

## Research Article

# New High-Order Compact ADI Algorithms for 3D Nonlinear Time-Fractional Convection-Diffusion Equation

Shuying Zhai,<sup>1</sup> Xinlong Feng,<sup>1</sup> and Zhifeng Weng<sup>2</sup>

<sup>1</sup> College of Mathematics and Systems Science, Xinjiang University, Urumqi 830046, China

<sup>2</sup> School of Mathematics and Statistics, Wuhan University, Wuhan 430072, China

Correspondence should be addressed to Shuying Zhai; zhaishuying123456@163.com and Xinlong Feng; fxlmath@gmail.com

Received 14 January 2013; Revised 8 April 2013; Accepted 15 April 2013

Academic Editor: Jun Jiang

Copyright © 2013 Shuying Zhai et al. 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.

Numerical approximations of the three-dimensional (3D) nonlinear time-fractional convection-diffusion equation is studied, which is firstly transformed to a time-fractional diffusion equation and then is solved by linearization method combined with alternating direction implicit (ADI) method. By using fourth-order Padé approximation for spatial derivatives and classical backward differentiation method for time derivative, two new high-order compact ADI algorithms with orders  $O(\tau^{\min(1+\alpha, 2-\alpha)} + h^4)$  and  $O(\tau^{2-\alpha} + h^4)$  are presented. The resulting schemes in each ADI solution step corresponding to a tridiagonal matrix equation can be solved by the Thomas algorithm which makes the computation cost effective. Numerical experiments are shown to demonstrate the high accuracy and robustness of two new schemes.

## 1. Introduction

In this paper, we consider numerical approximations of the following 3D nonlinear time-fractional convection-diffusion equation:

$$\begin{aligned} \frac{\partial^\alpha u}{\partial t^\alpha} \Big|_{\Omega \times (0, T]} \\ = (\Delta u - pu_x - qu_y - ru_z + su + \phi(u) + f) \Big|_{\Omega \times (0, T]}, \end{aligned} \quad (1)$$

with the initial and boundary conditions

$$\begin{aligned} u(x, y, z, 0) \Big|_{\Omega} &= u_0(x, y, z) \Big|_{\Omega}, \\ u(x, y, z, t) \Big|_{\partial\Omega \times (0, T]} &= 0, \end{aligned} \quad (2)$$

where  $0 < \alpha < 1$ ,  $p, q, r$ , and  $s$  are constants, and  $\phi(u)$  is some reasonable nonlinear function of  $u = u(x, y, z, t)$ .  $\Omega$  is a continuous convex domain in 3D space with suitable boundary conditions prescribed on its boundary  $\partial\Omega$ . The solution  $u$  and the source terms  $\phi(u)$  and  $f = f(x, y, z, t)$  are assumed to be sufficiently smooth and have the necessary continuous partial derivatives up to certain orders. The fractional derivative

$\partial^\alpha u / \partial t^\alpha$  is Grünward-Letnikov fractional derivative defined by

$$\frac{\partial^\alpha u}{\partial t^\alpha} = \frac{t^{-\alpha}}{\Gamma(1-\alpha)} u^0 + \frac{1}{\Gamma(2-\alpha)} \int_0^t \frac{\partial u(x, y, z, s)}{\partial s} \frac{d_s}{(t-s)^\alpha}. \quad (3)$$

It should be noted that Grünward-Letnikov fractional derivative will become Caputo fractional derivative when  $u^0 = 0$ .

Fractional differential equations (FDEs) have become popular in recent years both in mathematics and in applications. They have been increasingly used in physical and chemical processes [1–3], relaxation in polymer systems, dynamics of protein molecules, and the diffusion of contaminants in complex geological formations [4]. Because of the absence or appropriate mathematical methods, most of the analytical solutions for FDEs are difficult to obtain. However, several numerical methods have been proposed to solve them.

For the one-dimensional (1D) time FDEs, the schemes in [5–10] are convergent with order not higher than two for the space variable, and it is interesting to seek high-order numerical methods for FDEs. Recently, such problem was solved by the compact finite difference scheme with convergence order  $O(\tau + h^4)$  in [11], and a higher order

$O(\tau^{2-\alpha} + h^4)$  one can be found in [12, 13]. Lin and Xu [14], Jiang and Ma [15], and Wei et al. [16–18] proposed high order method based on the finite spectral method, the finite element method (FEM), and local discontinuous Galerkin (LDG) method for space discretization, respectively.

For the two-dimensional (2D) FDEs, Zhuang and Liu [19] considered a time-fractional diffusion equation on a finite domain, and a first-order implicit difference approximation was proposed to solve the equation. Additional references [20–22] treat explicit or/and implicit finite difference methods (FDMs) for 2D time FDEs, as well as FEM [23], LDG method [24], and spectral method [25] for such problems. Recently, Zhang and Sun [26] presented new numerical techniques for time FDEs and ADI schemes were considered, but both methods have only second-order accuracy in spatial direction. Then, Cui [27] proposed a high-order ADI compact finite difference scheme with order  $O(\tau^{2-\alpha} + h^4)$  for time-fractional diffusion equation. References [28–34] address FDM and FEM for space FDEs, respectively.

In this paper, we focus on the 3D nonlinear time-fractional convection-diffusion equation. To the authors' knowledge, the research for the 3D case is very lacking. The reason, of necessity, is that 3D problems present many computational challenges because of the prohibitive memory and time requirement often necessary to solve them on grids of sufficient resolution. Traditional numerical schemes have low accuracy and thus require fine discretization. Moreover, being opposite to differential equations of integer order, in which derivatives depend only on the local behaviour of the function, fractional differential equations accumulate the whole information of the function in a weighted form. This makes realistic 3D simulations of FDEs prohibitively expensive. Therefore, it is interesting to discuss high-order numerical methods for 3D time-fractional convection-diffusion equation.

It is well known that Padé approximation has the advantages of the fourth-order accuracy to approximate the second-order derivatives and of keeping the desirable tridiagonal nature of the finite difference equations. So, the main purpose of this paper is to solve the nonlinear time-fractional convection-diffusion problem using the Padé approximation combined with ADI method and the one-dimensional tridiagonal Thomas algorithm. However, the convection terms will impede implementation of tridiagonal Thomas algorithm. So, we propose a transformation based on the compact finite difference method [35]. In this method, the original time-fractional convection-diffusion equation is transformed to a time-fractional diffusion equation; the resulting time-fractional diffusion problem is then solved by an efficient high-order method. Another advantage of our method is that the nonlinear term is transformed into a linear term which makes the computation cost effective and the approximation solutions more accurate.

This paper is organized as follows. In Section 2, two new high-order ADI algorithms are presented for solving 3D nonlinear time-fractional convection-diffusion equation. In Section 3, numerical examples are carried out to verify the

high efficiency of our method. Finally, conclusions are drawn in Section 4.

## 2. Two New High-Order Compact ADI Algorithms

For simplicity of the presentation, we consider a rectangular domain  $\Omega = [0, 1]^3$ , which is divided into an  $N \times N \times N$  mesh with the spatial step size  $h_x = h_y = h_z = h = 1/N$ , while the time step size is chosen as  $\tau = T/M$  and  $M$  is the number of time coordinate direction.

To obtain an efficient scheme, it is important to approximate the nonlinear term  $\phi(u)$  in (1) explicitly. Otherwise, one would need to perform a nonlinear iteration in each time step which is quite time consuming. Let  $u^m = u(x, y, z, t_m)$  and  $f^m = f(x, y, z, t_m)$ . The following linear method is used to transform  $\phi(u)$  into linear term:

$$\phi(u^{m+1}) = 2\phi(u^m) - \phi(u^{m-1}) + O(\tau^2), \quad 1 \leq m \leq M-1. \quad (4)$$

Next we discuss the linear equation

$$\begin{aligned} \frac{\partial^\alpha u^{m+1}}{\partial t^\alpha} = & \Delta u^{m+1} - pu_x^{m+1} - qu_y^{m+1} - ru_z^{m+1} \\ & + su^{m+1} + 2\phi(u^m) - \phi(u^{m-1}) + f^{m+1} \end{aligned} \quad (5)$$

for  $1 \leq m \leq M-1$ .

