Journal of Computational Physics 393 (2019) 313–336

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

A fast compact time integrator method for a family of general order semilinear evolution equations

Jianguo Huang [a,1], Lili Ju [b,*,2], Bo Wu [a]

[a] School of Mathematical Sciences, and MOE-LSC, Shanghai Jiao Tong University, Shanghai 200240, PR China
[b] Department of Mathematics, University of South Carolina, Columbia, SC 29208, USA

ARTICLE INFO

Article history:
Received 15 May 2018
Received in revised form 5 May 2019
Accepted 8 May 2019
Available online 15 May 2019

Keywords:
Semilinear evolution equation
Time integrator
Compact difference scheme
Fast Fourier transform
Multistep approximation
Sine-Gordon equation

ABSTRACT

In this paper we develop a fast compact time integrator method for numerically solving a family of general order semilinear evolution equations in regular domains. The spatial discretization is carried out by a fourth-order accurate compact difference scheme in which fast Fourier transform can be utilized for efficient implementation. The resulting semi-discretized problem consists of a system of ordinary differential equations whose solution can be explicitly expressed in term of time integrators, and a desired numerical method is then obtained by further adopting multistep approximations of the nonlinear terms based on the solution formula. Linear stability analysis is performed for the method for second-order in time evolution equations. Extensive numerical experiments with applications are also presented to demonstrate efficiency, accuracy, and stability of the proposed method in practice.

1. Introduction

Many important problems in science and engineering can be modeled as a n-th order in time semilinear evolution equation of the following standard form:

$$\begin{cases} L_{n,t}(u(\boldsymbol{x},t)) = D\triangle u(\boldsymbol{x},t) + f(u,\boldsymbol{x},t), & (\boldsymbol{x},t)\in\Omega\times(0,T], \\ \dfrac{\partial^k u}{\partial t^k}(\boldsymbol{x},0) = u_0^k(\boldsymbol{x}), & \boldsymbol{x}\in\Omega,\ 0\le k\le n-1, \end{cases} \tag{1.1}$$

where $\Omega\in\mathbb{R}^d$ is an open domain, the diffusion coefficient $D>0$, and the linear operator

$$L_{n,t}(u) := \frac{\partial^n u}{\partial t^n} + a_1\frac{\partial^{n-1}u}{\partial t^{n-1}} + \cdots + a_{n-1}\frac{\partial u}{\partial t} \tag{1.2}$$

with $\{a_k\}_{k=1}^{n-1}$ being real numbers. Assume that Dirichlet or periodic boundary conditions are imposed. The equation (1.1) with $n=1$ or 2 includes some typical mathematical equations in a variety of real-world applications, for instance, diffusion-

* Corresponding author.
E-mail addresses: jghuang@sjtu.edu.cn (J. Huang), ju@math.sc.edu (L. Ju), sanshiyayan@sjtu.edu.cn (B. Wu).

[1] J. Huang's research has been partially supported by National Natural Science Foundation of China under grant number 11571237.
[2] L. Ju's research has been partially supported by US National Science Foundation under grant numbers DMS-1521965 and DMS-1818438.

https://doi.org/10.1016/j.jcp.2019.05.013
0021-9991/

reaction problems in chemical reactions and population dynamics [28,46], the Allen-Cahn equation for phase transition/separation modeling [2], hyperbolic equations [19] such as the sine-Gordon equation [8] in differential geometry and relativistic field theory; for $n = 3$, an equation with slightly modified form can be found in [34,39], which is used to describe the propagation of weakly nonlinear waves in relaxing media.

During the past decades, a lot of researches were devoted to developing numerical methods for numerically solving the problem (1.1). Finite element method [37], finite volume method [5,29], and differential quadrature method [19,33] were frequently used for its spatial discretization, and in particular compact difference schemes have been a very popular approach in recent years due to their flexibility and spectral-like resolution. In [25], Li and Tang developed a fourth-order compact difference scheme to solve a vorticity equation, which was also used to solve one dimensional Burger's equation in [27]. In [44], Xie et al. proposed two compact difference schemes to solve the one-dimensional nonlinear Schrödinger equation. In [9] Ding et al. introduced a three-level compact difference scheme for two-dimensional second-order hyperbolic equation. The stability and convergence of a compact difference for the heat conduction problem with Neumann boundary conditions was studied in [36]. Since the time discretizations are all done by implicit schemes in these methods, solutions of large-scale linear systems are needed, which is often time-consuming in general. Thanks to the development of algorithms for fast matrix inversion, compact difference schemes are proving more advantageous when the solution domain is regular. For instance, Wang at al. [41] proposed a fast implementation for the Poisson equation by using the fast discrete sine transform, and in [26] Li et al. further implemented this compact difference scheme in an adaptive refinement framework.

On the other hand, the exponential integrator-based methods have enjoyed great popularity due to its numerical stability and high-order accuracy for time discretization of stiff problems. The integrating factor (IF) method [20,24,31,32,40] applies quadrature rules directly to the whole integrands, while the exponential time differencing (ETD) method utilizes the Runge-Kutta or multistep approaches to approximate the nonlinear terms and then compute the resulting integrals exactly [6,15,16,21,23,42,43]. In [11,12], Du et al. studied stability properties of the ETD method for parabolic equations. More recently, Ju et al. [22] proposed a discrete fast Fourier transform (FFT) based algorithm to efficiently solve a wide class of semilinear parabolic equations, where the second-order central compact difference and the explicit multistep exponential time integration was integrated together to discretize the spatial and temporal variables. Later in [47] efficient numerical methods were proposed which combines exponential time differencing Runge-Kutta approximations with a fourth-order compact difference for spatial discretization.

However, to the best of our knowledge, there are few efficient numerical methods for high-order in time ($n \geq 2$) semilinear evolution problems such as (1.1). A routine way is to reformulate the original problem as a first-order system by introducing auxiliary variables and then devise numerical methods for solving the produced system. A weakness of such an approach is that the size of the discrete problem could become very large, leading to lower computational efficiency. In this paper, we propose a fast compact time integrator (FCTI) method for solving the equation (1.1). We use a fourth-order accurate compact difference method for spatial discretization, yielding a diagonalizable matrix system of ordinary differential equations (ODEs) whose solution can be explicitly expressed in term of the time integrator. By approximating the nonlinear terms in the integrands using Lagrange interpolation functions, and then performing exact integrations just like the approach used by the ETD method, we obtain a fully discrete scheme for solving the model problem. This technique can be viewed as a generalization of the fast compact exponential integrator method studied in [22]. Following the ideas in [11,12], we also perform a linear stability analysis of the method for the second order evolution equation. Furthermore, the proposed FCTI method can also be easily adapted with minor modifications to solve some nonstandard high order semilinear evolution equations efficiently as derived in the appendix.

The rest of the paper is organized as follows. In Section 2, we derive the FCTI method with details for two types of boundary conditions (Dirichlet and periodic) in two dimensions and then present its extension to three dimensions in Section 3. In Section 4, we present the linear stability analysis of the method for the second order evolution equation. Extensive experiments with applications are also reported to numerically demonstrate accuracy and efficiency of the proposed method in Section 5. Some concluding remarks are finally given in Section 6.

2. A compact time integrator method in two dimensions and fast implementation

Let us consider the model problem (1.1) in an open rectangular domain in the two-dimensional space

$$\Omega = \{x_b < x < x_e, y_b < y < y_e\}.$$

We partition the spatial domain Ω by a rectangular grid which is uniform in each direction, i.e., $h_x = \frac{x_e - x_b}{N_x}, h_y = \frac{y_e - y_b}{N_y}$, respectively. Grid points are defined as $(x_i, y_j) = (x_b + ih_x, y_b + jh_y)$ for $0 \leq i \leq N_x$ and $0 \leq j \leq N_y$ and denote the semi-discretized (in space) solution by $u_{i,j} = u_{i,j}(t) \approx u(x_i, y_j, t)$. Similarly, $u^{xx}_{i,j}$ and $u^{yy}_{i,j}$ are denoted as the semi-discretized approximations of the second-order partial derivatives $u_{xx}(x_i, y_j, t)$ and $u_{yy}(x_i, y_j, t)$, respectively.

2.1. Spatial discretization: a fourth-order compact finite difference and discrete sine transform

Assume that the Dirichlet boundary condition is imposed as follows:

$$u = u^b, \qquad (x, y) \in \partial\Omega,\ t \in [0, T]. \tag{2.1}$$

We shall construct a spatial discretization which is fourth-order accurate in space by using the compact finite difference scheme [41,45] as shown below:

$$\begin{cases} \frac{1}{12}(u^{xx}_{i-1,j} + 10u^{xx}_{i,j} + u^{xx}_{i+1,j}) = \frac{1}{h_x^2}(u_{i-1,j} - 2u_{i,j} + u_{i+1,j}), \\ \frac{1}{12}(u^{yy}_{i,j-1} + 10u^{yy}_{i,j} + u^{yy}_{i,j+1}) = \frac{1}{h_y^2}(u_{i,j-1} - 2u_{i,j} + u_{i,j+1}), \end{cases} \tag{2.2}$$

for $i = 1, 2, \cdots N_x - 1$, $j = 1, 2, \cdots, N_y - 1$. In order to write (2.2) in the two-dimensional array (i.e., matrix) form, we introduce the following notations: the approximate solution matrix

$$\boldsymbol{U} = \begin{pmatrix} u_{1,1} & u_{1,2} & \dots & u_{1,N_y-1} \\ u_{2,1} & u_{2,2} & \dots & u_{2,N_y-1} \\ \vdots & \vdots & \ddots & \vdots \\ u_{N_x-1,1} & u_{N_x-1,2} & \dots & u_{N_x-1,N_y-1} \end{pmatrix}_{(N_x-1)\times(N_y-1)} \tag{2.3}$$

and similarly the corresponding second order derivatives matrices along x-axis and y-axis, $\boldsymbol{U}^{xx}$ and $\boldsymbol{U}^{yy}$ respectively. We also define two special operators "$\circledast$" and "$\circledcirc$" as follows [20]: for any $\boldsymbol{A}_x \in \mathbb{R}^{(N_x-1)\times(N_x-1)}$ and $\boldsymbol{A}_y \in \mathbb{R}^{(N_y-1)\times(N_y-1)}$,

$$(\boldsymbol{A}_x \circledast \boldsymbol{U})_{i,j} = \sum_{l=1}^{N_x-1} (\boldsymbol{A}_x)_{i,l} u_{l,j}, \qquad (\boldsymbol{A}_y \circledcirc \boldsymbol{U})_{i,j} = \sum_{l=1}^{N_y-1} (\boldsymbol{A}_y)_{j,l} u_{i,l}. \tag{2.4}$$

Let

$$\boldsymbol{A}^d_{P\times P} = \begin{pmatrix} 10 & 1 & & \\ 1 & 10 & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & 10 \end{pmatrix}_{P\times P}, \qquad \boldsymbol{B}^d_{P\times P} = \begin{pmatrix} -2 & 1 & & \\ 1 & -2 & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & -2 \end{pmatrix}_{P\times P},$$

and set

$$\begin{aligned} \boldsymbol{A}_x &= \tfrac{1}{12}\boldsymbol{A}^d_{(N_x-1)\times(N_x-1)}, \quad \boldsymbol{A}_y = \tfrac{1}{12}\boldsymbol{A}^d_{(N_y-1)\times(N_y-1)}, \\ \boldsymbol{B}_x &= \tfrac{D}{h_x^2}\boldsymbol{B}^d_{(N_x-1)\times(N_x-1)}, \quad \boldsymbol{B}_y = \tfrac{D}{h_y^2}\boldsymbol{B}^d_{(N_y-1)\times(N_y-1)}. \end{aligned} \tag{2.5}$$

Then one can check that (2.2) can be written in the following compact representation:

$$\begin{cases} D\boldsymbol{A}_x \circledast \boldsymbol{U}^{xx} + \boldsymbol{U}_{x2} = \boldsymbol{B}_x \circledast \boldsymbol{U} + \boldsymbol{U}_{x0}, \\ D\boldsymbol{A}_y \circledcirc \boldsymbol{U}^{yy} + \boldsymbol{U}_{y2} = \boldsymbol{B}_y \circledcirc \boldsymbol{U} + \boldsymbol{U}_{y0}, \end{cases} \tag{2.6}$$

where

$$\boldsymbol{U}_{x0} = \frac{D}{h_x^2}\begin{pmatrix} u_{0,1} & u_{0,2} & \dots & u_{0,N_y-1} \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 0 \\ u_{N_x,1} & u_{N_x,2} & \dots & u_{N_x,N_y-1} \end{pmatrix}_{(N_x-1)\times(N_y-1)}, \quad \boldsymbol{U}_{y0} = \frac{D}{h_y^2}\begin{pmatrix} u_{1,0} & 0 & \dots & 0 & u_{1,N_y} \\ u_{2,0} & 0 & \dots & 0 & u_{2,N_y} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ u_{N_x-1,0} & 0 & \dots & 0 & u_{N_x-1,N_y} \end{pmatrix}_{(N_x-1)\times(N_y-1)},$$

$$\boldsymbol{U}_{x2} = \frac{D}{12}\begin{pmatrix} u^{xx}_{0,1} & u^{xx}_{0,2} & \dots & u^{xx}_{0,N_y-1} \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 0 \\ u^{xx}_{N_x,1} & u^{xx}_{N_x,2} & \dots & u^{xx}_{N_x,N_y-1} \end{pmatrix}_{(N_x-1)\times(N_y-1)}, \quad \boldsymbol{U}_{y2} = \frac{D}{12}\begin{pmatrix} u^{yy}_{1,0} & 0 & \dots & 0 & u^{yy}_{1,N_y} \\ u^{yy}_{2,0} & 0 & \dots & 0 & u^{yy}_{2,N_y} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ u^{yy}_{N_x-1,0} & 0 & \dots & 0 & u^{yy}_{N_x-1,N_y} \end{pmatrix}_{(N_x-1)\times(N_y-1)}.$$

Thus we obtain a semi-discretization in space of the model equation (1.1) as follows:

$$\boldsymbol{L}_{n,t}(\boldsymbol{U}) - (\boldsymbol{A}_x^{-1} \circledast \boldsymbol{B}_x \circledast \boldsymbol{U} + \boldsymbol{A}_y^{-1} \circledcirc \boldsymbol{B}_y \circledcirc \boldsymbol{U}) = \boldsymbol{F}(\boldsymbol{U}, t) + \boldsymbol{W}(t), \tag{2.7}$$

where

$$\begin{aligned}
&\boldsymbol{L}_{n,t}(\boldsymbol{U}) = \boldsymbol{U}^{(n)}(t) + a_1\boldsymbol{U}^{(n-1)}(t) + \cdots + a_{n-1}\boldsymbol{U}'(t),\\
&\mathcal{F}(\boldsymbol{U},t) = (f(u_{i,j}(t), x_i, y_j, t))_{(N_x-1)\times(N_y-1)},\\
&\boldsymbol{W}(t) = \boldsymbol{A}_x^{-1}\circledS_x(\boldsymbol{U}_{x0} - \boldsymbol{U}_{x2}) + \boldsymbol{A}_y^{-1}\circledS_y(\boldsymbol{U}_{y0} - \boldsymbol{U}_{y2}).
\end{aligned}$$

Remark 1. From the Dirichlet boundary condition (2.1), it is clear that $u_{0,j} = u^b_{0,j} = u^b(x_0, y_j, t)$, $u_{N_x,j} = u^b_{N_x,j} = u^b(x_{N_x}, y_j, t)$, $u_{i,0} = u^b_{i,0} = u^b(x_i, y_0, t)$, and $u_{i,N_y} = u^b_{i,N_y} = u^b(x_i, y_{N_y}, t)$ in $\boldsymbol{U}_{x0}$ and $\boldsymbol{U}_{y0}$. To get the corresponding values of $\boldsymbol{U}_{x2}$ and $\boldsymbol{U}_{y2}$, we can make use of the equation (1.1) together with the boundary condition (2.1). For example, to calculate $u^{xx}_{0,1}$, notice that the equation holds at $x = 0$ and one then obtains

$$\begin{cases}
L_{n,t}(u^b(x_0, y_1, t)) = D(u^{xx}_{0,1} + u^{yy}_{0,1}) + f(u^b, x_0, y_1, t),\\
u^{yy}_{0,1} = (u^b)^{yy}_{0,1},
\end{cases} \tag{2.8}$$

which yields $u^{xx}_{0,1} = \frac{1}{D}(L_{n,t}(u^b(x_0, y_1, t)) - f(u^b, x_0, y_1, t)) - (u^b)^{yy}_{0,1}$.

