04       | 5.09e-04       | 2.54e-05       | 4.70e-05       | 603.43         | 495.35         | 379.40         |
| 2    | 5.75e-04       | 1.00e-03       | 5.10e-05       | 1.01e-04       | 1207.73        | 981.90         | 754.58         |
| 3    | 8.49e-04       | 1.47e-03       | 7.71e-05       | 1.39e-04       | 1781.71        | 1474.36        | 1138.23        |
| 4    | 1.11e-03       | 1.92e-03       | 1.02e-04       | 1.56e-04       | 2379.17        | 1958.60        | 1521.54        |
| 5    | 1.36e-03       | 2.42e-03       | 1.27e-04       | 1.81e-04       | 2991.63        | 2450.68        | 1900.38        |
| 6    | 4.51e-03       | 7.88e-03       | 1.36e-03       | 2.07e-03       | 3601.52        | 2946.13        | 2281.71        |
| 10   | 8.55e-03       | 1.48e-02       | 2.73e-03       | 4.30e-03       | 6084.16        | 4901.33        | 3790.74        |
| 15   | 1.58e-02       | 2.75e-02       | 2.05e-02       | 2.90e-02       | 9132.54        | 7346.74        | 5686.56        |
| 20   | 1.07e-02       | 1.90e-02       | 2.74e-02       | 3.88e-02       | 12162.66       | 9803.18        | 7585.48        |

Figure 3: Errors at \( t = 1 \) versus the total CPU time \( T_{\text{all}} \) of LTSEM, pLTSEM, and CNSEM with \( \tau = 1/2, 1/2^2, \ldots, 1/2^8, \) Example 2. (a) \( \| \mathbf{e}(1) \|_{2} \). (b) \( \| \mathbf{e}(1) \|_{H} \).

Figure 4: Continued.
which describes the dynamic properties of one-component Bose–Einstein condensates under external rotation \( (\Theta > 0) \). We use the following initial data

\[
\Phi_0(x) = \frac{1}{\sqrt{\pi}} e^{-x^2+y^2/2}.
\]

and take parameters \( \omega(x) = 1.5x^2 + 0.5y^2, \Theta = 0.7 \) on the bounded domain \( \Omega = [-8, 8]^2 \). The problem is solved using pLTSEM with 4 cores. The quadrilateral element mesh with the size of 1 is employed, and the parameters are taken as \( N = 16, \tau = 0.01, \text{and} M = 16 \). Figure 4 shows the numerical solutions \( \text{Re}(\Phi), \text{Im}(\Phi), \text{and} |\Phi| \) at \( t = 1 \).

7. Conclusion

This paper presents an efficient parallel Laplace transform spectral element method (pLTSEM) that greatly outperforms the Crank–Nicolson scheme in the temporal direction for solving the linear Schrödinger equation and the cubic nonlinear Schrödinger equation. The basic idea is to use the Laplace transform method to eliminate the time dependence and apply spectral element method to discretize the subsequent BVPs with different Laplace transform parameters. Since the solving procedures of the discretized systems are independent of each other, they can be handled in the parallel computation for efficiency reason. The parallel computing could be directly realized with the MATLAB parallel computing toolbox. For the cubic nonlinear Schrödinger equation, pLTSEM is also robust when used in conjunction with the increment linearization method. Finally, the approximation is inverted from the transformed domain to the time domain by the numerical inversion of Laplace transform using Talbot’s method. The numerical results indicate that our proposed method is competitive, and good accuracy and speed-up are obtained for both linear and nonlinear problems.

Here, we restrict our attention to a class of Schrödinger equations. This is the first step of applying pLTSEM to solve the time-dependent problems. Actually, the numerical frameworks presented in this work are essential for extension to more complicated cases, for example, the coupled systems of nonlinear Schrödinger equations. Discrete schemes can be established following an analogous analysis to this paper with slight technical modifications.

Data Availability

The datasets used and/or analyzed during the current study are available from the author upon reasonable request.

Conflicts of Interest

The author declares that there are no conflicts of interest or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The author would like to thank Prof Cheng Wang and Prof Yingwei Wang for their helpful advice. The work was supported by the National Natural Science Foundation of China (nos.12001362 and 11526132) and the Natural Science Foundation of Shanghai (no.16ZR1412700).

References