In order to keep the fourth-order accuracy in space and the tridiagonal nature of the scheme by utilizing the Padé approximation, we first introduce a transformation that is similar to that of Liao [35] to eliminate the convection terms in (1).

Let

$$u = \lambda_1(x) \lambda_2(y) \lambda_3(z) v, \quad (6)$$

where  $v = v(x, y, z, t)$  and  $\lambda_1(x)$ ,  $\lambda_2(y)$ , and  $\lambda_3(z)$  are functions to be determined. Substituting (6) into the first formula of (1), we have

$$\begin{aligned} & \lambda_1 \lambda_2 \lambda_3 \left( \frac{\partial^\alpha v}{\partial t^\alpha} - \Delta v \right) \\ = & (\lambda_{1xx} \lambda_2 \lambda_3 - p \lambda_{1x} \lambda_2 \lambda_3 + \lambda_1 \lambda_{2yy} \lambda_3 \\ & - q \lambda_1 \lambda_{2y} \lambda_3 + \lambda_1 \lambda_2 \lambda_{3zz} - r \lambda_1 \lambda_2 \lambda_{3z}) v \\ & + (2\lambda_{1x} - p \lambda_1) \lambda_2 \lambda_3 v_x + (2\lambda_{2y} - q \lambda_2) \lambda_1 \lambda_3 v_y \\ & + (2\lambda_{3z} - r \lambda_3) \lambda_1 \lambda_2 v_z. \end{aligned} \quad (7)$$

Setting the coefficients of  $v_x$ ,  $v_y$ , and  $v_z$  to zero leads to

$$\begin{aligned} 2\lambda_{1x} - p\lambda_1 &= 0, \\ 2\lambda_{2y} - q\lambda_2 &= 0, \\ 2\lambda_{3z} - r\lambda_3 &= 0. \end{aligned} \quad (8)$$

Then, we have

$$\lambda_1(x) = e^{(p/2)x}, \quad \lambda_2(y) = e^{(q/2)y}, \quad \lambda_3(z) = e^{(r/2)z}. \quad (9)$$

Set

$$\lambda(x, y, z) = \lambda_1(x) \lambda_2(y) \lambda_3(z) = e^{(p/2)x + (q/2)y + (r/2)z}. \quad (10)$$

Combining (1) and (5)–(10), we obtain the following time-fractional diffusion equation satisfied by  $v$ :

$$\begin{aligned} \frac{\partial^\alpha v^{m+1}}{\partial t^\alpha} \Big|_\Omega &= \Delta v^{m+1} + \left( s - \frac{p^2 + q^2 + r^2}{4} \right) v^{m+1} \\ &\quad + \lambda^{-1} (2\phi(\lambda v^m) - \phi(\lambda v^{m-1}) + f^{m+1}) \Big|_\Omega, \\ m &\geq 1, \end{aligned} \quad (11)$$

$$v(x, y, z, 0) \Big|_\Omega = \lambda^{-1} u_0(x, y, z) \Big|_\Omega,$$

$$v(x, y, z, t^m) \Big|_{\partial\Omega} = 0, \quad m \geq 0.$$

In the following, we will introduce a high-order compact difference method for (11).

The time-fractional derivative  $\partial^\alpha v / \partial t^\alpha$  at  $t_{m+1}$  is estimated by

$$\begin{aligned} \frac{\partial^\alpha v^{m+1}}{\partial t^\alpha} &= \frac{t_{m+1}^{-\alpha}}{\Gamma(1-\alpha)} v^0 + \frac{1}{\Gamma(1-\alpha)} \\ &\quad \times \sum_{l=0}^m \int_{t_l}^{t_{l+1}} \frac{\partial v(x, y, z, \xi)}{\partial \xi} \frac{d\xi}{(t_{m+1} - \xi)^\alpha} \\ &= \frac{t_{m+1}^{-\alpha}}{\Gamma(1-\alpha)} v^0 + \frac{1}{\Gamma(1-\alpha)} \\ &\quad \times \sum_{l=0}^m \frac{v^{l+1} - v^l}{\tau} \int_{t_l}^{t_{l+1}} \frac{d\xi}{(t_{m+1} - \xi)^\alpha} + E_\tau^{m+1} \\ &= \frac{t_{m+1}^{-\alpha}}{\Gamma(1-\alpha)} v^0 + \frac{1}{\Gamma(2-\alpha)} \sum_{l=0}^m b_l \frac{v^{m+1-l} - v^{m-l}}{\tau^\alpha} + E_\tau^{m+1}, \end{aligned} \quad (12)$$

where  $b_l = (l+1)^{1-\alpha} - l^{1-\alpha}$  and the truncation error  $E_\tau^{m+1}$  has the following form:

$$\begin{aligned} E_\tau^{m+1} &= \frac{1}{\Gamma(1-\alpha)} \\ &\quad \times \sum_{l=0}^m \int_{t_l}^{t_{l+1}} \left( \frac{\partial v(x, y, z, \xi)}{\partial \xi} - \frac{v^{l+1} - v^l}{\tau} \right) \\ &\quad \times \frac{d\xi}{(t_{m+1} - \xi)^\alpha}. \end{aligned} \quad (13)$$

If we expend  $v(x, y, z, \xi)$  in Taylor series about  $\xi$  at  $t_{l+1/2}$  in (13) and use the analysis in [14], we will get

$$\begin{aligned} E_\tau^{m+1} &\leq C_v \left( \frac{1}{\Gamma(1-\alpha)} \sum_{l=0}^m \int_{t_l}^{t_{l+1}} \frac{t_{l+1} + t_l - 2\xi}{(t_{m+1} - \xi)^\alpha} d\xi + O(\tau^2) \right) \\ &\leq C_v \tau^{2-\alpha}, \end{aligned} \quad (14)$$

where  $C_v$  is a positive constant only depending on  $v$ .

As for the spatial derivatives, we can use the fourth-order Padé scheme. That is, the spatial derivatives can be approximated by

$$\begin{aligned} \Delta v &= \left( \frac{\delta_{x^2}}{h^2 (1 + (1/12) \delta_{x^2})} + \frac{\delta_{y^2}}{h^2 (1 + (1/12) \delta_{y^2})} \right. \\ &\quad \left. + \frac{\delta_{z^2}}{h^2 (1 + (1/12) \delta_{z^2})} \right) \times v + O(h^4), \end{aligned} \quad (15)$$

where  $\delta_x^2$ ,  $\delta_y^2$ , and  $\delta_z^2$  are the second-order central difference operators along  $x$ - and  $y$ - and  $z$ -directions, respectively.