It is easy to find that there exist the following spectral decompositions

$$\boldsymbol{A}_x = \boldsymbol{P}_x\tilde{\boldsymbol{D}}_{a,x}\boldsymbol{P}_x^{-1}, \qquad \boldsymbol{A}_y = \boldsymbol{P}_y\tilde{\boldsymbol{D}}_{a,y}\boldsymbol{P}_y^{-1}, \qquad \boldsymbol{B}_x = \boldsymbol{P}_x\tilde{\boldsymbol{D}}_{b,x}\boldsymbol{P}_x^{-1}, \qquad \boldsymbol{B}_y = \boldsymbol{P}_y\tilde{\boldsymbol{D}}_{b,y}\boldsymbol{P}_y^{-1},$$

where $\tilde{\boldsymbol{D}}_{a,x}$, $\tilde{\boldsymbol{D}}_{a,y}$ and $\tilde{\boldsymbol{D}}_{b,x}$, $\tilde{\boldsymbol{D}}_{b,y}$ are diagonal matrices given by

$$\begin{aligned}
&\tilde{\boldsymbol{D}}_{a,x} = \mathrm{diag}(d_1^{a,x}, d_2^{a,x}, \cdots, d_{N_x-1}^{a,x}), \quad \tilde{\boldsymbol{D}}_{a,y} = \mathrm{diag}(d_1^{a,y}, d_2^{a,y}, \cdots, d_{N_y-1}^{a,y}),\\
&\tilde{\boldsymbol{D}}_{b,x} = \mathrm{diag}(d_1^{b,x}, d_2^{b,x}, \cdots, d_{N_x-1}^{b,x}), \quad \tilde{\boldsymbol{D}}_{b,y} = \mathrm{diag}(d_1^{b,y}, d_2^{b,y}, \cdots, d_{N_y-1}^{b,y}),
\end{aligned}$$

with

$$d_i^{a,x} = 1 - \tfrac{1}{3}\sin^2(\tfrac{i\pi}{2N_x}), \quad d_i^{b,x} = -\tfrac{4D}{h_x^2}\sin^2(\tfrac{i\pi}{2N_x}), \quad d_j^{a,y} = 1 - \tfrac{1}{3}\sin^2(\tfrac{j\pi}{2N_y}), \quad d_j^{b,y} = -\tfrac{4D}{h_y^2}\sin^2(\tfrac{j\pi}{2N_y}),$$

and $\boldsymbol{P}_x$ and $\boldsymbol{P}_y$ are orthonormal matrices consisting of corresponding eigenvectors. Plugging the above equations into (2.7), and multiplying $\boldsymbol{P}_x^{-1}$ from the left hand side and $(\boldsymbol{P}_y^{-1})^T$ from the right hand side, we immediately obtain

$$\boldsymbol{L}_{n,t}(\boldsymbol{V}) - (\tilde{\boldsymbol{D}}_{a,x}^{-1}\circledS_x\tilde{\boldsymbol{D}}_{b,x}\circledS_x\boldsymbol{V} + \tilde{\boldsymbol{D}}_{a,y}^{-1}\circledS_y\tilde{\boldsymbol{D}}_{b,y}\circledS_y\boldsymbol{V}) = \boldsymbol{P}_y^{-1}\circledS_y\boldsymbol{P}_x^{-1}\circledS_x(\mathcal{F}(\boldsymbol{P}_y\circledS_y\boldsymbol{P}_x\circledS_x\boldsymbol{V}, t) + \boldsymbol{W}(t)), \tag{2.9}$$

where $\boldsymbol{V} = \boldsymbol{P}_y^{-1}\circledS_y\boldsymbol{P}_x^{-1}\circledS_x\boldsymbol{U}$. We also note that the operations $\boldsymbol{P}_y\circledS_y\boldsymbol{P}_x\circledS_x\boldsymbol{V}$ and $\boldsymbol{P}_y^{-1}\circledS_y\boldsymbol{P}_x^{-1}\circledS_x\boldsymbol{V}$ are exactly the two-dimensional discrete sine transform (DST) and the inverse DST respectively, and can be efficiently calculated by Fast Fourier Transform (FFT) [22]. It is easy to check that (2.9) is a matrix system of ODEs. Define $\boldsymbol{H} = (h_{i,j})_{(N_x-1)\times(N_y-1)}$ with

$$h_{i,j} = \frac{d_i^{b,x}}{d_i^{a,x}} + \frac{d_j^{b,y}}{d_j^{a,y}},$$

and denote by the special operator "$\odot$" the element-wise multiplication between two matrices of the same size. Then the equation (2.9) can be rewritten as

$$\mathcal{L}(\boldsymbol{V}) = \boldsymbol{f}(\boldsymbol{V}, t),$$

where

$$\begin{aligned}
&\mathcal{L}(\boldsymbol{V}) := \boldsymbol{V}^{(n)} + a_1\boldsymbol{V}^{(n-1)} + \cdots + a_{n-1}\boldsymbol{V}' - \boldsymbol{H}\odot\boldsymbol{V},\\
&\boldsymbol{f}(\boldsymbol{V}, t) := \boldsymbol{P}_y^{-1}\circledS_y\boldsymbol{P}_x^{-1}\circledS_x\left[\boldsymbol{F}(\boldsymbol{P}_y\circledS_y\boldsymbol{P}_x\circledS_x\boldsymbol{V}, t) + \boldsymbol{W}(t)\right],
\end{aligned}$$

with $\boldsymbol{V}^{(k)} = \frac{d^k\boldsymbol{V}}{dt^k}$ for $k = 1, 2, \cdots, n$.

We are now left with solving the following system of ODEs:

$$\begin{cases}
\mathcal{L}(\boldsymbol{V}) = \boldsymbol{f}(\boldsymbol{V}, t), \qquad t \in (0, T),\\
\boldsymbol{V}(0) = \boldsymbol{V}_0^{(0)},\ \boldsymbol{V}'(0) = \boldsymbol{V}_0^{(1)},\ \cdots,\ \boldsymbol{V}^{(n-1)}(0) = \boldsymbol{V}_0^{(n-1)}
\end{cases} \tag{2.10}$$

where $\boldsymbol{V}_0^{(k)} = \boldsymbol{P}_y^{-1}\circledS_y\boldsymbol{P}_x^{-1}\circledS_x\boldsymbol{U}_0^{(k)}$ with $\boldsymbol{U}_0^{(k)} = \left(u_0^k(x_i, y_j)\right)_{(N_x-1)\times(N_y-1)}$.

Remark 2. For the purpose of numerical stabilization, a linear splitting scheme sometimes could be used as suggested in [22,31] for the semilinear parabolic equation $(n = 1)$ to get $\tilde{\mathcal{L}}(\boldsymbol{V}) = \tilde{\boldsymbol{f}}(\boldsymbol{V}, t)$ with

$$\tilde{\mathcal{L}}(\boldsymbol{V}) := \mathcal{L} + \kappa \boldsymbol{V}, \qquad \tilde{\boldsymbol{f}}(\boldsymbol{V}, t) := \boldsymbol{f}(\boldsymbol{V}, t) + \kappa \boldsymbol{V},$$

where the parameter κ is often required to be at least

$$\kappa \geq \frac{1}{2} \max_u \{0, f_u(u)\}.$$

2.2. Temporal discretization: a time integrator multistep approximation

Next we develop a time integrator approach for temporal discretization of the semi-discrete in space problem (2.10). Our key idea is first to obtain an explicit formulation for the solution of (2.10) with the help of time integrators, then make use of multistep approximations of the nonlinear terms and perform exact integrations.

2.2.1. The time integrator formula

Let us recall some well-known results from the theory of ODEs. First we consider a homogeneous ODE in the following form:

$$\hat{L}_{n,t}(y) = y^{(n)} + a_1 y^{(n-1)} + a_2 y^{(n-2)} + \cdots + a_{n-1} y' + a_n y = 0. \tag{2.11}$$

Its characteristic equation is given by

$$p(\lambda) = \lambda^n + a_1 \lambda^{n-1} + a_2 \lambda^{n-2} + \cdots + a_{n-1}\lambda + a_n = 0. \tag{2.12}$$

Lemma 1 *(Theorem 2.27, [3]). Suppose that the characteristic equation (2.12) has $k^* \geq 1$ distinct roots $\mu_1, \mu_2, \ldots, \mu_{k^*}$ (could be real or complex valued), each of which has the multiplicity respectively equal to $n_1, n_2, \cdots, n_{k^*}$ ($\sum_{k=1}^{k^*} n_k = n$). Then the initial value problem*

$$\begin{cases} \hat{L}_{n,t}(y) = 0, \\ y(0) = y'(0) = \cdots = y^{(n-2)}(0) = 0, \quad y^{(n-1)}(0) = 1 \end{cases}$$

has a unique solution $g(t)$ given by

$$g(t) = P_1(t)\mathrm{e}^{\mu_1 t} + \cdots + P_{k^*}(t)\mathrm{e}^{\mu_{k^*} t},$$

where

$$P_k(t) = c_{1,k} + c_{2,k} t + \cdots + c_{n_k,k} t^{n_k - 1}, \qquad 1 \leq k \leq k^*.$$

Note that we just need solve a $n \times n$ linear system determined from the coefficients of $\hat{L}_{n,t}(y)$ to find all values of $\{c_{1,k}, c_{2,k}, \cdots, c_{n_k,k}\}_{k=1}^{k^*}$ and thus get g.

Lemma 2 *(Theorem 2.19, [3]). Suppose that the differential operator $\hat{L}_{n,t}$ satisfies all the conditions given in Lemma 1. Let $m_k \in \mathbb{R}$, $0 \leq k \leq n-1$ and $t_* \in [0, T]$. Then the initial value problem*

$$\begin{cases} \hat{L}_{n,t}(y) = f(y, t), \\ y(t_*) = m_0,\ y'(t_*) = m_1,\ \cdots,\ y^{(n-1)}(t_*) = m_{n-1} \end{cases}$$

has a unique solution $y(t)$ given by

$$y(t) = \int_{t_*}^{t} g(t-s) f(y(s), s)\, \mathrm{d}s + \sum_{k=0}^{n-1} m_k y_k^*(t - t_*), \tag{2.13}$$

where

$$\begin{cases} y_0^*(t) = a_{n-1} g(t) + a_{n-2} g'(t) + \cdots + a_2 g^{(n-3)}(t) + a_1 g^{(n-2)}(t) + g^{(n-1)}(t), \\ y_1^*(t) = a_{n-2} g(t) + a_{n-3} g'(t) + \cdots + a_1 g^{(n-3)}(t) + g^{(n-2)}(t), \\ \cdots \\ y_{n-2}^*(t) = a_1 g(t) + g'(t), \\ y_{n-1}^*(t) = g(t). \end{cases} \tag{2.14}$$

Observing each (i, j)-entry of the system (2.10) is an independent ODE corresponding to the differential operator $\hat{L}_{n,t}$ in (2.11) with $a_n = -h_{i,j}$, thus we can use Lemmas 1 and 2 to derive its solution explicitly. Let us write the function g in Lemma 1 as $g_{i,j}$, and the corresponding y_k^* in (2.14) with $t_* = 0$ as $y_{i,j}^k$. Denote $\boldsymbol{g} := (g_{i,j})$ and $\boldsymbol{y}_k = (y_{i,j}^k)$. Then we can obtain an explicit solution formulation for the semi-discrete problem (2.10) as follows.

Corollary 3. *The initial value problem* (2.10) *has a unique solution* $\boldsymbol{V}(t) = (v_{i,j})_{(N_x-1)\times(N_y-1)}$ *given by*

$$\boldsymbol{V}(t) = \int_0^t \boldsymbol{g}(t-s) \odot \boldsymbol{f}(\boldsymbol{V}(s), s)\, \mathrm{d}s + \sum_{k=0}^{n-1} \boldsymbol{y}_k(t) \odot \boldsymbol{V}_0^{(k)}. \tag{2.15}$$

Note that $g_{ij}(0) = g'_{ij}(0) = \cdots = g_{ij}^{(n-2)}(0) = 0$, so we further have

$$\boldsymbol{V}^{(l)}(t) = \int_0^t \boldsymbol{g}^{(l)}(t-s) \odot \boldsymbol{f}(\boldsymbol{V}(s), s)\, \mathrm{d}s + \sum_{k=0}^{n-1} \boldsymbol{y}_k^{(l)}(t) \odot \boldsymbol{V}_0^{(k)}, \qquad 1 \le l \le n-1. \tag{2.16}$$

Next let us discretize the time interval $[0, T]$ by $t_m = m\Delta t$ for $m = 0, 1, \cdots, N_t$ with $\Delta t = T/N_t$. Then a recursive time integrator formula for the solution of (2.10) can be expressed as: for $m = 0, 1, \cdots, N_t - 1$,

$$\boldsymbol{V}^{(l)}(t_{m+1}) = \int_{t_m}^{t_{m+1}} \boldsymbol{g}^{(l)}(t_{m+1}-s) \odot \boldsymbol{f}(\boldsymbol{V}(s), s)\, \mathrm{d}s + \sum_{k=0}^{n-1} \boldsymbol{y}_k^{(l)}(\Delta t) \odot \boldsymbol{V}^{(k)}(t_m), \qquad 0 \le l \le n-1. \tag{2.17}$$

By the change of variable $s \to t_m + \tau$, we can rewrite (2.17) as

$$\boldsymbol{V}^{(l)}(t_{m+1}) = \int_0^{\Delta t} \boldsymbol{g}^{(l)}(\Delta t - \tau) \odot \boldsymbol{f}(\boldsymbol{V}(t_m+\tau), t_m+\tau)\, \mathrm{d}\tau + \sum_{k=0}^{n-1} \boldsymbol{y}_k^{(l)}(\Delta t) \odot \boldsymbol{V}^{(k)}(t_m), \qquad 0 \le l \le n-1. \tag{2.18}$$

Finally, at each time step t_m, the exact solution of the semi-discrete in space system (2.7) satisfies $\boldsymbol{U}^{(l)}(t_m) = \boldsymbol{P}_y \circledcirc \boldsymbol{P}_x \circledast \boldsymbol{V}^{(l)}(t_m)$.

2.2.2. Evaluation of the integrals based on multistep approximations

In order to obtain a fully discrete solution $\boldsymbol{V}_m^{(l)} \approx \boldsymbol{V}^{(l)}(t_m)$, we are left with the problem of evaluating the integrals on the right side hand of (2.18). We will adopt the multistep approach. Note that the integrand is $\boldsymbol{g}^{(l)}(\Delta t - \tau) \odot \boldsymbol{f}(\boldsymbol{V}(t_m+\tau), t_m+\tau)$ with

$$\boldsymbol{f}(\boldsymbol{V}(s), s) = \boldsymbol{P}_y^{-1} \circledcirc \boldsymbol{P}_x^{-1} \circledast \boldsymbol{F}(\boldsymbol{P}_y \circledcirc \boldsymbol{P}_x \circledast \boldsymbol{V}(s), s) + \boldsymbol{P}_y^{-1} \circledcirc \boldsymbol{P}_x^{-1} \circledast \boldsymbol{W}(s).$$

Adams-Moulton approximation of the integral related to the boundary condition term $\boldsymbol{W}$. We first evaluate the integrals associated with the inhomogeneous boundary condition

$$\boldsymbol{Q}_l^W = (q_{i,j}^{W,l})_{(N_x-1)\times(N_y-1)} = \int_0^{\Delta t} \boldsymbol{g}^{(l)}(\Delta t - \tau) \odot \left(\boldsymbol{P}_y^{-1} \circledcirc \boldsymbol{P}_x^{-1} \circledast \boldsymbol{W}(t_m+\tau)\right) \mathrm{d}\tau.$$

Notice that each term of $g_{ij}(\Delta t - \tau)$ has the form of $t^k \mathrm{e}^{\lambda(\Delta t - \tau)}$, we will develop accurate evaluations of the above integrals using the approach taken in [22]. Let $\overline{\boldsymbol{W}}(t_m+\tau) = \boldsymbol{P}_y^{-1} \circledcirc \boldsymbol{P}_x^{-1} \circledast \boldsymbol{W}(t_m+\tau)$. We use the Lagrange interpolation polynomial $\boldsymbol{P}_r^W(\tau)$ of degree r to approximate $\overline{\boldsymbol{W}}(t_m+\tau)$, based on the values of $\overline{\boldsymbol{W}}(t)$ at $t_{m+1}, t_m, \ldots, t_{m+1-r}$, i.e.,

$$\boldsymbol{P}_r^W(t_m+\tau) = \sum_{s=-1}^{r-1} w_{r,s}(\tau) \overline{\boldsymbol{W}}(t_{m-s}),$$

where $w_{r,s}(\tau) = \prod_{k=-1, k\neq s}^{r-1} \frac{\tau + k\Delta t}{(k-s)\Delta t}$. Thus we can get

$$q_{i,j}^{W,l} \approx \sum_{s=-1}^{r-1} \overline{w}_{i,j}(t_{m-s}) \left(\int_0^{\Delta t} g_{i,j}^{(l)}(\Delta t - \tau) w_{r,s}(\tau)\, \mathrm{d}\tau \right) = \sum_{s=-1}^{r-1} \overline{w}_{i,j}(t_{m-s}) \alpha_{i,j}^{(r,s,l)}.$$

Note that $\alpha_{i,j}^{(r,s,l)}$ is independent of the time steps t_m with a uniform time step size Δt being used. For simplicity, we only present the values of $\alpha_{i,j}^{(r,s,l)}$ for $r = 0, 1, 2$:

$$\begin{aligned}
&\alpha_{i,j}^{(0,-1,l)} = \phi_{i,j}^{(0,l)},\\
&\alpha_{i,j}^{(1,-1,l)} = \phi_{i,j}^{(1,l)}, \alpha_{i,j}^{(1,0,l)} = \phi_{i,j}^{(0,l)} - \phi_{i,j}^{(1,l)},\\
&\alpha_{i,j}^{(2,-1,l)} = \frac{1}{2}(\phi_{i,j}^{(1,l)} + \phi_{i,j}^{(2,l)}), \alpha_{i,j}^{(2,0,l)} = \phi_{i,j}^{(0,l)} - \phi_{i,j}^{(2,l)}, \alpha_{i,j}^{(2,1,l)} = -\frac{1}{2}(\phi_{i,j}^{(1,l)} - \phi_{i,j}^{(2,l)}),
\end{aligned} \tag{2.19}$$

where

$$\phi_{i,j}^{(r,l)} = \int_0^{\Delta t} g_{i,j}^{(l)}(\Delta t - \tau)\left(\frac{\tau}{\Delta t}\right)^r d\tau. \tag{2.20}$$

Denote $\boldsymbol{S}_{r,s,l}^{W} = (\alpha_{i,j}^{(r,s,l)})_{(N_x-1)\times(N_y-1)}$, then we have the approximation of

$$\boldsymbol{Q}_l^{\,W} \approx \sum_{s=-1}^{r-1} \overline{\boldsymbol{W}}_{m-s} \odot \boldsymbol{S}_{r,s,l}^{W}, \tag{2.21}$$

which is $(r+1)$-th order accurate in time.