Combining with (12) and (15), then the implicit compact finite difference scheme for (11) is given as follows:

$$\begin{aligned} &\frac{((m+1)\tau)^{-\alpha}}{\Gamma(1-\alpha)} v_{ijk}^0 + \frac{1}{\Gamma(2-\alpha)} \sum_{l=0}^m b_l \frac{v_{ijk}^{m+1-l} - v_{ijk}^{m-l}}{\tau^\alpha} \\ &= \left[ \frac{\delta_{x^2}}{h^2 (1 + (1/12) \delta_{x^2})} + \frac{\delta_{y^2}}{h^2 (1 + (1/12) \delta_{y^2})} \right. \\ &\quad \left. + \frac{\delta_{z^2}}{h^2 (1 + (1/12) \delta_{z^2})} \right] v_{ijk}^{m+1} \\ &\quad + \left( s - \frac{p^2 + q^2 + r^2}{4} \right) v_{ijk}^{m+1} \\ &\quad + \lambda^{-1} (2\phi_{ijk}^m - \phi_{ijk}^{m-1} + f_{ijk}^{m+1}), \\ &1 \leq i, j, k \leq N-1, \quad 1 \leq m \leq M-1, \end{aligned} \quad (16)$$

with the initial discretization

$$v_{ijk}^0 = \lambda(x_i, y_j, z_k) u_0(x_i, y_j, z_k), \quad 0 \leq i, j, k \leq N, \quad (17)$$

and the boundary discretization

$$\begin{aligned} v_{0jk}^m &= v_{Njk}^m = v_{i0k}^m = v_{iNk}^m = v_{ij0}^m = v_{ijn}^m = 0, \\ 0 &\leq i, j, k \leq N, \quad 0 \leq m \leq M, \end{aligned} \quad (18)$$

where  $v_{ijk}^m$  denotes  $v(ih, jh, kh, m\tau)$ . In the following,  $v^m$  will be written in short for  $v_{ijk}^m$  if there is no confusion about the notation.

TABLE 1:  $T = 1$ ,  $N = 40$ , and  $\alpha = 0.5$  with different  $M$  for Problem 5.

	$M$	10	20	30	40	50
Algorithm 1	$L^\infty$ -error	$8.5329e-1$	$3.9114e-1$	$2.1763e-1$	$1.4039e-1e$	$9.9234e-2$
	$L^\infty$ -rate	—	1.1253	1.4459	1.5239	1.5547
	$L^2$ -error	$2.7920e-1$	$1.2721e-1$	$7.0539e-2$	$4.5420e-2$	$3.2070e-2$
	$L^2$ -rate	—	1.1341	1.4544	1.5302	1.5596
Algorithm 2	$L^\infty$ -error	$1.6081e-1$	$2.4544e-2$	$8.4991e-3$	$4.0461e-3$	$2.2764e-3$
	$L^\infty$ -rate	—	2.7119	2.6156	2.5800	2.5776
	$L^2$ -error	$5.2043e-2$	$7.8663e-3$	$2.7077e-3$	$1.2840e-3$	$7.2062e-3$
	$L^2$ -rate	—	2.7259	2.6303	2.5936	2.5884

TABLE 2:  $T = 0.01$ ,  $M = 100$ , and  $\alpha = 0.5$  with different  $N$  using Algorithm 2 for Problem 5.

$N$	5	10	20	30	40
$L^\infty$ -error	$4.8628e-4$	$3.5004e-5$	$2.1860e-6$	$4.3559e-7$	$1.4123e-7$
$L^\infty$ -rate	—	3.7962	4.0012	3.9784	3.9152
$L^2$ -error	$1.5138e-4$	$1.0654e-5$	$7.1348e-7$	$1.4566e-7$	$4.7836e-8$
$L^2$ -rate	—	3.8287	3.9004	3.9186	3.8706

By reformulating (16), we obtain an equivalent form

$$v^{m+1} = \left[ \frac{(\kappa_0 h^2)^{-1} \delta_{x^2}}{(1 + (1/12) \delta_{x^2})} + \frac{(\kappa_0 h^2)^{-1} \delta_{y^2}}{(1 + (1/12) \delta_{y^2})} + \frac{(\kappa_0 h^2)^{-1} \delta_{z^2}}{(1 + (1/12) \delta_{z^2})} \right] \times v^{m+1} + F^{m+1}, \quad 1 \leq m \leq M-1, \quad (19)$$

where

$$\kappa_0 = \frac{\tau^{-\alpha}}{\Gamma(2-\alpha)} + \frac{p^2 + q^2 + r^2}{4} - s, \quad F^{m+1} = \frac{1}{\kappa_0} \left[ \frac{\tau^{-\alpha} v^m}{\Gamma(2-\alpha)} - \frac{\tau^{-\alpha} \sum_{l=1}^m b_l (v^{m+1-l} - v^{m-l})}{\Gamma(2-\alpha)} - \frac{((m+1)\tau)^{-\alpha} v^0}{\Gamma(1-\alpha)} + \lambda^{-1} (2\phi^m - \phi^{m-1} + f^{m+1}) \right]. \quad (20)$$

Due to the nature of three-time-level scheme, we need an additional method to compute the numerical solution at the first time step.

For the special case  $m = 0$ , the scheme simply reads

$$v^1 = \left[ \frac{(\kappa_0 h^2)^{-1} \delta_{x^2}}{(1 + (1/12) \delta_{x^2})} + \frac{(\kappa_0 h^2)^{-1} \delta_{y^2}}{(1 + (1/12) \delta_{y^2})} + \frac{(\kappa_0 h^2)^{-1} \delta_{z^2}}{(1 + (1/12) \delta_{z^2})} \right] \times v^1 + F^1 \quad (21)$$

with

$$F^1 = \frac{1}{\kappa_0} \left[ \frac{\alpha \tau^{-\alpha}}{\Gamma(2-\alpha)} v^0 + \lambda^{-1} (\phi^1 + f^1) \right]. \quad (22)$$

It is clear that (21) must be solved using some relaxation procedure because  $\phi^1$  is nonlinear.

For simplicity of the presentation, we write (19) and (21) in a unified form

$$v^m = \left[ \frac{(\kappa_0 h^2)^{-1} \delta_{x^2}}{(1 + (1/12) \delta_{x^2})} + \frac{(\kappa_0 h^2)^{-1} \delta_{y^2}}{(1 + (1/12) \delta_{y^2})} + \frac{(\kappa_0 h^2)^{-1} \delta_{z^2}}{(1 + (1/12) \delta_{z^2})} \right] \times v^m + F^m, \quad 1 \leq m \leq M. \quad (23)$$

Multiplying  $(1 + (1/12) \delta_{x^2})(1 + (1/12) \delta_{y^2})(1 + (1/12) \delta_{z^2})$  on both sides of (23), we have

$$\begin{aligned} & \left(1 + \frac{1}{12} \delta_{x^2}\right) \left(1 + \frac{1}{12} \delta_{y^2}\right) \left(1 + \frac{1}{12} \delta_{z^2}\right) v^m \\ &= (\kappa_0 h^2)^{-1} \delta_{x^2} \left(1 + \frac{1}{12} \delta_{y^2}\right) \left(1 + \frac{1}{12} \delta_{z^2}\right) v^m \\ &+ (\kappa_0 h^2)^{-1} \delta_{y^2} \left(1 + \frac{1}{12} \delta_{x^2}\right) \left(1 + \frac{1}{12} \delta_{z^2}\right) v^m \\ &+ (\kappa_0 h^2)^{-1} \delta_{z^2} \left(1 + \frac{1}{12} \delta_{x^2}\right) \left(1 + \frac{1}{12} \delta_{y^2}\right) v^m \\ &+ \left(1 + \frac{1}{12} \delta_{x^2}\right) \left(1 + \frac{1}{12} \delta_{y^2}\right) \left(1 + \frac{1}{12} \delta_{z^2}\right) F^m, \end{aligned} \quad (24)$$

$1 \leq m \leq M.$

In the following, two new high-order algorithms are proposed for (24) that will coincide with the ADI methods

TABLE 3:  $T = 2$ ,  $N = 20$ , and  $\alpha = 0.2$  with different  $M$  for Problem 6.

	$M$	10	20	30	40	50
Algorithm 1	$L^\infty$ -error	$5.5947e-1$	$2.9608e-1$	$1.9977e-1$	$1.5025e-1$	$1.2017e-1$
	$L^\infty$ -rate	—	0.9181	0.9703	0.9903	1.0012
	$L^2$ -error	$1.2173e-1$	$6.4253e-2$	$4.3325e-2$	$3.2575e-2$	$2.6049e-2$
	$L^2$ -rate	—	0.9219	0.9720	0.9913	1.0019
Algorithm 2	$L^\infty$ -error	$6.1711e-2$	$1.5422e-2$	$6.7626e-3$	$3.7638e-3$	$2.3877e-3$
	$L^\infty$ -rate	—	2.0006	2.0331	2.0369	2.0396
	$L^2$ -error	$1.3403e-2$	$3.3369e-3$	$1.4631e-3$	$8.1426e-4$	$5.1651e-4$
	$L^2$ -rate	—	2.0060	2.0335	2.0370	2.0398

employed. First, some finite difference operations are introduced that will be used hereafter as follows:

$$\begin{aligned}
L_x &= 1 + \frac{1}{12}\delta_x^2 - \frac{1}{\kappa_0 h^2}\delta_x^2, \\
L_y &= 1 + \frac{1}{12}\delta_y^2 - \frac{1}{\kappa_0 h^2}\delta_y^2, \\
L_z &= 1 + \frac{1}{12}\delta_z^2 - \frac{1}{\kappa_0 h^2}\delta_z^2, \\
L_F &= \left(1 + \frac{1}{12}\delta_x^2\right)\left(1 + \frac{1}{12}\delta_y^2\right)\left(1 + \frac{1}{12}\delta_z^2\right), \\
L_H &= \left(\left(1 + \frac{1}{12}\delta_x^2\right)\frac{\delta_y^2\delta_z^2}{(\kappa_0 h^2)^2} + \left(1 + \frac{1}{12}\delta_y^2\right)\frac{\delta_x^2\delta_z^2}{(\kappa_0 h^2)^2}\right. \\
&\quad \left.+ \left(1 + \frac{1}{12}\delta_z^2\right)\frac{\delta_x^2\delta_y^2}{(\kappa_0 h^2)^2}\right) \\
&\quad - \frac{\delta_x^2\delta_y^2\delta_z^2}{(\kappa_0 h^2)^3}.
\end{aligned} \tag{25}$$

*Algorithm 1.* (1) For  $m \geq 1$ , add splitting term

$$\mathbf{I} : L_H(v^m - v^{m-1}) = O(\tau^{1+2\alpha}) \tag{26}$$

to the left-hand side of (24), and rewrite it into an equivalent form

$$\mathbf{SI} : L_x L_y L_z v^m = L_F F^m + L_H v^{m-1}. \tag{27}$$

(2) Introducing two intermediate variables  $v^*$  and  $v^{**}$ , scheme **SI** can be solved in three steps as

$$L_x \Delta v^* = L_F F^m + L_H v^{m-1}, \tag{28a}$$

$$L_y \Delta v^{**} = v^*, \tag{28b}$$

$$L_z v = v^{**}. \tag{28c}$$

Note that the intermediate values of  $v^{**}$  and  $v^*$  on the boundary are easily obtained from (28c) and (28b).

*Algorithm 2.* (1) For the first time step,  $v^1$  can be obtained from Algorithm 1.

(2) For  $m \geq 2$ , add splitting term

$$\mathbf{II} : L_H(v^m - 2v^{m-1} + v^{m-2}) = O(\tau^{2+2\alpha}) \tag{29}$$

to the left-hand side of (24), and rewrite it into an equivalent form

$$\mathbf{SII} : L_x L_y L_z v^m = L_F F^m + L_H(2v^{m-1} - v^{m-2}). \tag{30}$$

(3) Use the same way as the second step of Algorithm 1.

*Remark 3.* It is clear that the splitting error of **II** is higher order in  $\tau$  than the local error for the time-fractional derivative approximation, so, for the time accuracy, scheme **SII** can ensure the local truncation error to be  $2 - \alpha$  order. However, as for the splitting term **I**, it should be  $1 + \alpha \geq 2 - \alpha$  in order to maintain the time accuracy. That is, the order of local error for scheme **SI** is determined by the following equation:

$$\min(1 + \alpha, 2 - \alpha) = \begin{cases} 1 + \alpha, & \alpha \in (0, 0.5], \\ 2 - \alpha, & \alpha \in (0.5, 1). \end{cases} \tag{31}$$

Based on the previous analysis, we propose two linear ADI schemes, and the truncation errors are  $O(\tau^{\min(1+\alpha, 2-\alpha)} + h^4)$  and  $O(\tau^{(2-\alpha)} + h^4)$ , respectively. The resulting ADI schemes in each ADI solution step corresponding to a strictly diagonally dominant matrix equation which can be solved using the one-dimensional tridiagonal Thomas algorithm with a considerable saving in computing time.

*Remark 4.* The previous analysis shows that splitting term **II** is better than **I** when  $\alpha \in (0, 0.5]$ . However, when  $\alpha \in (0.5, 1)$ , the splitting error is essentially negligible if the time step size is small enough, and so both schemes can obtain the same accuracy in this case. Nevertheless, numerical experiments in Section 3 show that scheme **SII** is much better than scheme **SI**, regardless of the value of  $\alpha$ . We would like to comment the fact that if the splitting error is much larger than the truncation error, as pointed out by Douglas and Kim [36], a good splitting term will play an important role in the accuracy of the solution, especially for the analytical solutions that decrease exponentially in time. So, we think that scheme **SII** is a better choice even with an additional computational cost than scheme **SI**.

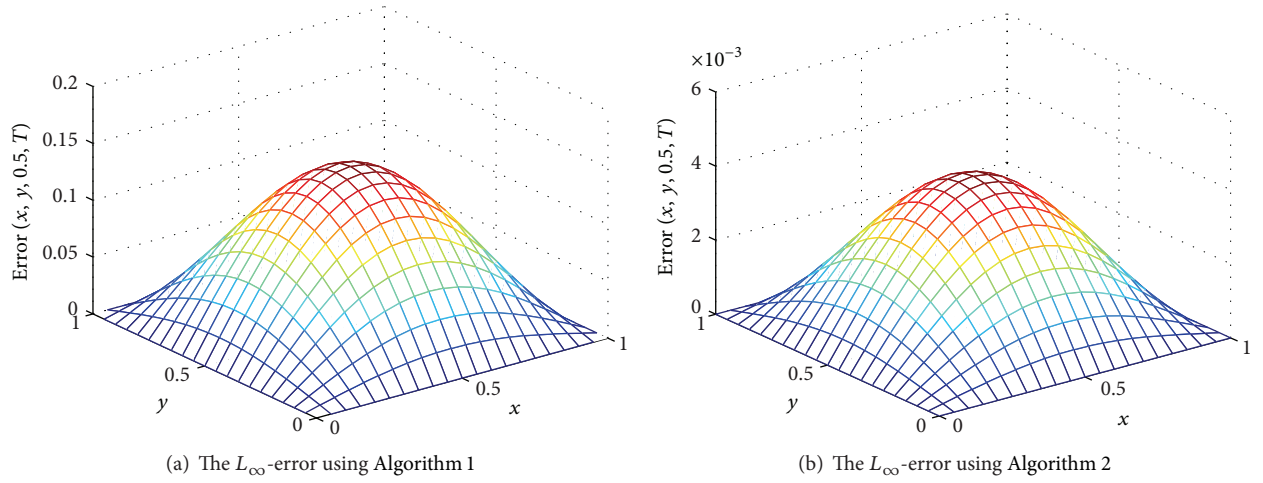


FIGURE 1: The plot of the  $L^\infty$ -error using both algorithms at  $T = 1$  and  $z = 0.5$  with  $N = 40$ ,  $M = 40$ , and  $\alpha = 0.5$  for Problem 5.