Adams-Bashforth approximation of the integral related to the nonlinear term $\boldsymbol{F}$. Now we evaluate the integrals $\boldsymbol{Q}_r^{\,F}$ resulted from the nonlinear reaction and source term

$$\boldsymbol{Q}_l^{\,F} = (q_{i,j}^{F,l})_{(N_x-1)\times(N_y-1)} = \int_0^{\Delta t} \boldsymbol{g}^{(l)}(\Delta t - \tau) \odot \boldsymbol{F}(t_m + \tau)\, \mathrm{d}\tau$$

where $\boldsymbol{F}(t_m + \tau) = \boldsymbol{P}_y^{-1} \circledcirc \boldsymbol{P}_x^{-1} \circledast \boldsymbol{F}(\boldsymbol{P}_y \circledcirc \boldsymbol{P}_x \circledast \boldsymbol{V}(t_m + \tau), t_m + \tau)$. We use an explicit multistep approach since $\boldsymbol{V}(s)$ is unknown at time t_{m+1}, i.e., we interpolate $\boldsymbol{F}(t_m + \tau)$ using its values at $t_m, t_{m-1}, \ldots, t_{m-r}$. The corresponding Lagrange interpolation polynomial $P_r^F(\tau)$ of degree r to approximate $\boldsymbol{F}(t_m + \tau)$ is

$$\boldsymbol{P}_r^F(t_m + \tau) = \sum_{s=0}^{r} \eta_{r,s}(\tau) \boldsymbol{F}(t_{m-s})$$

with $\eta_{r,s}(\tau) = \prod_{k=0,k\neq s}^{r} \frac{\tau + k\Delta t}{(k-s)\Delta t}$.

Thus we can approximate $q_{i,j}^{F,l}$ by

$$q_{i,j}^{F,l} \approx \sum_{s=0}^{r} f_{i,j}(t_{m-s}) \left(\int_0^{\Delta t} g_{i,j}^{(l)} \eta_{r,s}(\tau)\, \mathrm{d}\tau \right) = \sum_{s=0}^{r} f_{i,j}(t_{m-s}) \beta_{i,j}^{(r,s,l)}.$$

We again find that $\beta_{i,j}^{(r,s,l)}$ is independent of the time steps t_m when a uniform time step size Δt is used. We only present the value of $\beta_{i,j}^{(r,s,l)}$ for $r = 0, 1, 2$ below:

$$\begin{aligned}
&\beta_{i,j}^{(0,0,l)} = \phi_{i,j}^{(0,l)},\\
&\beta_{i,j}^{(1,0,l)} = \phi_{i,j}^{(0,l)} + \phi_{i,j}^{(1,l)}, \beta_{i,j}^{(1,0,r)} = -\phi_{i,j}^{(1,l)},\\
&\beta_{i,j}^{(2,0,l)} = \frac{1}{2}(2\phi_{i,j}^{(0,l)} + 3\phi_{i,j}^{(1,l)} + \phi_{i,j}^{(2,l)}), \beta_{i,j}^{(2,1,l)} = -(2\phi_{i,j}^{(1,l)} + \phi_{i,j}^{(2,l)}), \beta_{i,j}^{(2,2,l)} = \frac{1}{2}(\phi_{i,j}^{(1,l)} + \phi_{i,j}^{(2,l)}),
\end{aligned} \tag{2.22}$$

where $\{\phi_{i,j}^{(r,l)}\}_{r=0}^{2}$ are defined in (2.20). Denote $\boldsymbol{S}_{r,s,l}^{F} = (\beta_{i,j}^{(r,s,l)})_{(N_x-1)\times(N_y-1)}$, then we have the approximation of

$$\boldsymbol{Q}_l^{\,F} \approx \sum_{s=0}^{r} \boldsymbol{F}_{m-s} \odot \boldsymbol{S}_{r,s,l}^{F}, \tag{2.23}$$

which is $(r+1)$-th order accurate.

Finally we obtain a fast compact time integrator (FCTI) scheme for solving the n-th order evolution equation (1.1) with the Dirichlet boundary condition (2.1) in the rectangular domain as follows:

$$\begin{aligned}\boldsymbol{U}_{m+1}^{(l)} = \boldsymbol{P}_y \textcircled{y} \boldsymbol{P}_x \textcircled{x} \Bigg(\sum_{k=0}^{n-1} \boldsymbol{y}_k^{(l)}(\Delta t) \odot (\boldsymbol{P}_y^{-1} \textcircled{y} \boldsymbol{P}_x^{-1} \textcircled{x} \boldsymbol{U}_m^{(k)}) + \sum_{s=-1}^{r-1} (\boldsymbol{P}_y^{-1} \textcircled{y} \boldsymbol{P}_x^{-1} \textcircled{x} \boldsymbol{W}_{m-s}) \odot \boldsymbol{S}_{r,s,l}^{W} \\ + \sum_{s=0}^{r} \left(\boldsymbol{P}_y^{-1} \textcircled{y} \boldsymbol{P}_x^{-1} \textcircled{x} \boldsymbol{F}(\boldsymbol{U}_{m-s}, t_{m-s})\right) \odot \boldsymbol{S}_{r,s,l}^{F} \Bigg), \qquad 0 \le l \le n-1,\end{aligned} \tag{2.24}$$

which is fourth-order accurate in space and $(r+1)$-th order accurate in time. The overall complexity is $O(N^2 \log(N))$ per time step where $N = \max(N_x, N_y)$ based on the FFT implementation. Later on, we also call (2.24) as the FCTI-$(r+1)$ scheme in order to highlight its $(r+1)$-th order accuracy in time.

Remark 3. When the model equation (1.1) is a semilinear parabolic equation ($n = 1$), the above FCTI method is identical to the compact exponential time differencing method studied in [22].

2.3. The problem with the periodic boundary condition

If the periodic boundary condition as

$$\begin{cases} u(t, x_b, y) = u(t, x_e, y),\ \frac{\partial^k u}{\partial x^k}(t, x_b, y) = \frac{\partial^k u}{\partial x^k}(t, x_e, y) \text{ for } k = 1, 2, 3, & y \in [y_b, y_e], t \in [t_0, t_0 + T], \\ u(t, x, y_b) = u(t, x, y_e),\ \frac{\partial^k u}{\partial y^k}(t, x, y_b) = \frac{\partial^k u}{\partial y^k}(t, x, y_e) \text{ for } k = 1, 2, 3, & x \in [x_b, x_e], t \in [t_0, t_0 + T] \end{cases} \tag{2.25}$$

is imposed, then we denote the unknowns as

$$\boldsymbol{U} = (u_{i-1,j-1})_{N_x \times N_y} = \begin{pmatrix} u_{0,0} & u_{0,1} & \dots & u_{0,N_y-1} \\ u_{1,0} & u_{1,1} & \dots & u_{1,N_y-1} \\ \vdots & \vdots & \ddots & \vdots \\ u_{N_x-1,0} & u_{N_x-1,1} & \dots & u_{N_x-1,N_y-1} \end{pmatrix}_{N_x \times N_y}$$

and define $\boldsymbol{U}^{xx}$ and $\boldsymbol{U}^{yy}$ correspondingly. Let

$$\boldsymbol{A}_{P\times P}^{p} = \begin{pmatrix} 10 & 1 & \dots & 0 & 1 \\ 1 & 10 & \dots & 0 & 0 \\ & \ddots & \ddots & \ddots & \\ 0 & 0 & \dots & 10 & 1 \\ 1 & 0 & \dots & 1 & 10 \end{pmatrix}_{P\times P}, \quad \boldsymbol{B}_{P\times P}^{p} = \begin{pmatrix} -2 & 1 & \dots & 0 & 1 \\ 1 & -2 & \dots & 0 & 0 \\ & \ddots & \ddots & \ddots & \\ 0 & 0 & \dots & -2 & 1 \\ 1 & 0 & \dots & 1 & -2 \end{pmatrix}_{P\times P},$$

and

$$\boldsymbol{A}_x = \tfrac{1}{12}\boldsymbol{A}_{N_x\times N_x}^{p}, \quad \boldsymbol{A}_y = \tfrac{1}{12}\boldsymbol{A}_{N_y\times N_y}^{p}, \quad \boldsymbol{B}_x = \tfrac{D}{h_x^2}\boldsymbol{B}_{N_x\times N_x}^{p}, \quad \boldsymbol{B}_y = \tfrac{D}{h_y^2}\boldsymbol{B}_{N_y\times N_y}^{p}. \tag{2.26}$$

Then the compact finite difference approximations in the two-dimensional array form are given by

$$D\boldsymbol{A}_x \textcircled{x} \boldsymbol{U}^{xx} = \boldsymbol{B}_x \textcircled{x} \boldsymbol{U}, \qquad D\boldsymbol{A}_y \textcircled{y} \boldsymbol{U}^{yy} = \boldsymbol{B}_y \textcircled{y} \boldsymbol{U}. \tag{2.27}$$

Then we can obtain the semi-discretization in space of (1.1) with the periodic boundary condition (2.25) as follows:

$$\boldsymbol{L}_{n,t}(\boldsymbol{U}) - (\boldsymbol{A}_x^{-1} \textcircled{x} \boldsymbol{B}_x \textcircled{x} \boldsymbol{U} + \boldsymbol{A}_y^{-1} \textcircled{y} \boldsymbol{B}_y \textcircled{y} \boldsymbol{U}) = \boldsymbol{F}(\boldsymbol{U}, t). \tag{2.28}$$

Note that there is no boundary condition related term $\boldsymbol{W}$ appeared anymore. Then the corresponding FCTI-$(r+1)$ scheme can be obtained as:

$$\begin{aligned}\boldsymbol{U}_{m+1}^{(l)} = \boldsymbol{P}_y \textcircled{y} \boldsymbol{P}_x \textcircled{x} \Bigg(\sum_{k=0}^{n-1} \boldsymbol{y}_k^{(l)}(\Delta t) \odot (\boldsymbol{P}_y^{-1} \textcircled{y} \boldsymbol{P}_x^{-1} \textcircled{x} \boldsymbol{U}_m^{(k)}) + \sum_{s=0}^{r} \left(\boldsymbol{P}_y^{-1} \textcircled{y} \boldsymbol{P}_x^{-1} \textcircled{x} \boldsymbol{F}(\boldsymbol{U}_{m-s}, t_{m-s})\right) \odot \boldsymbol{S}_{r,s,l}^{F} \Bigg), \\ 0 \le l \le n-1,\end{aligned} \tag{2.29}$$

which is fourth-order accurate in space [25] and $(r+1)$-th order accurate in time. Under the periodic boundary condition, we have

$$d_i^{a,x} = 1 - \frac{1}{3}\sin^2(\tfrac{(i-1)\pi}{N_x}),\ \ d_i^{b,x} = -\frac{4D}{h_x^2}\sin^2(\tfrac{(i-1)\pi}{N_x}),\ \ d_j^{a,y} = 1 - \frac{1}{3}\sin^2(\tfrac{(j-1)\pi}{N_y}),\ \ d_j^{b,y} = -\frac{4D}{h_y^2}\sin^2(\tfrac{(j-1)\pi}{N_y}),$$

and the corresponding operations $\boldsymbol{P}_y \textcircled{y} \boldsymbol{P}_x \textcircled{x} \boldsymbol{V}$ and $\boldsymbol{P}_y^{-1} \textcircled{y} \boldsymbol{P}_x^{-1} \textcircled{x} \boldsymbol{V}$ are exactly the two-dimensional discrete Fourier transform (DFT) and the inverse DFT respectively, which again can be efficiently calculated by the FFT technique.

Remark 4. The proposed FCTI method also can be naturally generalized with some modifications to solve the general order evolution equation (1.1) with Neumann boundary condition by following the approach in [17] developed for handling semilinear parabolic problems.

3. Extension to three dimensions

The extension of the above FCTI method for solving the general order semilinear evolution problem (1.1) to three dimensions is straightforward. Let $\Omega = \{x_b < x < x_e, y_b < y < y_e, z_b < z < z_e\}$ and N_x, N_y, N_z be some positive integers. The mesh sizes in x-, y-, z-direction are $h_x = (x_e - x_b)/N_x$, $h_y = (y_e - y_b)/N_y$, and $h_z = (z_e - z_b)/N_z$, respectively. Let $u_{i,j,k}$, $u_{i,j,k}^{xx}$, $u_{i,j,k}^{yy}$, $u_{i,j,k}^{zz}$ denote the approximations of $u(x,y,z,t)$, $u_{xx}(x,y,z,t)$, $u_{yy}(x,y,z,t)$, $u_{zz}(x,y,z,t)$ at the grid point (x_i, y_j, z_k). Let us again use the Dirichlet boundary condition (2.1) as an illustration. Define $\boldsymbol{A}_x, \boldsymbol{A}_y, \boldsymbol{B}_x, \boldsymbol{B}_y$ as before and $\boldsymbol{A}_z = \frac{1}{12}\boldsymbol{A}^d_{(N_z-1)\times(N_z-1)}$, $\boldsymbol{B}_z = \frac{1}{12}\boldsymbol{B}^d_{(N_z-1)\times(N_z-1)}$.

Similar to the analysis for the two dimensional system, we can obtain a semi-discretization in space system of (1.1) in three dimensions as follows:

$$\boldsymbol{L}_{n,t}(\boldsymbol{U}) - (\boldsymbol{A}_x^{-1}\textcircled{x}\boldsymbol{B}_x\textcircled{x}\boldsymbol{U} + \boldsymbol{A}_y^{-1}\textcircled{y}\boldsymbol{B}_y\textcircled{y}\boldsymbol{U} + \boldsymbol{A}_z^{-1}\textcircled{z}\boldsymbol{B}_z\textcircled{z}\boldsymbol{U}) = \boldsymbol{F}(\boldsymbol{U},t) + \boldsymbol{W}(t), \tag{3.1}$$

where the above special operators are defined as

$$(\boldsymbol{A}\textcircled{x}\boldsymbol{U})_{i,j,k} = \sum_{l=1}^{N_x-1}(A)_{i,l}u_{l,j,k}, \quad (\boldsymbol{A}\textcircled{y}\boldsymbol{U})_{i,j,k} = \sum_{l=1}^{N_y-1}(A)_{j,l}u_{i,l,k}, \quad (\boldsymbol{A}\textcircled{z}\boldsymbol{U})_{i,j,k} = \sum_{l=1}^{N_z-1}(A)_{k,l}u_{i,j,l}$$

and

$$\mathcal{F}(\boldsymbol{U},t) = (f(u_{i,j,k}, x_i, y_j, z_k, t))_{(N_x-1)\times(N_y-1)\times(N_z-1)},$$
$$\boldsymbol{W} = \boldsymbol{A}_x^{-1}\textcircled{x}(\boldsymbol{U}_{x0} - \boldsymbol{U}_{x2}) + \boldsymbol{A}_y^{-1}\textcircled{y}(\boldsymbol{U}_{y0} - \boldsymbol{U}_{y2}) + \boldsymbol{A}_z^{-1}\textcircled{z}(\boldsymbol{U}_{z0} - \boldsymbol{U}_{z2})$$

with

$$\boldsymbol{U}_{x0} = \frac{D}{h_x^2}\left(\gamma_{i,j,k}^{x,0}\right)_{(N_x-1)\times(N_y-1)\times(N_z-1)}, \qquad \gamma_{i,j,k}^{x,0} = \begin{cases} u^b(x_0, y_j, z_k, t), & i = 1, \\ 0, & 1 < i < N_x - 1, \\ u^b(x_{N_x}, y_j, z_k, t), & i = N_x - 1, \end{cases}$$

$$\boldsymbol{U}_{y0} = \frac{D}{h_y^2}\left(\gamma_{i,j,k}^{y,0}\right)_{(N_x-1)\times(N_y-1)\times(N_z-1)}, \qquad \gamma_{i,j,k}^{y,0} = \begin{cases} u^b(x_i, y_0, z_k, t), & j = 1, \\ 0, & 1 < j < N_y - 1, \\ u^b(x_i, y_{N_y}, z_k, t), & j = N_y - 1, \end{cases}$$

$$\boldsymbol{U}_{z0} = \frac{D}{h_z^2}\left(\gamma_{i,j,k}^{z,0}\right)_{(N_x-1)\times(N_y-1)\times(N_z-1)}, \qquad \gamma_{i,j,k}^{z,0} = \begin{cases} u^b(x_i, y_j, z_0, t), & k = 1, \\ 0, & 1 < k < N_z - 1, \\ u^b(x_i, y_j, z_{N_z}, t), & k = N_z - 1, \end{cases}$$

$$\boldsymbol{U}_{x2} = \frac{D}{12}\left(\gamma_{i,j,k}^{x,2}\right)_{(N_x-1)\times(N_y-1)\times(N_z-1)}, \qquad \gamma_{i,j,k}^{x,2} = \begin{cases} u_{xx}(x_0, y_j, z_k, t), & i = 1, \\ 0, & 1 < i < N_x - 1, \\ u_{xx}(x_{N_x}, y_j, z_k, t), & i = N_x - 1, \end{cases}$$

$$\boldsymbol{U}_{y2} = \frac{D}{12}\left(\gamma_{i,j,k}^{y,2}\right)_{(N_x-1)\times(N_y-1)\times(N_z-1)}, \qquad \gamma_{i,j,k}^{y,2} = \begin{cases} u_{yy}(x_i, y_0, z_k, t), & j = 1, \\ 0, & 1 < j < N_y - 1, \\ u_{yy}(x_i, y_{N_y}, z_k, t), & j = N_y - 1, \end{cases}$$

$$\boldsymbol{U}_{z2} = \frac{D}{12}\left(\gamma_{i,j,k}^{z,2}\right)_{(N_x-1)\times(N_y-1)\times(N_z-1)}, \qquad \gamma_{i,j,k}^{z,2} = \begin{cases} u_{zz}(x_i, y_j, z_0, t), & k = 1, \\ 0, & 1 < k < N_z - 1, \\ u_{zz}(x_i, y_j, z_{N_z}, t), & k = N_z - 1. \end{cases}$$