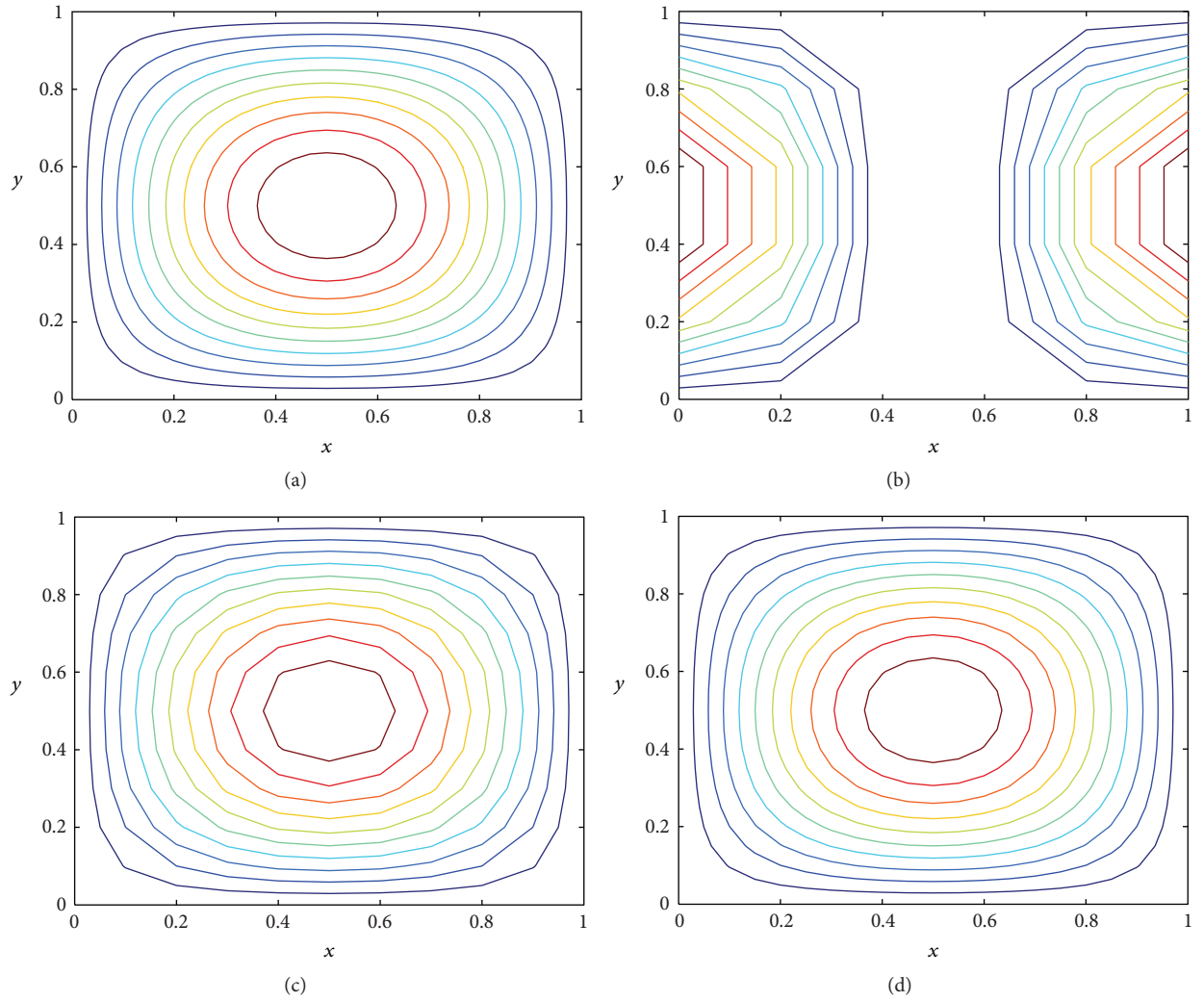


FIGURE 2: The contour plot of the exact solution and numerical solutions for different  $N$  at  $T = 0.01$  and  $z = 0.5$  with  $\alpha = 0.5$  and  $M = 100$  for Problem 5. (a) Exact solution; (b)  $N = 5$ ; (c)  $N = 10$ ; (d)  $N = 20$ .



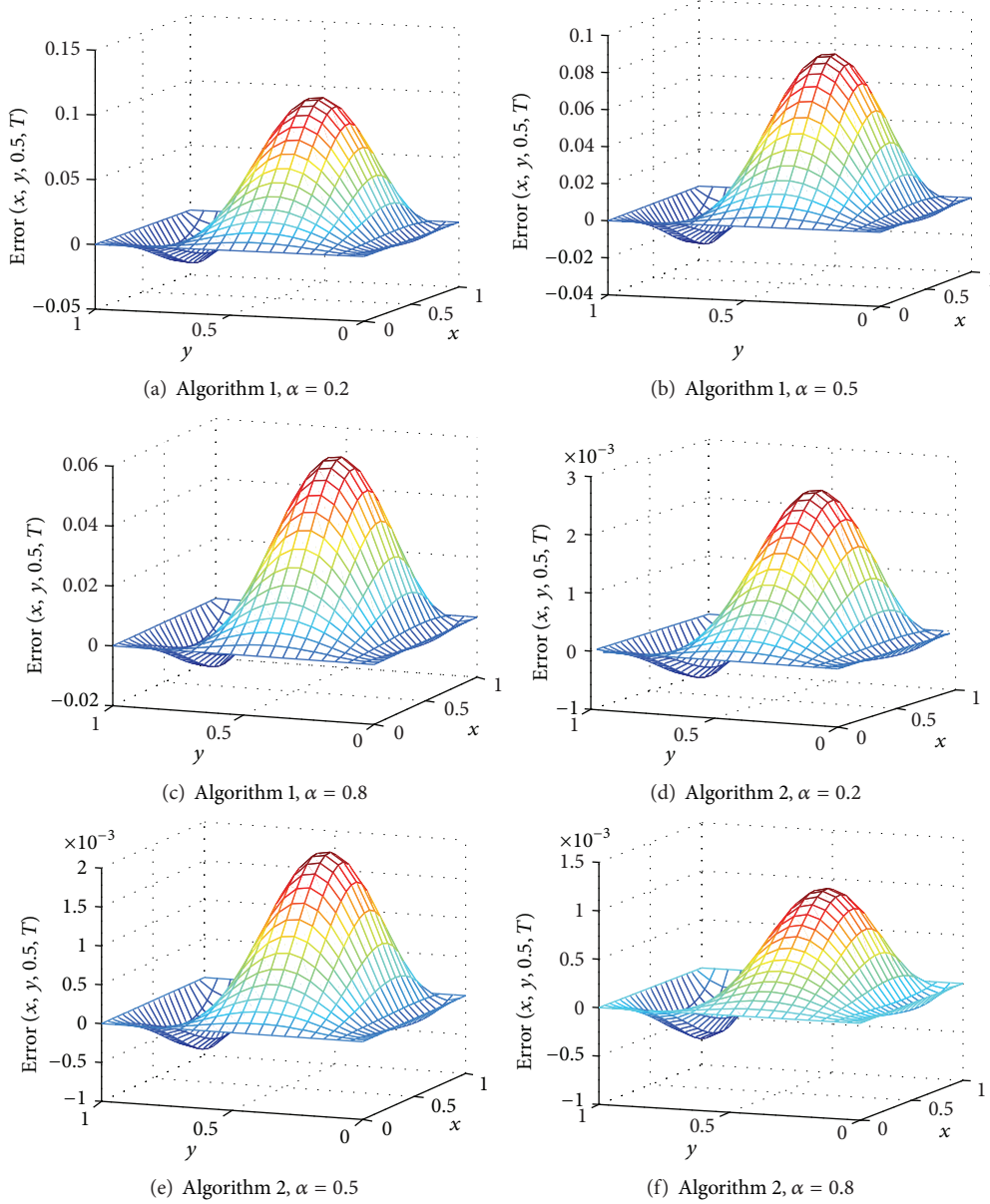


FIGURE 3: The plot of the  $L^\infty$ -error using both algorithms at  $T = 2$  and  $z = 0.5$  with  $N = 40$ ,  $M = 40$ , and different  $\alpha$  for Problem 6.

### 3. Numerical Experiments

In this section, two numerical examples are presented to demonstrate the high efficiency and accuracy of the new method. The numerical results got from Algorithm 2 will be compared with Algorithm 1. The  $L^\infty$ -norm error (denoted as  $L^\infty$ -error) and the  $L^2$ -norm error (denoted as  $L^2$ -error) of the numerical solutions are shown. Note that the treatment of space is the same for two schemes all of them can achieve the same results when  $\tau$  is small enough to eliminate the influence of the temporal approximation. So, in the following experiments, we only choose Algorithm 2 to investigate the spatial convergence rate.

Moreover, we compute the convergent order of accuracy for both schemes. Let us consider two mesh sizes  $\Delta H$  and  $\Delta h$

on  $\Omega^H$  and  $\Omega^h$ , respectively. The  $L^2$ -norm errors of these two grids are denoted as  $\|\cdot\|_0^H$  and  $\|\cdot\|_0^h$ . If we set the convergent order of accuracy to be Rate, then we have the following form:

$$\left(\frac{\Delta H}{\Delta h}\right)^{\text{Rate}} = \frac{\|\cdot\|_0^H}{\|\cdot\|_0^h}. \quad (32)$$

So, the convergent order of accuracy Rate can be computed as

$$\text{Rate} = \frac{\log \|\cdot\|_0^H / \|\cdot\|_0^h}{\log (\Delta H / \Delta h)}. \quad (33)$$

The order of accuracy is formally defined when the mesh size approaches to zero. Therefore, when the mesh size is relatively

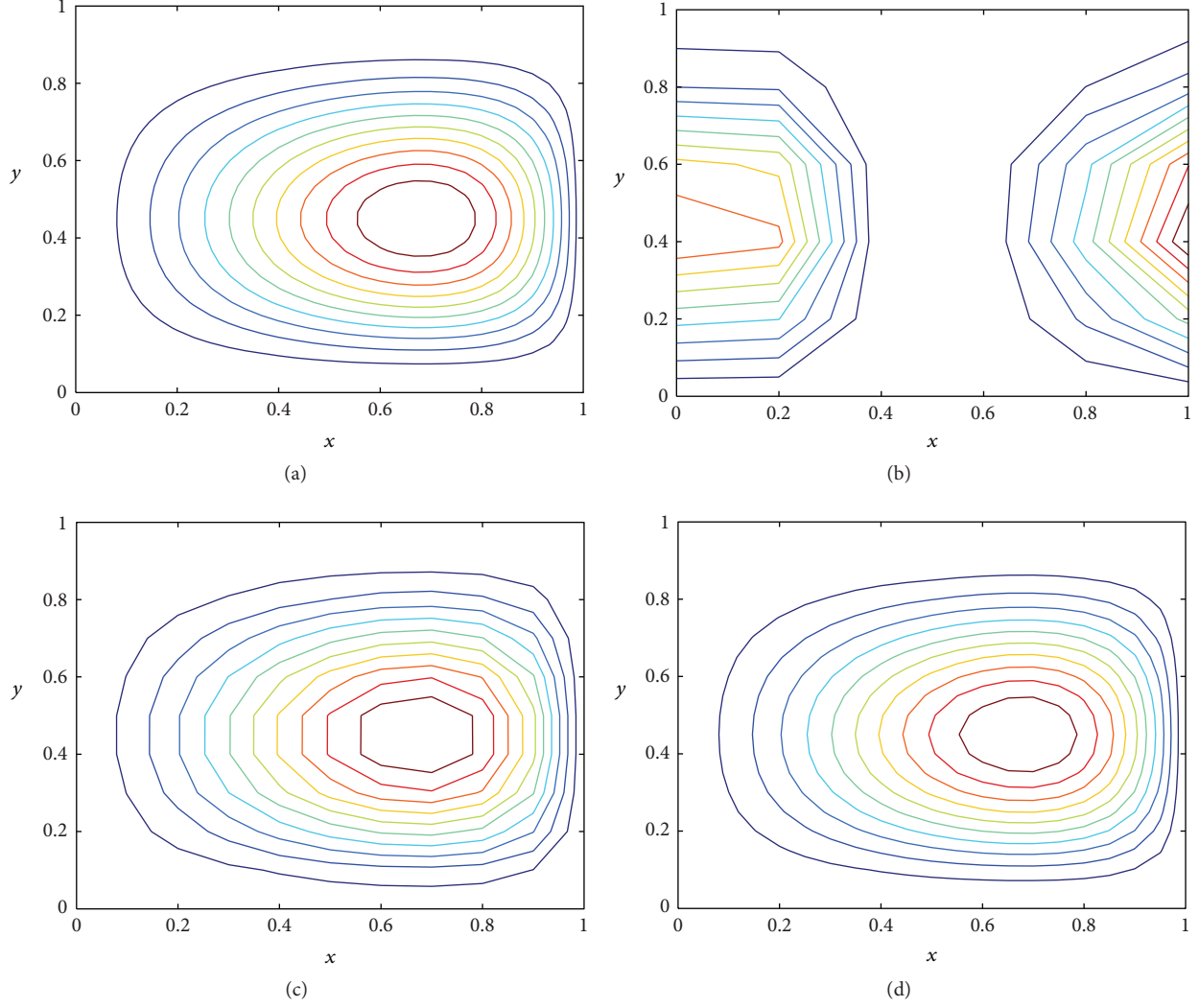


FIGURE 4: The contour plot of the exact solution and numerical solutions for different  $N$  at  $T = 0.01$  and  $z = 0.5$  with  $\alpha = 0.5$  and  $M = 100$  for Problem 6. (a) Exact solution; (b)  $N = 5$ ; (c)  $N = 10$ ; (d)  $N = 20$ .

large, the discretization scheme may not achieve its formal order of accuracy.

**Problem 5.** We firstly consider a pure diffusion problem. The exact analytical solution and the corresponding forcing term and initial condition are given by

$$\begin{aligned} \frac{\partial^\alpha u}{\partial t^\alpha} &= \Delta u + u^2 + f(x, y, z, t), \\ u(x, y, z, t) &= e^t \sin(\pi x) \sin(\pi y) \sin(\pi z), \\ u_0(x, y, z) &= \sin(\pi x) \sin(\pi y) \sin(\pi z). \end{aligned} \quad (34)$$

Since it is difficult to accurately solve  $\partial^\alpha u / \partial t^\alpha$  based on  $u$  except for  $\alpha = 0.5$ , in this problem, we only discuss the case with  $\alpha = 0.5$  and the corresponding

$$\begin{aligned} f(x, y, z, t) &= \left( \frac{t^{-\alpha}}{\Gamma(1-\alpha)} + \sqrt{\pi} e^t \operatorname{erf}(\sqrt{t}) + 3\pi^2 (t^2 + 1) \right) \\ &\quad \times \sin(\pi x) \sin(\pi y) \sin(\pi z) \\ &\quad - e^{2t} \sin^2(\pi x) \sin^2(\pi y) \sin^2(\pi z). \end{aligned} \quad (35)$$

We first investigate the time accuracy with  $T = 1$  for both algorithms. To this end,  $N = 40$  is chosen such that the errors stemming from the spatial approximation are negligible. Numerical results with  $\alpha = 0.5$  are presented in Table 1. It is clearly show that Algorithm 2 is much better than Algorithm 1. For instance, in Table 1, the maximum absolute error of the computed solution by Algorithm 2 is around  $2.2764e-3$  when  $N = 40$ ,  $M = 50$ . Whereas for Algorithm 1 with the same mesh size, the maximum absolute error is around  $9.9234e-2$ . Similar comparison can be made to reach similar conclusions.



TABLE 4:  $T = 2$ ,  $N = 20$ , and  $\alpha = 0.5$  with different  $M$  for Problem 6.