The values of $\boldsymbol{U}_{x2}, \boldsymbol{U}_{y2}$ and $\boldsymbol{U}_{z2}$ again can be obtained by combining the equation (1.1) and the Dirichlet boundary condition (2.1). Following the same derivations in the case of two dimensions, we can obtain a FCTI-$(r+1)$ scheme of fourth-order accurate in space and $(r+1)$-th order accurate in time as follows:

$$\boldsymbol{U}_{m+1}^{(l)} = \boldsymbol{P}_z ⓩ \boldsymbol{P}_y ⓨ \boldsymbol{P}_x ⓧ \Bigg(\sum_{k=0}^{n-1} \boldsymbol{y}_k^{(l)} (\Delta t) \odot (\boldsymbol{P}_z^{-1} ⓩ \boldsymbol{P}_y^{-1} ⓨ \boldsymbol{P}_x^{-1} ⓧ \boldsymbol{U}_m^{(k)}) + \sum_{s=-1}^{r-1} (\boldsymbol{P}_z^{-1} ⓩ \boldsymbol{P}_y^{-1} ⓨ \boldsymbol{P}_x^{-1} ⓧ \boldsymbol{W}_{m-s}) \odot \boldsymbol{S}_{r,s,l}^{W}$$
$$+ \sum_{s=0}^{r} \big(\boldsymbol{P}_z^{-1} ⓩ \boldsymbol{P}_y^{-1} ⓨ \boldsymbol{P}_x^{-1} ⓧ \boldsymbol{F}(\boldsymbol{U}_{m-s}, t_{m-s}) \big) \odot \boldsymbol{S}_{r,s,l}^{F} \Bigg), \qquad 0 \le l \le n-1. \tag{3.2}$$

Note the operations $\boldsymbol{P}_z ⓩ \boldsymbol{P}_y ⓨ \boldsymbol{P}_x ⓧ \boldsymbol{V}$ and $\boldsymbol{P}_z^{-1} ⓩ \boldsymbol{P}_y^{-1} ⓨ \boldsymbol{P}_x^{-1} ⓧ \boldsymbol{V}$ are exactly the three-dimensional discrete sine transform (DST) and the inverse DST respectively, and again can be efficiently calculated by FFT. The overall complexity is $O(N^3 \log(N))$ per time step where $N = \max(N_x, N_y, N_z)$.

4. Linear stability analysis

For the proposed FCTI method, we perform its stability analysis for the following second order $(n = 2)$ linear evolution equation

$$u_{tt} + \rho u_t = \boldsymbol{L}u + \lambda u \tag{4.1}$$

with a homogeneous Dirichlet boundary condition, where $\rho \ge 0$ is a sufficiently small number and $\boldsymbol{L}u = -qu$. For its linear stability analysis for the case of $n = 1$, we refer to [22] for details. As is done in [11,12,22], we consider the cases where λ is complex-valued and λu represents the linearization of the nonlinear term, and q is a positive real number that corresponds to a Fourier mode of the self-adjoint elliptic operator Δ.

Let $\alpha = -\rho/2$, $\beta = \sqrt{q - \rho^2/4}$. Assume that $q - \rho^2/4 > 0$. Let us consider the FCTI-1 scheme $(r = 0)$ applied to (4.1), which leads to

$$\begin{aligned} u_{m+1} &= e^{\alpha \Delta t} \big(\cos(\beta \Delta t) - \alpha \varphi_0 \big) u_m + e^{\alpha \Delta t} \varphi_0 v_m + \varphi_1 \lambda u_m, \\ v_{m+1} &= -q e^{\alpha \Delta t} \varphi_0 u_m + e^{\alpha \Delta t} \big(\cos(\beta \Delta t) + \alpha \varphi_0 \big) v_m + e^{\alpha \Delta t} \varphi_0 \lambda u_m, \end{aligned} \tag{4.2}$$

where

$$\varphi_0 = \sin(\beta \Delta t)/\beta, \quad \varphi_1 = \frac{1 - e^{\alpha \Delta t} (\cos(\beta \Delta t) - \alpha \sin(\beta \Delta t)/\beta)}{q}$$

and $u_m \approx u(t_m)$, $v_m \approx u_t(t_m)$. The corresponding amplification matrix (i.e., $(u_{m+1}, v_{m+1})^T = \boldsymbol{A}(u_m, v_m)^T$) can be written as

$$\boldsymbol{A} = \begin{pmatrix} e^{\alpha \Delta t} \big(\cos(\beta \Delta t) - \alpha \varphi_0 \big) + \varphi_1 \lambda & e^{\alpha \Delta t} \varphi_0 \\ -q e^{\alpha \Delta t} \varphi_0 + e^{\alpha \Delta t} \varphi_0 \lambda & e^{\alpha \Delta t} \big(\cos(\beta \Delta t) + \alpha \varphi_0 \big) \end{pmatrix}. \tag{4.3}$$

Define

$$\begin{aligned} b_1 &:= \operatorname{trace}(\boldsymbol{A}/2) = e^{\alpha \Delta t} \cos(\beta \Delta t) + \varphi_1 \lambda / 2, \\ b_2 &:= \det(\boldsymbol{A}) = e^{2\alpha \Delta t} + \lambda e^{\alpha \Delta t} (\cos(\beta \Delta t) + \alpha \varphi_0 - e^{\alpha \Delta t})/q. \end{aligned}$$

Then, the eigenvalues are

$$\mu_{1,2}(\boldsymbol{A}) = b_1 \pm (b_1^2 - b_2)^{1/2}.$$

To ensure the stability of the scheme (4.2), we need to block magnification of $\boldsymbol{A}^m$ as m increases. Based on the spectral decomposition of $\boldsymbol{A}$, we shall discuss the eigenvalues of $\boldsymbol{A}$ in the following.

Case 1: If $\mu_1(\boldsymbol{A}) = \mu_2(\boldsymbol{A}) = \mu(\boldsymbol{A})$ and $\boldsymbol{A} = \mu(\boldsymbol{A})\boldsymbol{I}$. Then, according the expression of $\boldsymbol{A}$, we easily know $\varphi_0 = \varphi_1 = 0$ must hold and $\boldsymbol{A}$ is an identity matrix in this case. Hence, the FCTI-1 scheme is stable.

Case 2: If $\mu_1(\boldsymbol{A}) = \mu_2(\boldsymbol{A}) = \mu(\boldsymbol{A})$ and

$$\boldsymbol{A} = \boldsymbol{Q}\, \boldsymbol{J}\, \boldsymbol{Q}^{-1}, \qquad \boldsymbol{J} = \begin{pmatrix} \mu(\boldsymbol{A}) & 1 \\ 0 & \mu(\boldsymbol{A}) \end{pmatrix},$$

then it holds

$$\boldsymbol{A}^m = \boldsymbol{Q}\, \boldsymbol{J}^m\, \boldsymbol{Q}^{-1}, \qquad \boldsymbol{J}^m = \begin{pmatrix} \mu(\boldsymbol{A})^m & m\mu(\boldsymbol{A})^{m-1} \\ 0 & \mu(\boldsymbol{A})^m \end{pmatrix}.$$

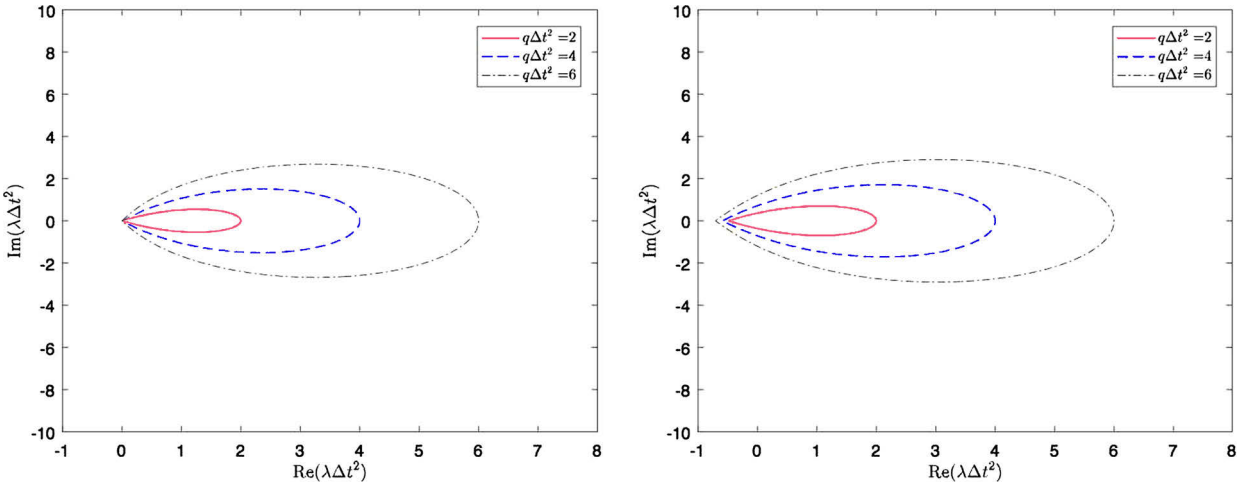


Fig. 1. Stability regions of the first-order time integrator scheme (FCTI-1) in the complex plane of $\lambda\Delta t^2$ for $q\Delta t^2 = 2, 4, 6$. Left: $\rho\Delta t = 0$; right: $\rho\Delta t = 0.16$.

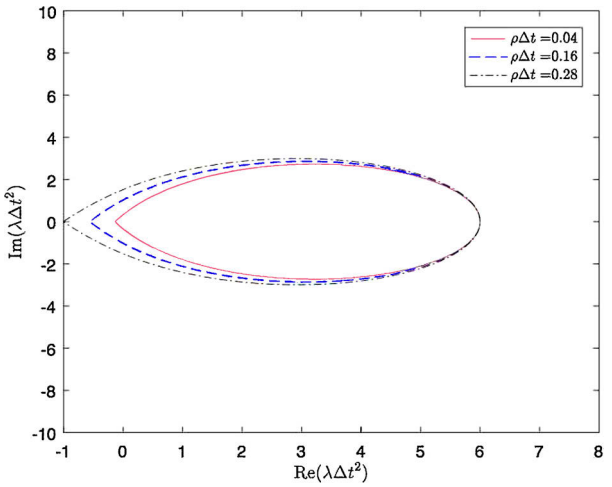


Fig. 2. Stability regions of the first-order time integrator scheme (FCTI-1) in the complex plane of $\lambda\Delta t^2$ with $q\Delta t^2 = 6$ for $\rho\Delta t = 0.04, 0.16, 0.28$.

Noted that the scheme is stable if $|\mu(\boldsymbol{A})| < 1$ and unstable if $|\mu(\boldsymbol{A})| > 1$. If $|\mu(\boldsymbol{A})| = 1$, we find $|m\mu(\boldsymbol{A})^{m-1}| \to \infty$ as $m \to \infty$, thus the FCTI-1 scheme is also unstable in this case.

Case 3: If $\mu_1(\boldsymbol{A}) \neq \mu_2(\boldsymbol{A})$ and

$$\boldsymbol{A}^m = \boldsymbol{Q}\,\boldsymbol{J}^m\,\boldsymbol{Q}^{-1}, \qquad \boldsymbol{J} = \begin{pmatrix} \mu_1(\boldsymbol{A}) & 1 \\ 0 & \mu_2(\boldsymbol{A}) \end{pmatrix},$$

it is easy to see that the scheme is stable if and only if

$$\rho(\boldsymbol{A}) = \max\{|\mu_1|, |\mu_2|\} \leq 1. \tag{4.4}$$

We also can obtain the boundary locus curve equation of the stability region as

$$\begin{cases} \mathrm{Re}(\mu_1)^2 + \mathrm{Im}(\mu_1)^2 = 1, \\ \mathrm{Re}(\mu_2)^2 + \mathrm{Im}(\mu_2)^2 = 1. \end{cases} \tag{4.5}$$

The stability regions of the FCTI-1 scheme with $\rho\Delta t = 0$ for different values of $q\Delta t^2$ are given in Fig. 1-(left). It can be seen that the stability region gets larger obviously when the value of $q\Delta t^2$ gets bigger. Fig. 1-(right) shows the case with $\rho\Delta t = 0.16$; it is observed that $\mathrm{Re}(\lambda\Delta t^2)$ can be smaller than zero. The stability regions with $q\Delta t^2 = 6$ for $\rho\Delta t = 0.04, 0.16, 0.28$ are plotted in Fig. 2. We observe that the stability region grows larger with the increasing of $\rho\Delta t$.

Table 1
The L_∞ errors of the approximate solution by the FCTI-1 scheme in Example 1 with different λ.

Time	$\rho=0$			$\rho=1$		
	$\lambda=-10$	$\lambda=10$	$\lambda=30$	$\lambda=-10$	$\lambda=10$	$\lambda=30$
$t=0.1$	1.5341e-02	1.0847e-02	6.2936e-03	1.4815e-02	1.0465e-02	6.0569e-03
$t=12.5$	3.6510e-02	1.7913e-02	1.4124e-02	2.7157e-02	2.2159e-02	1.4141e-02
$t=25.0$	3.4587e-01	2.1519e-02	1.3886e+04	2.7103e-02	2.2047e-02	9.4677e-01
$t=50.0$	3.6521e+00	2.1538e-02	6.2126e+24	2.6440e-02	2.1520e-02	4.0863e+16

5. Numerical experiments

In this section, we perform various numerical experiments to illustrate the performance of the proposed FCTI method. All computations are done using Matlab R2016b software on an Intel(R) i7-8750H, 2.20GHz CPU Laptop with 8 GB of memory. All reported CPU times are measured in seconds.

5.1. Stability tests

In this subsection, we numerically investigate the stability of the FCTI method following the linear stability analysis presented in Section 4.

Example 1. Consider the following second-order linear hyperbolic evolution equation with the Dirichlet boundary condition in one dimension:

$$\frac{\partial^2 u}{\partial t^2}+\rho\frac{\partial u}{\partial t}=Du_{xx}+\lambda u+f(x,t),\qquad 0<x<2\pi,\ t>0. \tag{5.1}$$

The initial and boundary conditions and the source function $f(x,t)$ are determined accordingly from the given exact solution

$$u(x,t)=\sin(x)\sin(t).$$

After spatial discretization by fourth-order compact difference scheme, we can obtain a semi-discretization of (5.1) as

$$\boldsymbol{U}''(t)+\rho\boldsymbol{U}'(t)-\boldsymbol{A}_x^{-1}\boldsymbol{B}_x\boldsymbol{U}=\lambda\boldsymbol{U}+\boldsymbol{F}(t), \tag{5.2}$$

where $\boldsymbol{U}\in\mathbb{R}^{(N_x-1)\times 1}$. Multiplying $\boldsymbol{P}_x^{-1}$ from the left hand side yields the ODE system

$$\boldsymbol{V}''+\rho\boldsymbol{V}'=-\boldsymbol{Q}\odot\boldsymbol{V}+\lambda\boldsymbol{V}+\widehat{\boldsymbol{F}}(t), \tag{5.3}$$

where $\boldsymbol{V}=\boldsymbol{P}_x^{-1}\boldsymbol{U}$, $\widehat{\boldsymbol{F}}(t)=\boldsymbol{P}_x^{-1}\boldsymbol{F}(t)$ and $\boldsymbol{Q}=(q_i)_{(N_x-1)\times 1}$ with

$$q_i=\frac{48D}{h_x^2}\frac{\sin^2(i\pi/(2N_x))}{12-4\sin^2(i\pi/(2N_x))}=\frac{12DN_x^2}{\pi^2}\frac{\sin^2(i\pi/(2N_x))}{12-4\sin^2(i\pi/(2N_x))},\qquad i=1,2,\ldots,N_x-1.$$

We fix $D=100$, $N_x=32$ and $\Delta t=0.05$. Assume that the parameter λ is a real number. We first study the case of $\rho=0$. According to the previous linear stability analysis and the stability criterion (9.1.52)-(9.1.53) in [18, p. 498], we can see that the stability interval for the i-th equation in (5.3) is $\lambda\in[0,q_i]$, and the stability interval for the system (5.3) is then obtained as $\lambda\in[0,q]$ with $q=\min_{1\le i\le N_x-1}q_i$. It easy to verify $q=25$ in this example. Table 1 reports the L_∞ errors of the approximate solutions obtained by using the FCTI-1 scheme with different λ when $\rho=0$. It is observed that if $\lambda=10$ (resp. $\lambda=-10,30$), the FCTI-1 scheme is stable (resp. unstable), the L_∞ errors with respect to t are convergent (resp. divergent). This demonstrates the validity of our stability analysis. For comparison, we also check the case of $\rho=1$, in which there exists a damping term in the underlying equation. Hence, it is expected that the stability interval should be larger than that of the $\rho=0$ case from the physical point of view. As a matter of fact, using the similar arguments given above, one can find the stability interval now is $\lambda\in[-40.55,25]$, which does contains $[0,25]$. Results on $\rho=1$ reported in Table 1 also confirm this analysis.