	$M$	10	20	30	40	50
Algorithm 1	$L^\infty$ -error	$5.1863e-1$	$2.5840e-1$	$1.6670e-1$	$1.2088e-1$	$9.3705e-2$
	$L^\infty$ -rate	—	1.0051	1.0810	1.1173	1.1411
	$L^2$ -error	$1.1282e-1$	$5.6080e-2$	$3.6162e-2$	$2.6216e-2$	$2.0321e-2$
	$L^2$ -rate	—	1.0084	1.0822	1.1180	1.1414
Algorithm 2	$L^\infty$ -error	$5.6623e-2$	$1.3141e-2$	$5.4618e-3$	$2.9080e-3$	$1.7745e-3$
	$L^\infty$ -rate	—	2.1073	2.1653	2.1910	2.2137
	$L^2$ -error	$1.2280e-2$	$2.8416e-3$	$1.1808e-3$	$6.2852e-4$	$3.8345e-4$
	$L^2$ -rate	—	2.1116	2.1659	2.1918	2.2146

TABLE 5:  $T = 2$ ,  $N = 20$ , and  $\alpha = 0.8$  with different  $M$  for Problem 6.

	$M$	10	20	30	40	50
Algorithm 1	$L^\infty$ -error	$4.7068e-1$	$2.1003e-1$	$1.2397e-1$	$8.3453e-2$	$6.0692e-2$
	$L^\infty$ -rate	—	1.1641	1.3002	1.3757	1.4272
	$L^2$ -error	$1.0234e-1$	$4.5581e-2$	$2.6896e-2$	$1.8105e-2$	$1.3168e-2$
	$L^2$ -rate	—	1.1669	1.3010	1.3758	1.4271
Algorithm 2	$L^\infty$ -error	$4.9685e-2$	$9.7522e-3$	$3.4660e-3$	$1.5740e-3$	$8.1067e-4$
	$L^\infty$ -rate	—	2.3490	2.5514	2.7440	2.9734
	$L^2$ -error	$1.0752e-2$	$2.1064e-3$	$7.5044e-4$	$3.4373e-4$	$1.8083e-4$
	$L^2$ -rate	—	2.3517	2.5455	2.7141	2.8783

TABLE 6:  $T = 0.01$  and  $M = 100$  with different  $\alpha$  and  $N$  using Algorithm 2 for Problem 6.

	$N$	5	10	20	30	40
$\alpha = 0.2$	$L^\infty$ -error	$2.7154e-4$	$1.8711e-5$	$1.1750e-6$	$2.2446e-7$	$6.4536e-8$
	$L^\infty$ -rate	—	3.8592	3.9932	4.0825	4.3329
	$L^2$ -error	$6.3987e-5$	$4.5423e-6$	$3.0190e-7$	$5.9489e-8$	$1.7738e-8$
	$L^2$ -rate	—	3.8163	3.9113	4.0060	4.2064
$\alpha = 0.5$	$L^\infty$ -error	$2.4981e-4$	$1.7177e-5$	$1.0951e-6$	$2.2343e-7$	$7.2771e-8$
	$L^\infty$ -rate	—	3.8623	3.9713	3.9202	3.8994
	$L^2$ -error	$5.8471e-5$	$4.1544e-6$	$2.7991e-7$	$5.8491e-8$	$1.9056e-8$
	$L^2$ -rate	—	3.8150	3.8916	3.8612	3.8983
$\alpha = 0.8$	$L^\infty$ -error	$2.1863e-4$	$1.5586e-5$	$1.0322e-6$	$2.1189e-7$	$6.8378e-8$
	$L^\infty$ -rate	—	3.8102	3.9165	3.9051	3.9315
	$L^2$ -error	$5.2327e-5$	$3.7485e-6$	$2.4905e-7$	$5.0023e-8$	$1.6021e-8$
	$L^2$ -rate	—	3.8023	3.9118	3.9588	3.9578

Meanwhile, we can see that the convergence rate for Algorithm 1 is around 1.5, which agrees with the theoretical estimates  $\min(1 + \alpha, 2 - \alpha)$ . However, for Algorithm 2, the  $L^2$ -rate is approximately 2.6, which is superconvergent. For better comparison of two algorithms, we plot the  $L^\infty$  errors (as shown in Figure 1) at  $T = 1$  and  $z = 0.5$  with  $N = 40$  and  $M = 40$  for  $\alpha = 0.5$ . Comparing both subplots, we can see that Algorithm 2 has substantially higher accuracy than Algorithm 1.

For the spatial convergence rate, we choose  $T = 0.01$  and  $M = 100$  to eliminate the influence of the temporal approximation. Numerical results using Algorithm 2 are presented in Table 2. The convergence order is indeed 4.

Figure 2 shows the contour plot of the exact solution in the  $x$ - $y$  plane and numerical solutions using Algorithm 2 for different  $N$  at  $z = 0.5$  and  $T = 0.01$  with  $\alpha = 0.5$  and  $M = 100$ . We can see that when  $N = 20$ , the numerical solution is a good approximation of the exact solution.

**Problem 6.** In order to illustrate the effectiveness of our method, the following time-fractional convection-diffusion problem is considered:

$$\begin{aligned} \frac{\partial^\alpha u}{\partial t^\alpha} = & \Delta u - 2u_x - 4u_y - 6u_z + u \\ & + e^{-3x-6y-9z} \left( u^2 - e^{2x+4y+6z} \right)^2 + e^{x+2y+3z} f(x, y, z, t), \end{aligned} \quad (36)$$

$$u(x, y, z, t) = (t^2 + 1)x^2(1-x)^3 \sin(\pi y) \sin(\pi z) e^{x+2y+3z}, \quad (37)$$

$$u_0(x, y, z) = x^2(1-x)^3 \sin(\pi y) \sin(\pi z) e^{x+2y+3z}. \quad (38)$$

The right-hand side function  $f$  is specified to satisfy the given equation and the exact solution.

Numerical results for time with  $\alpha = 0.2, 0.5, 0.8$  are presented in Tables 3–5, respectively. Again, as expected, it is shown that Algorithm 2 has higher accuracy in comparison with Algorithm 1. For instance, in Table 5, the maximum absolute error of the computed solution by Algorithm 2 is around  $8.1067e-4$  when  $N = 20$ ,  $M = 50$ . Whereas for Algorithm 1 with the same mesh size, the maximum absolute error is around  $6.0692e-2$ .

However, we note that the convergence orders for Algorithm 1 with  $\alpha = 0.2$  and  $0.5$  are unsatisfying, as shown in Tables 3 and 4, respectively. We would like to comment on the fact that the order of accuracy is formally defined when the mesh size approaches to zero. Therefore, when the mesh size is relatively large, the discretization scheme may not achieve its formal order of accuracy. Moreover, the data show that the order is increasing now; hence, it will achieve its formal order  $1 + \alpha$  with the decrease of the time step  $\tau$ .

Figure 3 shows the  $L^\infty$  errors at  $T = 2$  and  $z = 0.5$  with  $N = 50$  and  $M = 40$  for different  $\alpha$ . Again, as expected, it indicates that Algorithm 2 has substantially higher accuracy than Algorithm 1 regardless of  $\alpha$ .

Table 6 presents space results with  $T = 0.01$  and  $M = 100$  for different  $\alpha$  and  $N$ , from which we can find that Algorithm 2 is stable and convergent and it can obtain fourth-order accuracy when mesh becomes finer, regardless of the value of  $\alpha$ .

Figure 4 shows the contour plot of the exact solution in the  $x$ - $y$  plane and numerical solutions using Algorithm 2 for different  $N$  at  $z = 0.5$  and  $T = 0.01$  with  $\alpha = 0.5$  and  $M = 100$ . We can see that when  $N = 20$ , the numerical solution is a good approximation of the exact solution.