5.2. Convergence and efficiency tests

In this subsection we first investigate the convergence and efficiency of the proposed FCTI method for second and third order evolution equations in two dimensions with Dirichlet or periodic boundary conditions.

Table 2
Numerical errors, convergence rates and the CPU times at $T = 1$ of Example 2 with Dirichlet or periodic boundary conditions by using the FCTI schemes.

$(N_x \times N_y) \times N_t$	Dirichlet boundary condition					Periodic boundary condition				
	L_2 error	CR	L_∞ error	CR	CPU	L_2 error	CR	L_∞ error	CR	CPU
Accuracy test of space discretization										
$(8^2) \times 1024$	3.0133e-03	–	1.2216e-03	–	0.396	4.6582e-03	.	1.3620e-03	–	0.102
$(16^2) \times 1024$	1.9105e-04	3.98	7.9303e-05	3.95	0.463	2.6501e-04	4.14	8.4979e-05	4.00	0.195
$(32^2) \times 1024$	1.1963e-05	4.00	5.0997e-06	3.96	0.910	1.5827e-05	4.07	5.4762e-06	3.96	0.432
$(64^2) \times 1024$	7.4880e-07	4.00	3.1993e-07	3.99	2.938	9.6875e-07	4.03	3.4283e-07	4.00	1.146
$(128^2) \times 1024$	4.7664e-08	3.97	2.0958e-08	3.93	7.828	6.1372e-08	3.98	2.2385e-08	3.94	3.688
Accuracy test of time discretization: FCTI-1 ($r = 0$)										
$(512^2) \times 4$	1.8393e-01	–	2.6377e-01	–	1.001	1.3695e-01	–	5.7609e-02	–	0.339
$(512^2) \times 8$	8.8637e-02	1.05	1.2207e-01	1.11	1.900	6.8759e-02	0.99	3.0256e-02	0.93	0.679
$(512^2) \times 16$	4.2203e-02	1.07	4.9362e-02	1.31	3.828	3.4608e-02	0.99	1.5599e-02	0.96	1.375
$(512^2) \times 32$	1.9399e-02	1.12	1.4647e-02	1.75	7.612	1.7384e-02	0.99	7.9333e-03	0.98	2.794
$(512^2) \times 64$	9.6058e-03	1.01	6.5981e-03	1.15	15.394	8.7151e-03	1.00	4.0022e-03	0.99	5.480
$(512^2) \times 128$	4.7817e-03	1.01	3.2903e-03	1.00	30.566	4.3637e-03	1.00	2.0103e-03	0.99	10.935
Accuracy test of time discretization: FCTI-2 ($r = 1$)										
$(512^2) \times 4$	2.6259e-02	–	1.9592e-02	–	1.007	3.4760e-02	–	2.1509e-02	–	0.406
$(512^2) \times 8$	7.5548e-03	1.80	5.6038e-03	1.81	1.983	1.0156e-02	1.78	6.4097e-03	1.75	0.819
$(512^2) \times 16$	1.9984e-03	1.92	1.4848e-03	1.92	3.965	2.7163e-03	1.90	1.7272e-03	1.89	1.662
$(512^2) \times 32$	5.1189e-04	1.96	3.8288e-04	1.96	7.992	7.0012e-04	1.96	4.4657e-04	1.95	3.326
$(512^2) \times 64$	1.2941e-04	1.98	9.7135e-05	1.98	15.889	1.7757e-04	1.98	1.1343e-04	1.98	6.645
$(512^2) \times 128$	3.2524e-05	1.99	2.4470e-05	1.99	31.896	4.4704e-05	1.99	2.8576e-05	1.99	13.311
Accuracy test of time discretization: FCTI-3 ($r = 2$)										
$(512^2) \times 4$	1.6012e-02	–	1.2685e-02	–	1.073	2.2089e-02	–	1.3846e-02	–	0.468
$(512^2) \times 8$	2.1945e-03	2.87	2.7079e-03	2.23	2.295	3.2384e-03	2.77	2.0135e-03	2.78	0.996
$(512^2) \times 16$	3.0707e-04	2.84	6.3837e-04	2.08	4.247	4.2947e-04	2.91	2.6222e-04	2.94	2.029
$(512^2) \times 32$	4.4023e-05	2.80	1.0365e-04	2.62	8.543	5.5211e-05	2.96	3.3287e-05	2.98	4.109
$(512^2) \times 64$	5.9269e-06	2.89	1.3929e-05	2.90	17.031	6.9982e-06	2.98	4.1886e-06	2.99	8.307
$(512^2) \times 128$	7.5923e-07	2.96	1.7109e-06	3.03	34.335	8.8093e-07	2.99	5.2525e-07	3.00	16.566

Example 2. Consider the following second-order nonlinear evolution problem

$$\begin{cases} \dfrac{\partial^2 u}{\partial t^2} + \dfrac{1}{4}\dfrac{\partial u}{\partial t} = \Delta u + u - u^2 + f(x, y, t), & (x, y) \in \Omega,\ t \in (0, T), \\ u(0, x, y) = \sin(x)\sin(y), & (x, y) \in \Omega, \\ u_t(0, x, y) = 0, & (x, y) \in \Omega, \end{cases} \tag{5.4}$$

where $\Omega = (-1, 2\pi - 1)^2$. The exact solution is chosen to be

$$u(x, y, t) = \cos(t)\sin(x)\sin(y)$$

and the source term $f(x, y, t)$ is determined correspondingly.

Numerical results on L_2 and L_∞ errors by using the FCTI schemes at the final time $T = 1$ and corresponding convergence rates and CPU times are reported in Table 2. For all spatial accuracy tests, we use the FCTI-3 scheme. We can clearly see the fourth order convergence in space from the results. For the temporal accuracy tests, we use a fixed spatial grid with $N_x = N_y = 512$. We observe as expected the first, second, and third order convergences in time for FCTI-1, FCTI-2 and FCTI-3, respectively.

Example 3. Consider the following third-order linear evolution problem

$$\begin{cases} \dfrac{\partial^3 u}{\partial t^3} - \dfrac{\partial u}{\partial t} = \Delta u - 2u - 2\mathrm{e}^{-2t}\sin(x)\sin(y), & (x, y) \in \Omega,\ t \in (0, T], \\ u(0, x, y) = \sin(x)\sin(y), & (x, y) \in \Omega, \\ u_t(0, x, y) = -2\sin(x)\sin(y), & (x, y) \in \Omega, \\ u_{tt}(0, x, y) = 4\sin(x)\sin(y), & (x, y) \in \Omega, \end{cases} \tag{5.5}$$

where $\Omega = (-1, 2\pi - 1)^2$. The exact solution is given by

$$u(x, y, t) = \mathrm{e}^{-2t}\sin(x)\sin(y).$$

Table 3
Numerical errors, convergence rates and the CPU times at $T = 0.25$ of Example 3 with Dirichlet or periodic boundary conditions by using the FCTI schemes.

$(N_x \times N_y) \times N_t$	Dirichlet boundary condition					Periodic boundary condition				
	L_2 error	CR	L_∞ error	CR	CPU	L_2 error	CR	L_∞ error	CR	CPU
Accuracy test of space discretization										
$(8^2) \times 1024$	2.0026e-05	–	7.7910e-06	–	0.587	2.7776e-05	–	7.1711e-06	–	0.133
$(16^2) \times 1024$	1.3439e-06	3.90	4.8472e-07	4.01	0.675	1.5783e-06	4.14	4.4706e-07	4.00	0.262
$(32^2) \times 1024$	8.6898e-08	3.95	2.8727e-08	4.08	1.583	9.4209e-08	4.07	2.8707e-08	3.96	0.586
$(64^2) \times 1024$	5.5261e-09	3.97	1.7919e-09	4.00	4.890	5.7559e-09	4.03	1.7919e-09	4.00	1.690
$(128^2) \times 1024$	3.5597e-10	3.96	1.4269e-10	3.65	13.208	3.5506e-10	4.02	1.1183e-10	4.00	5.833
Accuracy test of time discretization: FCTI-1 ($r = 0$)										
$(512^2) \times 4$	1.1486e-01	–	4.4166e-01	–	1.654	9.9518e-04	–	3.1589e-04	–	0.599
$(512^2) \times 8$	5.4659e-02	1.07	2.2583e-01	0.97	3.253	4.7760e-04	1.06	1.5160e-04	1.06	1.172
$(512^2) \times 16$	2.6761e-02	1.03	1.1511e-01	0.97	6.535	2.3346e-04	1.03	7.4104e-05	1.03	2.356
$(512^2) \times 32$	1.3249e-02	1.01	5.8167e-02	0.98	12.980	1.1535e-04	1.02	3.6615e-05	1.02	4.768
$(512^2) \times 64$	6.5925e-03	1.01	2.9241e-02	0.99	26.575	5.7327e-05	1.01	1.8197e-05	1.01	9.469
$(512^2) \times 128$	3.2884e-03	1.00	1.4661e-02	1.00	52.099	2.8575e-05	1.00	9.0704e-06	1.00	18.780
Accuracy test of time discretization: FCTI-2 ($r = 1$)										
$(512^2) \times 4$	2.2274e-03	–	1.0400e-02	–	1.703	1.8767e-05	–	5.9569e-06	–	0.685
$(512^2) \times 8$	5.4947e-04	2.02	2.4889e-03	2.06	3.398	4.7337e-06	1.99	1.5026e-06	1.99	1.377
$(512^2) \times 16$	1.3686e-04	2.01	6.1496e-04	2.02	6.783	1.1861e-06	2.00	3.7648e-07	2.00	2.733
$(512^2) \times 32$	3.4183e-05	2.00	1.5328e-04	2.00	13.629	2.9668e-07	2.00	9.4173e-08	2.00	5.475
$(512^2) \times 64$	8.5438e-06	2.00	3.8292e-05	2.00	27.448	7.4180e-08	2.00	2.3546e-08	2.00	10.979
$(512^2) \times 128$	2.1358e-06	2.00	9.5712e-06	2.00	54.519	1.8545e-08	2.00	5.8864e-09	2.00	22.014
Accuracy test of time discretization: FCTI-3 ($r = 2$)										
$(512^2) \times 4$	2.0104e-03	–	8.0884e-03	–	1.771	1.1177e-05	–	3.5478e-06	–	0.765
$(512^2) \times 8$	2.8940e-04	2.80	1.1081e-03	2.87	3.573	1.6195e-06	2.79	5.1408e-07	2.79	1.574
$(512^2) \times 16$	3.7976e-05	2.93	1.6598e-04	2.74	7.193	2.1732e-07	2.90	6.8982e-08	2.90	3.216
$(512^2) \times 32$	4.8647e-06	2.96	2.2346e-05	2.89	14.411	2.8125e-08	2.95	8.9274e-09	2.95	6.467
$(512^2) \times 64$	6.1637e-07	2.98	2.8864e-06	2.95	29.103	3.5754e-09	2.98	1.1349e-09	2.98	13.146
$(512^2) \times 128$	7.7608e-08	2.99	3.6636e-07	2.98	58.760	4.4955e-10	2.99	1.4270e-10	2.99	26.521

Numerical results at the final time $T = 0.25$ for Example 3 are reported in Table 3. As expected, we again observe the four-order accuracy of space discretization, and the first, second and third-order accuracy of time discretization for FCTI-1, FCTI-2 and FCTI-3, respectively. Moreover, from Tables 2 and 3 we can observe that all CPU times are quite small due to the use of FFT.

5.3. Comparison with the Fourier spectral IFRK method

The proposed FCTI method with the Fourier spectral IFRK method proposed in [1] will be compared in this subsection, while the FCTI can be used for solving problems with different types of boundary conditions, the Fourier spectral IFRK method usually only can handle the periodic boundary condition. In the following we will investigate their performance in accuracy and numerical stability under the periodic boundary condition, in particular, we take the FCTI-3 scheme with the Fourier spectral IFRK4 scheme for comparison.

Example 4. Consider the following second order linear hyperbolic equation in two dimensions

$$\frac{\partial^2 u}{\partial t^2} = \Delta u + f(x, y, t), \qquad (x, y) \in \Omega = (0, 2\pi)^2,\ t \in (0, 1], \tag{5.6}$$

imposed with the periodic boundary condition.

Note that the Fourier spectral IFRK method formulates the second order evolution equation as a system of first order evolution equations. It is well known [4,14] that the performance of Fourier spectral method depends strongly the regularity and the period conditions of the exact solution of the underlying problem. Hence, we discuss two cases with the exact solutions given respectively by Case (I): $u(x, y, t) = \sin(x)\sin(y)\sin(t)$ and Case (II): $u(x, y, t) = x^2(x - 2\pi)^2 y^2(y - 2\pi)^2 \sin(x)\sin(y)\sin(t)$. The right side function $f(x, y, t)$ is determined accordingly for each case. We report in Table 4 numerical results on L_2, L_∞ errors, and corresponding convergence rates and CPU times for both cases, computed by using the Fourier spectral IFRK4 scheme and the FCTI-3 scheme with $N_t = 2048$ and different values of $N_x \times N_y$. It is observed that the FCTI-3 scheme obtain fourth-order accuracy in space for both cases. The Fourier spectral IFRK4 scheme has the spectral accuracy for Case (I) and but only fourth-order accuracy for Case (II). Such behaviors can be explained as follows. For Case (I), the exact solution u is sufficiently smooth, i.e., $u \in C_p^\infty(\Omega)$, which indicates the periodic extension of u is a C^∞-smooth function in the plane $\mathbb{R}^2$, and thus the Fourier spectral method gives spectral accuracy in space [4,14]. However, such smoothness condition is sometimes too strong for practical applications. For Case (II), the exact solution u only has

Table 4
Numerical errors, convergence rates and CPU times of Example 4 by using the Fourier spectral IFRK4 scheme and the FCTI-3 scheme with $N_t = 2048$, respectively.

Case (I)										
$N_x \times N_y$	Fourier spectral IFRK4					FCTI-3				
	L_2 Error	CR	L_∞ error	CR	CPU	L_2 error	CR	L_∞ Error	CR	CPU
4^2	2.4425e-15	–	7.6734e-15	–	0.137	2.4616e-02	–	7.8355e-03	–	0.134
8^2	2.4425e-15	0.00	7.6336e-15	0.01	0.178	1.4582e-03	4.08	4.6415e-04	4.08	0.156
16^2	2.4425e-15	0.00	7.5431e-15	0.02	0.351	8.9581e-05	4.02	2.8514e-05	4.02	0.283
32^2	2.4425e-15	0.00	7.5903e-15	−0.01	0.783	5.5735e-06	4.01	1.7741e-06	4.01	0.603
64^2	2.4425e-15	0.00	7.6260e-15	−0.01	1.974	3.4794e-07	4.00	1.1075e-07	4.00	1.589
128^2	2.4425e-15	0.00	7.6669e-15	−0.01	5.869	2.1735e-08	4.00	6.9185e-09	4.00	4.690
256^2	2.8866e-15	−0.24	7.7256e-15	−0.01	44.652	1.3534e-09	4.01	4.3976e-10	4.01	16.564
512^2	2.7756e-15	0.01	7.7903e-15	−0.01	286.532	7.9930e-11	4.09	2.5211e-11	4.09	133.133
Case (II)										
$N_x \times N_y$	Fourier spectral IFRK4					FCTI-3				
	L_2 error	CR	L_∞ error	CR	CPU	L_2 error	CR	L_∞ error	CR	CPU
4^2	1.5468e+02	–	4.8594e+02	–	0.155	5.6122e+02	–	1.7864e+02	–	0.142
8^2	1.2099e+01	3.68	3.8519e+01	3.66	0.271	6.8185e+01	3.04	2.6498e+01	2.75	0.198
16^2	1.4247e+00	3.09	2.8761e+00	3.74	0.571	4.2155e+00	4.02	1.7113e+00	3.95	0.342
32^2	1.0561e-01	3.75	1.8439e-01	3.96	1.308	2.6074e-01	4.02	1.1149e-01	3.94	0.761
64^2	7.2262e-03	3.87	1.1732e-02	3.97	3.733	1.6246e-02	4.00	6.9456e-03	4.00	2.063
128^2	4.7244e-04	3.94	7.4017e-04	3.99	12.397	1.0146e-03	4.00	4.3529e-04	4.00	6.451
256^2	3.0217e-05	3.97	4.6483e-05	3.99	70.221	6.3365e-05	4.01	2.7185e-05	4.00	28.037
512^2	1.9101e-06	3.98	2.9122e-06	4.00	390.054	3.9269e-06	4.10	1.6831e-06	4.01	158.719

Table 5
Numerical errors, convergence rates and CPU times for Case (II) in Example 4 by using the Fourier spectral IFRK4 scheme and the FCTI-3 scheme, respectively.

$(N_x \times N_y) \times N_t$	Fourier spectral IFRK4			FCTI-3		
	L_2 Error	L_∞ error	CPU	L_2 error	L_∞ error	CPU
$(128^2) \times 4$	6.3580e+07	9.5824e+07	0.029	5.1367e+00	2.4402e+00	0.013
$(128^2) \times 8$	7.7659e+10	1.7019e+11	0.057	6.3357e-01	3.0101e-01	0.025
$(128^2) \times 16$	1.1039e+09	2.2767e+09	0.121	7.7793e-02	3.6989e-02	0.050
$(128^2) \times 32$	4.6163e-04	8.7341e-04	0.202	8.8332e-03	4.2268e-03	0.099
$(128^2) \times 64$	4.7155e-04	7.3569e-04	0.392	8.6708e-04	3.6296e-04	0.197
$(128^2) \times 128$	4.7238e-04	7.3981e-04	0.779	3.6971e-04	1.4936e-04	0.399
$(256^2) \times 4$	3.6222e+11	3.7293e+11	0.148	5.1377e+00	2.4406e+00	0.040
$(256^2) \times 8$	8.6325e+18	1.1010e+19	0.284	6.3449e-01	3.0142e-01	0.082
$(256^2) \times 16$	5.4973e+28	1.2390e+29	0.558	7.8712e-02	3.7395e-02	0.160
$(256^2) \times 32$	5.2208e+33	1.2253e+34	1.107	9.7490e-03	4.6334e-03	0.315
$(256^2) \times 64$	9.0234e+00	2.0771e+01	2.214	1.1627e-03	5.5435e-04	0.645
$(256^2) \times 128$	3.0186e-05	4.6160e-05	4.578	9.3136e-05	4.5506e-05	1.270

finite regularity, i.e. $u \in H_p^4(\Omega)$. In this case, the Fourier spectral IFRK4 scheme only gives fourth-order accuracy in space, the same order as the FCTI-3 scheme, and in fact the produced L^∞ errors are even larger than those by the FCTI-3 scheme as shown in the table. Hence, if the exact solution only has finite regularity, the two methods could enjoy the same approximation accuracy in space. Compared with the FCTI-3 scheme, the Fourier spectral IFRK4 scheme also costs more CPU times (more than twice on relatively large spatial meshes) since its computational complexity per time step is larger.

Furthermore, numerical results for Case (II) computed by the two methods with relative large time step sizes are presented in Table 5, in which we set the spatial grid size as $N_x = 128$ and 256, respectively. It is easy to see that the FCTI-3 scheme is more stable than the Fourier spectral IFRK4 scheme, for instance, the latter method is numerically unstable for $N_t \leq 16$ when $N_x = 128$ and for $N_t \leq 64$ when $N_x = 256$, while the FCTI-3 scheme always performs well.

5.4. Some application problems

5.4.1. The nonlinear sine-Gordon equation

The sine-Gordon equation takes the following form [8]:

$$\frac{\partial^2 u}{\partial t^2} + \rho \frac{\partial u}{\partial t} = D\Delta u - \sin(u), \qquad \boldsymbol{x} \in \Omega,\ t > 0, \tag{5.7}$$

where the parameter $\rho \geq 0$ is the dissipative coefficient. This equation appears in differential geometry, relativistic field theory, and a number of other physical applications such as the propagation of fluxons in Josephson junctions and dislocations in crystals. As a soliton wave equation, the sine-Gordon equation with the homogeneous Dirichlet boundary or the periodic boundary condition has the property of conservation of energy for the undamped case (i.e., $\rho = 0$), in which the energy is defined by [10,13]:

$$E(t) = \frac{1}{2}\int_{\Omega} [|\nabla u|^2 + u_t^2 + 2(1 - \cos u)]\, \mathrm{d}\boldsymbol{x}.$$

In this subsection, we will apply the FCTI method to simulate the sine-Gordon equation in two and three dimensions.

Example 5. Consider the behavior of a circular ring quasi-soliton [30,33,35] in two dimensions, arising from the sine-Gordon equation (5.7) with $D = 1$. The initial condition is given by

$$u(x, y, 0) = 4\arctan(\exp(3 - \sqrt{x^2 + y^2})), \quad u_t(x, y, 0) = 0,$$

and the boundary condition is set to be periodic.

In order to study the evolution of the ring solitons without damping (i.e., $\rho = 0$), numerical simulation of $\sin(u/2)$ at different times, by using the FCTI-3 scheme with $N_x = N_y = 256$ and $\Delta t = 0.005$ over the domain $\Omega = (-14, 14)^2$, were performed and the results are presented in Fig. 3. From the initial stage ($t = 0$), a ring soliton is shrinking. Then oscillations and radiations begin to form and continue until the time $t = 8$ as the time goes on. At $t = 11.5$, the ring soliton is nearly formed and then appears to be in its shrinking process again at $t = 12.6$. We also find that the center of the circle ring soliton does not move during the above process. The corresponding contour profiles can be found in Fig. 4. All simulated results coincide very well with those in the literature. In Fig. 5-(left), we plot its evolution of the energy $E(t)$ for $t \in [0, 60]$ over the domain $\Omega = (-7, 7)^2$, which demonstrates the proposed scheme conserves the energy quite well (with relative oscillations up to 0.0908%) in the undamped case even for large time. In addition, we also tested some damped cases and Fig. 5-(right) depicts the comparison of the energy evolutions with different damping coefficients $\rho = 0, 0.01, 0.03$ during the time interval $[0, 6]$. It is observed that for $\rho = 0.01$ and 0.03, the energy decreases as t increases and the larger ρ is, the faster $E(t)$ decreases.

Example 6. Consider the three-dimensional modified sine-Gordon equation with extra source [7]

$$\frac{\partial^2 u(x, y, z, t)}{\partial t^2} = \Delta u(x, y, z, t) - \sin(u(x, y, z, t)) + f(x, y, z, t), \qquad (x, y, z) \in \Omega,\ t \in (0, T],$$

where $\Omega = (0, 1)^3$ and $T = 1$. The exact solution is chosen to be an expending sphere defined by

$$u(x, y, z, t) = t^2 \exp\left(-\frac{(x - r)^2}{\beta} - \frac{(y - r)^2}{\beta} - \frac{(z - r)^2}{\beta}\right),$$

with two constant parameters $\beta > 0$ and $r \geq 0$. The source function f and the initial and Dirichlet boundary condition are determined accordingly from the exact solution.

It is easy to see that the isosurface of u is initially a sphere, and as the time goes on, the radius of the sphere become larger and gradually touch the domain boundary. To test the sphere expanding process, we set $\beta = 10$, $r = 0.5$ and solve the problem using the FCTI-3 scheme with $N_x = N_y = N_z = 128$ and $\Delta t = 1/256$. The simulated isosurfaces of $u \equiv 0.05$ at times $t = 0.25, 0.5$ and 1 respectively are illustrated in Fig. 6, which is in accordance with the exact solution.

5.4.2. A third-order nonlinear evolution equation

As shown in [34,38,39], the following differential equation can be used to describe the propagation of weakly nonlinear waves in relaxing media.

$$\frac{\partial}{\partial t}\left(\frac{\partial^2 u}{\partial t^2} - \Delta u\right) + \frac{\partial^2 u}{\partial t^2} - D\Delta u = \beta\Delta(u^2), \qquad \boldsymbol{x} \in \Omega,\ t > 0, \tag{5.8}$$

where $D \in (0, 1)$ and β are two dimensionless constants. As described in Appendix A the FCTI method can be slightly modified to solve (5.8) efficiently.

Example 7. Consider the numerical solution of problem (5.8) defined in $\Omega = (0, 2\pi)^2$ with the periodic boundary condition imposed. The initial conditions are given by

$$u(x, y, 0) = \sin(x)\sin(y), \quad u_t(x, y, 0) = 0, \quad u_{tt}(x, y, 0) = -\sin(x)\sin(y).$$

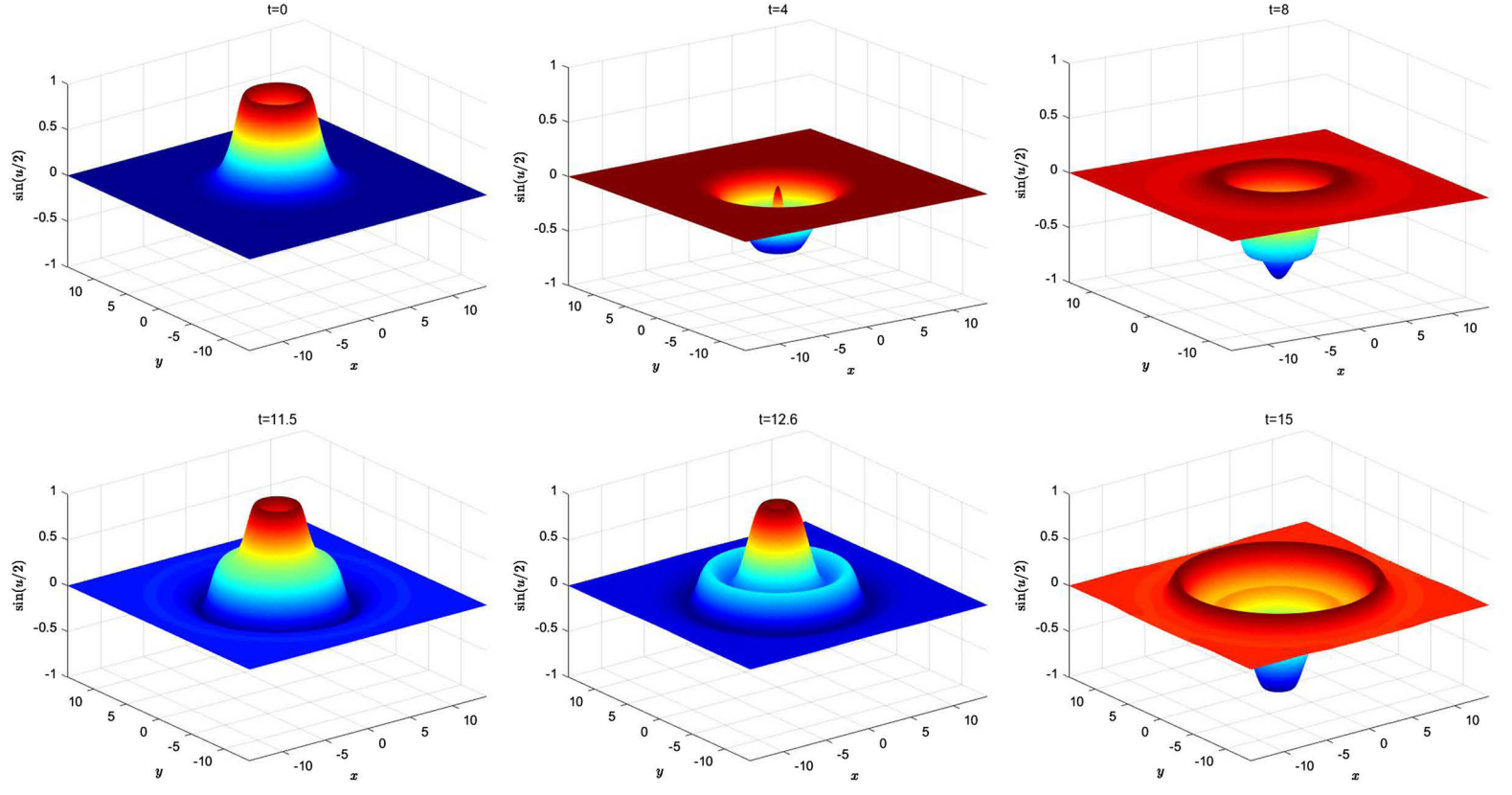


Fig. 3. Numerical simulations of $\sin(u/2)$ at times $t = 0, 4, 8, 11.5, 12.6, 15$ respectively, computed by the FCTI-3 scheme with $N_x = N_y = 256$ and $\Delta t = 0.005$ in Example 5.

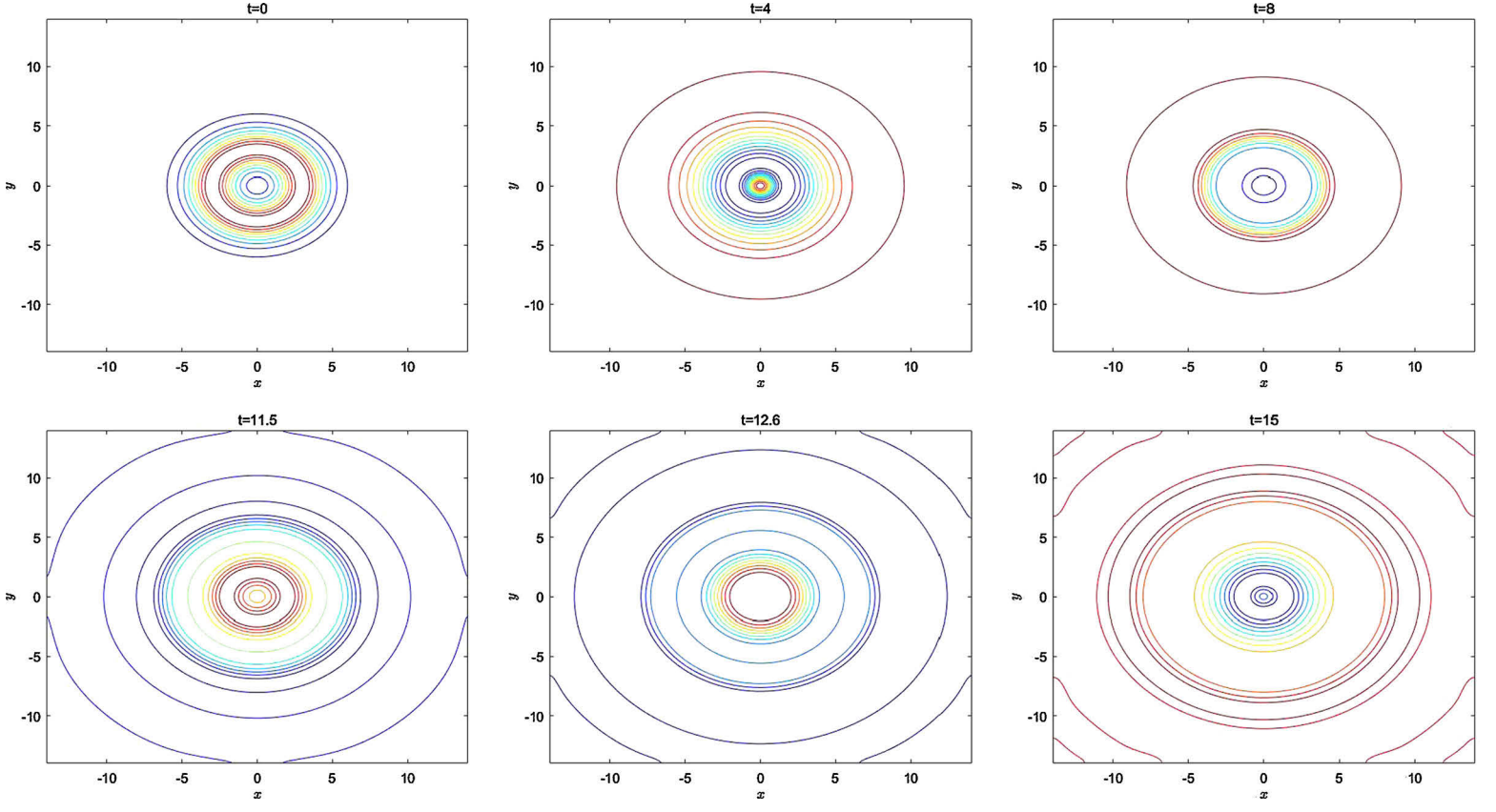


Fig. 4. Contour profiles of the simulated $\sin(u/2)$ at times $t = 0, 4, 8, 11.5, 12.6, 15$ respectively, computed by the FCTI-3 scheme with $N_x = N_y = 256$ and $\Delta t = 0.005$ in Example 5.

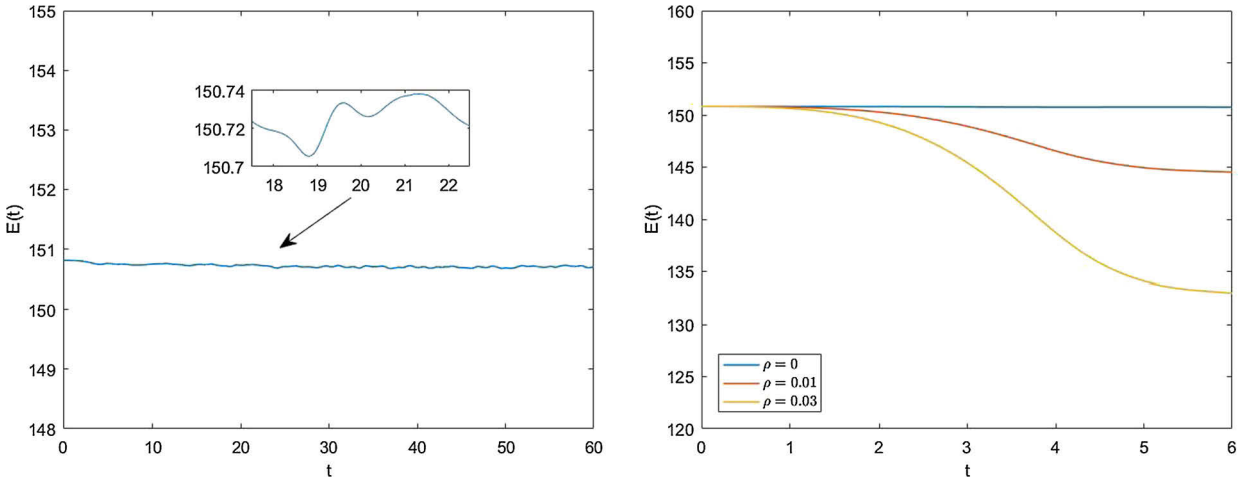


Fig. 5. Left: energy evolution of the approximate solution in the undamped case during the time interval $[0, 60]$ in Example 5; right: comparison of the energy evolutions of the approximate solutions under different damping coefficients during the time interval $[0, 6]$ in Example 5. All results are computed by the FCTI-3 scheme with $N_x = N_y = 256$ and $\Delta t = 0.005$. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

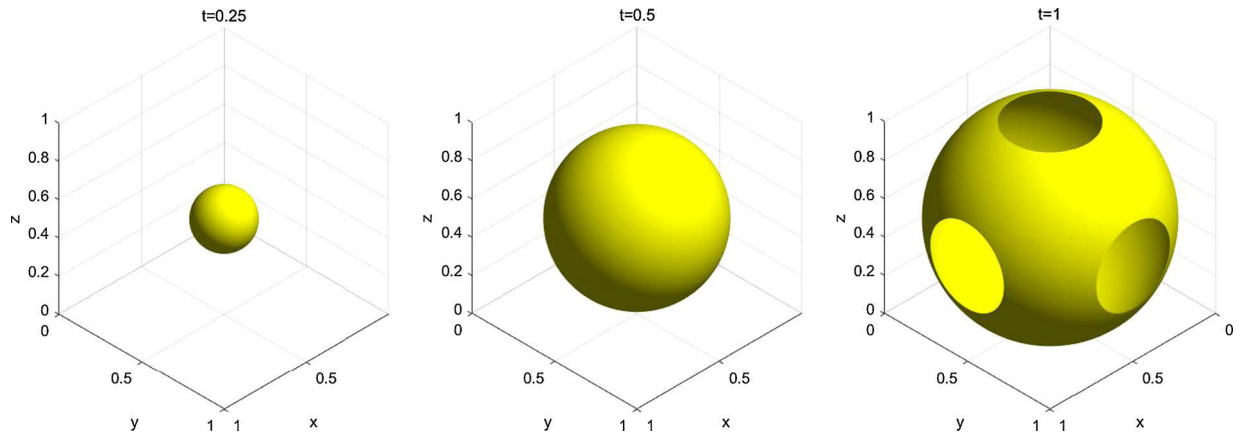


Fig. 6. Visualization of the simulated isosurfaces of $u \equiv 0.05$ of the expanding sphere problem with $\beta = 10$ and $r = 0.5$ at times $t = 0.25, 0.5, 1$ respectively, computed by the FCTI-3 scheme with $N_x = N_y = N_z = 128$ and $\Delta t = 1/256$ in Example 6.

We solve this third order problem using the FCTI-3 scheme with $N_x = N_y = 256$ and $\Delta t = 0.005$. Fig. 7 shows the approximate solution of the wave u for the case of $D = 0.625$ and $\beta = 0.2$ at some typical times and Fig. 8-(middle) report the corresponding evolution of the amplitude of u $(\max_{(x,y)\in\Omega}(u(x, y, t)))$ and $\min_{(x,y)\in\Omega}(u(x, y, t))))$ during the time interval $[0, 25]$. The crests in Fig. 8-middle appear at times t =2.66, 4.93, 7.305, 9.695, 12.035, 14.385, 16.79, 19.14, 21,485, 23.88 and the troughs at times $t = 1$, 1.675, 3.83, 6.255, 8.58, 10.945, 13.32, 15.68, 18.03, 20.405, 22.77. It can be clearly observed that u shows a quasi-periodic behavior and its amplitude gradually shrinks to zero along the time. Starting from the second line, from the first figure of each line to the first one of its successive line presents a quasi-period of changes of u in Fig. 7. We found that the duration of a period is approximately 4.702.

To illustrate the effect of β and D on the behaviors of the wave, we perform some further investigations. We first fix $D = 0.625$ and compare the results from the cases of $\beta = 0.1$, 0.2 and 0.4 respectively (see Fig. 8). We observe that all cases show quasi-periodic behaviors with almost the same period (approximately 4.702 for $\beta = 0.1$, 4.702 for $\beta = 0.2$ and 4.696 for $\beta = 0.4$) and the amplitude of u also shrinks at almost the same speed for all cases except some differences at the early stage of the evolution. Next we fix $\beta = 0.2$ and compare the results from the cases of $D = 0.375$, 0.625 and 0.875 respectively (see Fig. 9). It is found that all cases show quasi-periodic behaviors but with different periods, approximately 4.801 for $D = 0.375$, 4.702 for $D = 0.625$ and 4.522 for $D = 0.875$. The larger D is, the smaller the period is, and the much slower the amplitude of u shrinks.

6. Conclusions

Many important problems in science and engineering can be expressed as semilinear evolution equations of different orders. Generally speaking, no closed-form solutions are available for most of them, thus efficient numerical solution meth-

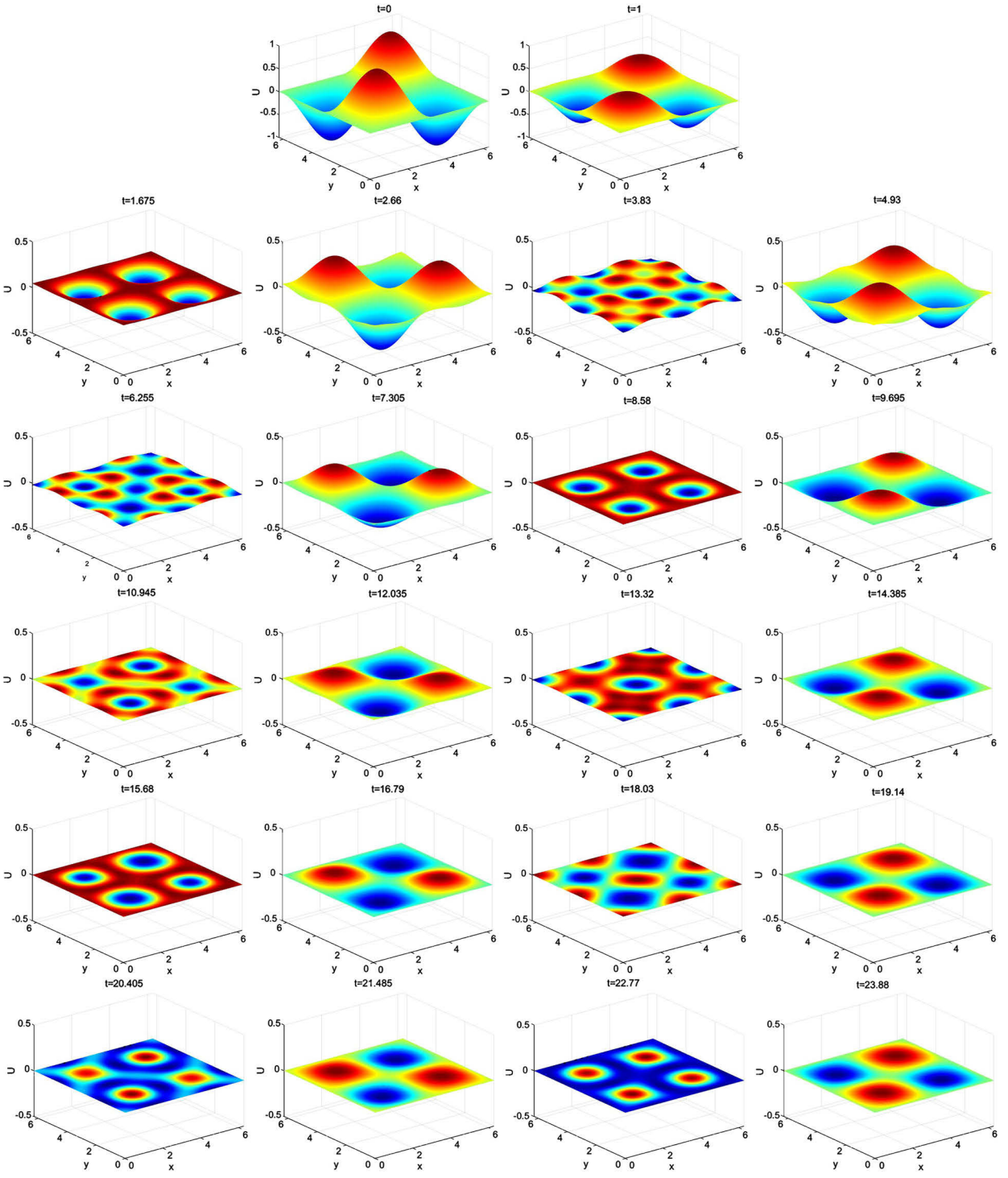


Fig. 7. Numerical simulation of the wave u for the case of $D = 0.625$ and $\beta = 0.2$ at times t =0, 1.0, 1.675, 2.66, 3.83, 4.93, 6.255, 7.305, 8.58, 9.695, 10.945, 12.035, 13.32, 14.385, 15.68, 16.79, 18.03, 19.14, 20.405, 21.485, 22.77, 23.88 respectively, computed by the FCTI-3 scheme with $N_x = N_y = 256$ and $\Delta t = 0.005$ in Example 7. Starting from the second line, from the first figure of each line to the first one of its successive line presents a quasi-period of changes of u. The amplitude of u gradually shrinks to zero.

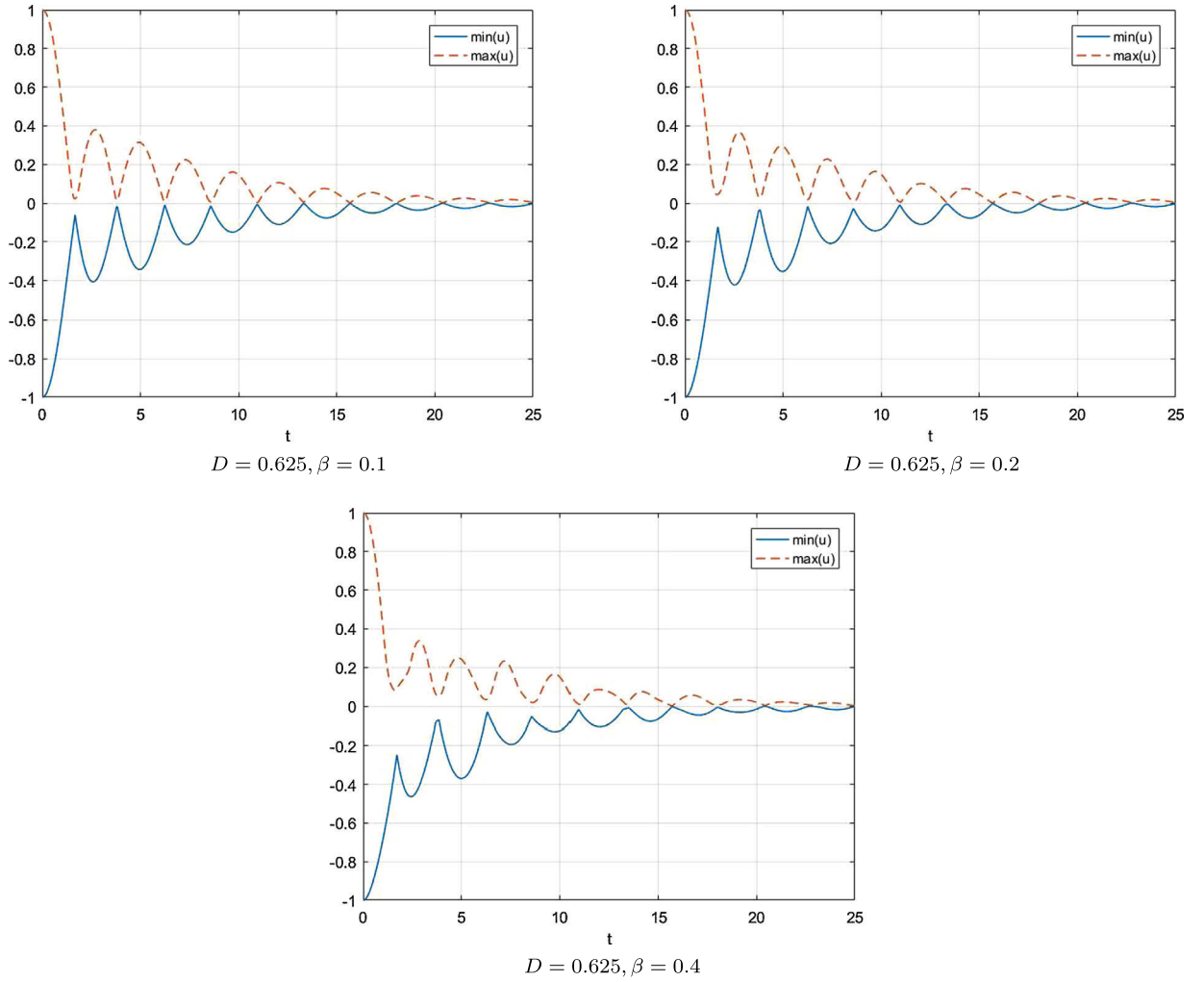
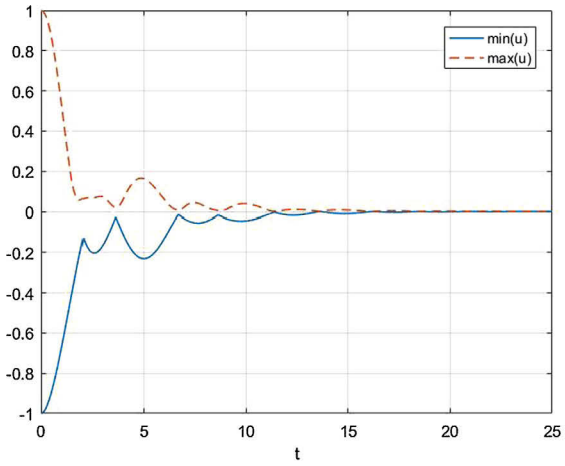


Fig. 8. Simulated evolutions of $\min(u)$ and $\max(u)$ during the time interval $[0, 25]$ with $D = 0.625$ and different values of β (0.1, 0.2, and 0.4 respectively), computed by the FCTI-3 scheme with $N_x = N_y = 256$ and $\Delta t = 0.005$ in Example 7. All cases show quasi-periodic behaviors with almost the same period (approximately 4.702 for $\beta = 0.1$, 4.702 for $\beta = 0.2$ and 4.696 for $\beta = 0.4$) and the amplitude of u also shrinks at almost the same speed in all cases except some differences at the early stage of the evolution.

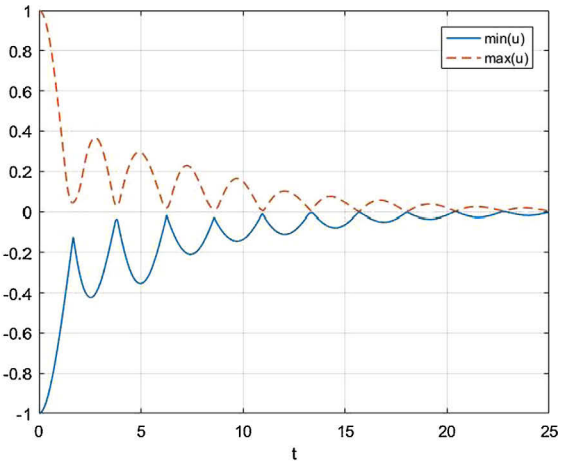
ods are highly desired. This paper is intended for devising a fast and accurate compact time integrator method, the FCTI method, for solving a family of general order semilinear equations in regular domains. More specifically, we discretize the model evolution equation by a fourth-order accurate compact difference scheme in space, giving rise to a system of ODEs. One advantage of such compact representation is that it can be implemented through FFT. We then express the solution of the discrete problem explicitly in terms of time integrator based on the theory of ODEs. The fully discrete method is finally obtained by approximating the boundary condition and the nonlinear term with multistep approximations and Lagrange interpolations. We also present the linear stability analysis of the first order FCTI scheme for second order evolution problems. Extensive numerical experiments with applications are performed to demonstrate accuracy, stability and efficiency of the proposed method. Although only Dirichlet and periodic boundary conditions are studied for the FCTI method in this paper, it also can be naturally generalized with some modifications to the case of Neumann boundary condition as stated in Remark 4. We also would like to note that the proposed FCTI method can be easily adapted to solve some nonstandard high order semilinear evolution equations as derived in Appendix A. In the end, the convergence analysis of the FCTI method remains an interesting topic to be studied in our next step.

Acknowledgements

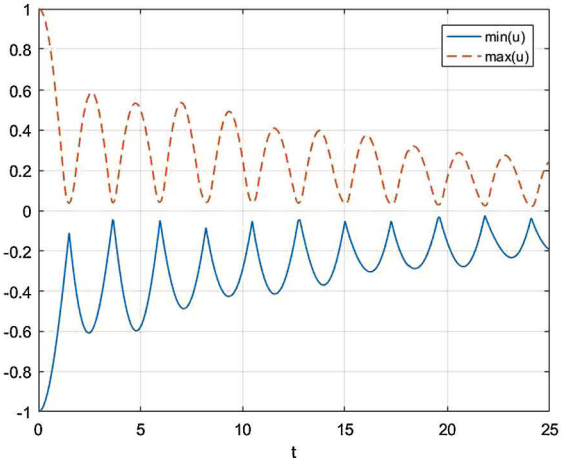
The authors would like to thank the referees for valuable and constructive comments, which led to a great improvement of an early version of the paper.



$D = 0.375, \beta = 0.2$



$D = 0.625, \beta = 0.2$



$D = 0.875, \beta = 0.2$

Fig. 9. Simulated evolutions of $\min(u)$ and $\max(u)$ during the time interval $[0, 25]$ with $\beta = 0.2$ and different values of D (0.375, 0.625, and 0.875 respectively), computed by the FCTI-3 scheme with $N_x = N_y = 256$ and $\Delta t = 0.005$ in Example 7. All cases show quasi-periodic behaviors but with different periods (approximately 4.801 for $D = 0.375$, 4.702 for $D = 0.625$ and 4.522 for $D = 0.875$). The larger D is, the smaller the period is and the much slower the amplitude of u shrinks.

Appendix A

We here present how to modify the original FCTI method to solve the third-order nonlinear evolution equation (5.8). For simplicity assume that the periodic boundary condition is imposed. Similar to (2.2), we use the fourth-order compact finite differences for spatial discretization to u^2 and obtain

$$\begin{cases} \dfrac{1}{12}\left((u^2_{i-1,j})^{xx} + 10(u^2_{i,j})^{xx} + (u^2_{i+1,j})^{xx}\right) = \dfrac{1}{h_x^2}(u^2_{i-1,j} - 2u^2_{i,j} + u^2_{i+1,j}), \\ \dfrac{1}{12}((u^2_{i,j-1})^{yy} + 10(u^2_{i,j})^{yy} + (u^2_{i,j+1})^{yy}) = \dfrac{1}{h_y^2}\left(u^2_{i,j-1} - 2u^2_{i,j} + u^2_{i,j+1}\right), \end{cases} \tag{A.1}$$

for $i = 0, 1, \cdots N_x - 1$, $j = 0, 1, \cdots N_y - 1$. In order to write the above scheme in matrix notation, define

$$\boldsymbol{U}^2 := \boldsymbol{U} \odot \boldsymbol{U} = (u^2_{i-1,j-1})_{N_x \times N_y} = \begin{pmatrix} u^2_{0,0} & u^2_{0,1} & \cdots & u^2_{0,N_y-1} \\ u^2_{1,0} & u^2_{1,1} & \cdots & u^2_{1,N_y-1} \\ \vdots & \vdots & \ddots & \vdots \\ u^2_{N_x-1,0} & u^2_{N_x-1,1} & \cdots & u^2_{N_x-1,N_y-1} \end{pmatrix}_{N_x \times N_y}$$

and define $(\boldsymbol{U}^2)^{xx} = ((u^2_{i-1,j-1})^{xx})_{N_x \times N_y}$ and $(\boldsymbol{U}^2)^{yy} = ((u^2_{i-1,j-1})^{yy})_{N_x \times N_y}$ correspondingly.

Then the above difference scheme can be formulated as

$$\boldsymbol{A}_x(\boldsymbol{U}^2)^{xx} = \frac{1}{D}\boldsymbol{B}_x\boldsymbol{U}^2, \qquad \boldsymbol{A}_y\textcircled{y}(\boldsymbol{U}^2)^{yy} = \frac{1}{D}\boldsymbol{B}_y\textcircled{y}\boldsymbol{U}^2, \tag{A.2}$$

where A_x, A_y and B_x, B_y are given in (2.26). Thus we can obtain a semi-discretization in space of (5.8) as follows:

$$\begin{aligned}\boldsymbol{U}'''(t) + \boldsymbol{U}''(t) - \frac{1}{D}\left(\boldsymbol{A}_x^{-1}\textcircled{x}\boldsymbol{B}_x\textcircled{x}\boldsymbol{U}'(t) + \boldsymbol{A}_y^{-1}\textcircled{y}\boldsymbol{B}_y\textcircled{y}\boldsymbol{U}'(t)\right) - \left(\boldsymbol{A}_x^{-1}\textcircled{x}\boldsymbol{B}_x\textcircled{x}\boldsymbol{U} + \boldsymbol{A}_y^{-1}\textcircled{y}\boldsymbol{B}_y\textcircled{y}\boldsymbol{U}\right)\\ = \frac{\beta}{D}\left(\boldsymbol{A}_x^{-1}\textcircled{x}\boldsymbol{B}_x\textcircled{x}\boldsymbol{U}^2 + \boldsymbol{A}_y^{-1}\textcircled{y}\boldsymbol{B}_y\textcircled{y}\boldsymbol{U}^2\right).\end{aligned} \tag{A.3}$$

Multiplying $\boldsymbol{P}_x^{-1}$ (defined in (2.29)) to the left hand side and $(\boldsymbol{P}_y^{-1})^T$ to the right hand side, we obtain

$$\boldsymbol{V}'''(t) + \boldsymbol{V}''(t) - \frac{1}{D}\boldsymbol{H}\odot\boldsymbol{V}'(t) - \boldsymbol{H}\odot\boldsymbol{V}(t) = \frac{\beta}{D}\boldsymbol{H}\odot\hat{F}(\boldsymbol{V}), \tag{A.4}$$

where $\boldsymbol{H} = (h_{ij})$ with its (i,j)-th entry given by

$$h_{ij} = -\frac{\frac{4D}{h_x^2}\sin^2(\frac{(i-1)\pi}{N_x})}{1-\frac{1}{3}\sin^2(\frac{(i-1)\pi}{N_x})} - \frac{\frac{4D}{h_y^2}\sin^2(\frac{(j-1)\pi}{N_y})}{1-\frac{1}{3}\sin^2(\frac{(j-1)\pi}{N_y})}, \qquad 1\le i\le N_x, 1\le j\le N_y,$$

$\boldsymbol{V} = \boldsymbol{P}_y^{-1}\textcircled{y}\boldsymbol{P}_x^{-1}\textcircled{x}\boldsymbol{U}$ and $\widehat{\boldsymbol{F}}(\boldsymbol{V}) = \boldsymbol{P}_y^{-1}\textcircled{y}\boldsymbol{P}_x^{-1}\textcircled{x}(\boldsymbol{U}^2)$. Then the corresponding FCTI-$(r+1)$ can be formulated by

$$\boldsymbol{U}_{m+1}^{(l)} = \boldsymbol{P}_y\textcircled{y}\boldsymbol{P}_x\textcircled{x}\left(\sum_{k=0}^{2}\boldsymbol{y}_k^{(l)}(\Delta t)\odot\left(\boldsymbol{P}_y^{-1}\textcircled{y}\boldsymbol{P}_x^{-1}\textcircled{x}\boldsymbol{U}_m^{(k)}\right) + \sum_{s=0}^{r}\left(\boldsymbol{P}_y^{-1}\textcircled{y}\boldsymbol{P}_x^{-1}\textcircled{x}(\boldsymbol{U}_m^2)\right)\odot\boldsymbol{S}_{r,s,l}^{F}\right), \quad 0\le l\le 2. \tag{A.5}$$

References

[1] Z. Asgari, S.M. Hosseini, Numerical solution of two-dimensional sine-Gordon and MBE models using Fourier spectral and high order explicit time stepping methods, Comput. Phys. Commun. 184 (3) (2013) 565–572.
[2] S. Allen, J.W. Cahn, A microscopic theory for antiphase boundary motion and its application to antiphase domain coarsening, Acta Metall. 27 (1979) 1084–1095.
[3] R. Camporesi, An Introduction to Linear Ordinary Differential Equations Using the Impulsive Response Method and Factorization, Springer-Verlag, Berlin, 2016.
[4] C. Canuto, M.Y. Hussaini, A. Quarteroni, T.A. Zang, Spectral Methods: Fundamentals in Single Domains, Springer-Verlag, Berlin, 2006.
[5] P. Chatzipantelidis, R. Lazarov, V. Thomée, Some error estimates for the finite volume element method for a parabolic problem, Comput. Methods Appl. Math. 13 (3) (2013) 251–279.
[6] F. de la Hoz, F. Vadillo, An exponential time differencing method for the nonlinear Schrödinger equation, Comput. Phys. Commun. 179 (7) (2008) 449–456.
[7] M. Dehghan, M. Abbaszadeh, A. Mohebbi, An implicit RBF meshless approach for solving the time fractional nonlinear sine-Gordon and Klein-Gordon equations, Eng. Anal. Bound. Elem. 50 (2015) 412–434.
[8] M. Dehghan, A. Ghesmati, Numerical simulation of two-dimensional sine-Gordon solitons via a local weak meshless technique based on the radial point interpolation method (RPIM), Comput. Phys. Commun. 181 (4) (2010) 772–786.
[9] H. Ding, Y. Zhang, A new fourth-order compact finite difference scheme for the two-dimensional second-order hyperbolic equation, J. Comput. Appl. Math. 230 (2) (2009) 626–632.
[10] K. Djidjeli, W.G. Price, E.H. Twizell, Numerical solutions of a damped sine-Gordon equation in two space variables, J. Eng. Math. 29 (4) (1995) 347–369.
[11] Q. Du, W.-X. Zhu, Stability analysis and application of the exponential time differencing schemes, J. Comput. Math. 22 (2) (2004) 200–209.
[12] Q. Du, W. Zhu, Analysis and applications of the exponential time differencing schemes and their contour integration modifications, BIT Numer. Math. 45 (2) (2005) 307–328.
[13] B.-Y. Guo, P.J. Pascual, M.A.J. Rodríguez, L. Vázquez, Numerical solution of the sine-Gordon equation, Appl. Math. Comput. 18 (1) (1986) 1–14.
[14] J.S. Hesthaven, S. Gottlieb, D. Gottlieb, Spectral Methods for Time-Dependent Problems, Cambridge Monographs on Applied and Computational Mathematics, vol. 21, Cambridge University Press, Cambridge, 2007.
[15] M. Hochbruck, C. Lubich, H. Selhofer, Exponential integrators for large systems of differential equations, SIAM J. Sci. Comput. 19 (5) (1998) 1552–1574.
[16] M. Hochbruck, A. Ostermann, Explicit exponential Runge-Kutta methods for semilinear parabolic problems, SIAM J. Numer. Anal. 43 (3) (2005) 1069–1090.
[17] J. Huang, J. Ju, B. Wu, A fast compact exponential time differencing method for semilinear parabolic equations with Neumann boundary conditions, Appl. Math. Lett. 94 (2019) 257–265.
[18] T.J. Hughes, The Finite Element Method: Linear Static and Dynamic Finite Element Analysis, Dover Publications, Inc., New York, 2012.
[19] R. Jiwari, Lagrange interpolation and modified cubic B-spline differential quadrature methods for solving hyperbolic partial differential equations with Dirichlet and Neumann boundary conditions, Comput. Phys. Commun. 193 (2015) 55–65.
[20] L. Ju, X. Liu, W. Leng, Compact implicit integration factor methods for a family of semilinear fourth-order parabolic equations, Discrete Contin. Dyn. Syst., Ser. B 19 (6) (2014) 1667–1687.
[21] L. Ju, Z. Wang, Exponential time differencing gauge method for incompressible viscous flows, Commun. Comput. Phys. 22 (2) (2017) 517–541.
[22] L. Ju, J. Zhang, L. Zhu, Q. Du, Fast explicit integration factor methods for semilinear parabolic equations, J. Sci. Comput. 62 (2) (2015) 431–455.
[23] A.K. Kassam, L.N. Trefethen, Fourth-order time-stepping for stiff PDEs, SIAM J. Sci. Comput. 26 (4) (2005) 1214–1233.
[24] S. Krogstad, Generalized integrating factor methods for stiff PDEs, J. Comput. Phys. 203 (1) (2005) 72–88.
[25] M. Li, T. Tang, B. Fornberg, A compact fourth-order finite difference scheme for the steady incompressible Navier-Stokes equations, Int. J. Numer. Methods Fluids 20 (10) (1995) 1137–1151.

[26] Y. Li, H.G. Lee, B. Xia, J. Kim, A compact fourth-order finite difference scheme for the three-dimensional Cahn–Hilliard equation, Comput. Phys. Commun. 200 (2016) 108–116.
[27] W. Liao, An implicit fourth-order compact finite difference scheme for one-dimensional Burgers' equation, Appl. Math. Comput. 206 (2) (2008) 755–764.
[28] W. Liao, J. Zhu, A.Q.M. Khaliq, An efficient high-order algorithm for solving systems of reaction-diffusion equations, Numer. Methods Partial Differ. Equ. 18 (3) (2002) 340–354.
[29] X. Ma, S. Shu, A. Zhou, Symmetric finite volume discretizations for parabolic problems, Comput. Methods Appl. Mech. Eng. 192 (39–40) (2003) 4467–4485.
[30] D. Mirzaei, M. Dehghan, Meshless local Petrov-Galerkin (MLPG) approximation to the two dimensional sine-Gordon equation, J. Comput. Appl. Math. 233 (10) (2010) 2737–2754.
[31] Q. Nie, F.Y.M. Wan, Y.-T. Zhang, X.-F. Liu, Compact integration factor methods in high spatial dimensions, J. Comput. Phys. 227 (10) (2008) 5238–5255.
[32] Q. Nie, Y.-T. Zhang, R. Zhao, Efficient semi-implicit schemes for stiff systems, J. Comput. Phys. 214 (2) (2006) 521–537.
[33] B. Pekmen, M. Tezer-Sezgin, Differential quadrature solution of nonlinear Klein-Gordon and sine-Gordon equations, Comput. Phys. Commun. 183 (8) (2012) 1702–1713.
[34] O.V. Rudenko, S.I. Soluyan, Theoretical Foundations of Nonlinear Acoustics, Springer-Verlag, New York, 1977.
[35] Q. Sheng, A.Q.M. Khaliq, D.A. Voss, Numerical simulation of two-dimensional sine-Gordon solitons via a split cosine scheme, Math. Comput. Simul. 68 (4) (2005) 355–373.
[36] Z.-Z. Sun, Compact difference schemes for heat equation with Neumann boundary conditions, Numer. Methods Partial Differ. Equ. 25 (6) (2009) 1320–1341.
[37] V. Thomée, Galerkin Finite Element Methods for Parabolic Problems, 2nd edition, Springer Series in Computational Mathematics, vol. 25, Springer-Verlag, Berlin, 2006.
[38] V. Varlamov, Long-time asymptotics of solutions of the third-order nonlinear evolution equation governing wave propagation in relaxing media, Q. Appl. Math. 58 (2) (2000) 201–218.
[39] V. Varlamov, The third-order nonlinear evolution equation governing wave propagation in relaxing media, Stud. Appl. Math. 99 (1) (1997) 25–48.
[40] D. Wang, L. Zhang, Q. Nie, Array-representation integration factor method for high-dimensional systems, J. Comput. Phys. 258 (2014) 585–600.
[41] H. Wang, Y. Zhang, X. Ma, J. Qiu, Y. Liang, An efficient implementation of fourth-order compact finite difference scheme for Poisson equation with Dirichlet boundary conditions, Comput. Math. Appl. 71 (9) (2016) 1843–1860.
[42] X. Wang, L. Ju, Q. Du, Efficient and stable exponential time differencing Runge-Kutta methods for phase field elastic bending energy models, J. Comput. Phys. 316 (2016) 21–38.
[43] P. Whalen, M. Brio, J.V. Moloney, Exponential time-differencing with embedded Runge-Kutta adaptive step control, J. Comput. Phys. 280 (2015) 579–601.
[44] S.-S. Xie, G.-X. Li, S. Yi, Compact finite difference schemes with high accuracy for one-dimensional nonlinear Schrödinger equation, Comput. Methods Appl. Mech. Eng. 198 (9–12) (2009) 1052–1060.
[45] Y. Zhang, Optimal error estimates of compact finite difference discretizations for the Schrödinger-Poisson system, Commun. Comput. Phys. 13 (5) (2013) 1357–1388.
[46] S. Zhao, J. Ovadia, X. Liu, Y.-T. Zhang, Q. Nie, Operator splitting implicit integration factor methods for stiff reaction-diffusion-advection systems, J. Comput. Phys. 230 (15) (2011) 5996–6009.
[47] L. Zhu, L. Ju, W. Zhao, Fast high-order compact exponential time differencing Runge-Kutta methods for second-order semilinear parabolic equations, J. Sci. Comput. 67 (3) (2016) 1043–1065.