## 4. Conclusions

In this paper, we have proposed two high-order compact ADI algorithms with order  $O(\tau^{\min(1+\alpha, 2-\alpha)} + h^4)$  and  $O(\tau^{2-\alpha} + h^4)$ , respectively, for solving the 3D nonlinear time-fractional convection-diffusion equation. Since it only involves three-point stencil for each one-dimensional operator, it can be solved by applying the one-dimensional tridiagonal Thomas algorithm with a considerable saving in computing time. Numerical experiments have shown that Algorithm 2 is very significantly reduced in the splitting perturbation error and is superconvergent, which is much better than Algorithm 1.

Compared with existing deducing methods, our approach is advantageous because

- (1) the original time-fractional convection-diffusion equation is first transformed to a time-fractional diffusion equation, and the resulting time-fractional diffusion problem is then solved by ADI method,

this is novel and much more simpler than other traditional methods;

- (2) the nonlinear term is transformed into a linear term which makes the computation cost effective and the approximation solutions more accurate;
- (3) we show that the critical modification in the ADI procedures can virtually eliminate the perturbation errors and produce identical approximations of the solution of the differential problem.

## Acknowledgments

The authors would like to thank the editor and referees for their criticism, valuable comments, and suggestions which helped to improve the results of this paper. This work is in part supported by the NSF of China (nos. 11271313 and 61163027), the China Postdoctoral Science Foundation (nos. 201104702 and 2012M512056), the Key Project of Chinese Ministry of Education (no. 212197), the NSF of Xinjiang Province (no. 2013211B01), and the Excellent Doctor Innovation Program of Xinjiang University (no. XJUBSCX-2012003).

## References

- [1] R. Gorenflo and F. Mainardi, "Fractional calculus: integral and differential equations of fractional order," in *Fractals and Fractional Calculus in Continuum Mechanics*, A. Carpinteri and F. Mainardi, Eds., vol. 378, pp. 223–276, Springer, Vienna, Austria, 1997.
- [2] F. Mainardi, "Fractional relaxation-oscillation and fractional diffusion-wave phenomena," *Chaos, Solitons and Fractals*, vol. 7, no. 9, pp. 1461–1477, 1996.
- [3] I. Podlubny, *Fractional Differential Equations*, Academic Press, New York, NY, USA, 1999.
- [4] I. M. Sokolov, J. Klafter, and A. Blumen, "Fractional kinetics," *Physics Today*, vol. 55, no. 11, pp. 48–54, 2002.
- [5] C.-M. Chen, F. Liu, I. Turner, and V. Anh, "A Fourier method for the fractional diffusion equation describing sub-diffusion," *Journal of Computational Physics*, vol. 227, no. 2, pp. 886–897, 2007.
- [6] P. Zhuang, F. Liu, V. Anh, and I. Turner, "New solution and analytical techniques of the implicit numerical method for the anomalous subdiffusion equation," *SIAM Journal on Numerical Analysis*, vol. 46, no. 2, pp. 1079–1095, 2008.
- [7] C.-M. Chen, F. Liu, and K. Burrage, "Finite difference methods and a Fourier analysis for the fractional reaction-subdiffusion equation," *Applied Mathematics and Computation*, vol. 198, no. 2, pp. 754–769, 2008.
- [8] R. Scherer, S. L. Kalla, L. Boyadjeiev, and B. Al-Saqabi, "Numerical treatment of fractional heat equations," *Applied Numerical Mathematics*, vol. 58, no. 8, pp. 1212–1223, 2008.
- [9] F. Liu, C. Yang, and K. Burrage, "Numerical method and analytic technique of the modified anomalous subdiffusion equation with a nonlinear source term," *Journal of Computational and Applied Mathematics*, vol. 231, no. 1, pp. 160–176, 2009.
- [10] X. Zhao and Z.-z. Sun, "A box-type scheme for fractional sub-diffusion equation with Neumann boundary conditions," *Journal of Computational Physics*, vol. 230, no. 15, pp. 6061–6074, 2011.

- [11] M. Cui, "Compact finite difference method for the fractional diffusion equation," *Journal of Computational Physics*, vol. 228, no. 20, pp. 7792–7804, 2009.
- [12] G.-H. Gao and Z.-Z. Sun, "A compact finite difference scheme for the fractional sub-diffusion equations," *Journal of Computational Physics*, vol. 230, no. 3, pp. 586–595, 2011.
- [13] J. C. Ren, Z.-Z. Sun, and X. Zhao, "Compact difference scheme for the fractional sub-diffusion equation with Neumann boundary conditions," *Journal of Computational Physics*, vol. 232, pp. 456–467, 2013.
- [14] Y. M. Lin and C. J. Xu, "Finite difference/spectral approximations for the time-fractional diffusion equation," *Journal of Computational Physics*, vol. 225, no. 2, pp. 1533–1552, 2007.
- [15] Y. J. Jiang and J. T. Ma, "High-order finite element methods for time-fractional partial differential equations," *Journal of Computational and Applied Mathematics*, vol. 235, no. 11, pp. 3285–3290, 2011.
- [16] L. L. Wei, Y. N. He, and B. Tang, "Analysis of the fractional Kawahara equation using an implicit fully discrete local discontinuous Galerkin method," *Numerical Methods for Partial Differential Equations*, 2012.
- [17] L. L. Wei, X. D. Zhang, S. Kumar, and A. Yildirim, "A numerical study based on an implicit fully discrete local discontinuous Galerkin method for the time-fractional coupled Schrödinger system," *Computers & Mathematics with Applications*, vol. 64, no. 8, pp. 2603–2615, 2012.
- [18] L. L. Wei, Y. N. He, and Y. Zhang, "Numerical analysis of the fractional seventh-order KdV equation using an implicit fully discrete local discontinuous Galerkin method," *International Journal of Numerical Analysis & Modeling*, vol. 10, pp. 430–444, 2013.
- [19] P. Zhuang and F. Liu, "Implicit difference approximation for the time fractional diffusion equation," *Journal of Applied Mathematics & Computing*, vol. 22, no. 3, pp. 87–99, 2006.
- [20] S. B. Yuste and L. Acedo, "An explicit finite difference method and a new von Neumann-type stability analysis for fractional diffusion equations," *SIAM Journal on Numerical Analysis*, vol. 42, no. 5, pp. 1862–1874, 2005.
- [21] C.-M. Chen, F. Liu, I. Turner, and V. Anh, "Numerical schemes and multivariate extrapolation of a two-dimensional anomalous sub-diffusion equation," *Numerical Algorithms*, vol. 54, no. 1, pp. 1–21, 2010.
- [22] C.-M. Chen, F. Liu, V. Anh, and I. Turner, "Numerical methods for solving a two-dimensional variable-order anomalous subdiffusion equation," *Mathematics of Computation*, vol. 81, no. 277, pp. 345–366, 2012.
- [23] X. D. Zhang, J. Liu, L. L. Wei, and C. X. Ma, "Finite element method for Grunwald-Letnikov time-fractional partial differential equation," *Journal of Applied Analysis*, vol. 1, pp. 1–12, 2012.
- [24] L. L. Wei, X. D. Zhang, and Y. N. He, "Analysis of a local discontinuous Galerkin method for time-fractional advection-diffusion equations," *International Journal of Numerical Methods for Heat & Fluid Flow*, vol. 23, no. 4, pp. 634–648, 2013.
- [25] X. Li and C. Xu, "A space-time spectral method for the time fractional diffusion equation," *SIAM Journal on Numerical Analysis*, vol. 47, no. 3, pp. 2108–2131, 2009.
- [26] Y.-N. Zhang and Z.-Z. Sun, "Alternating direction implicit schemes for the two-dimensional fractional sub-diffusion equation," *Journal of Computational Physics*, vol. 230, no. 24, pp. 8713–8728, 2011.
- [27] M. R. Cui, "Compact alternating direction implicit method for two-dimensional time fractional diffusion equation," *Journal of Computational Physics*, vol. 231, no. 6, pp. 2621–2633, 2012.
- [28] M. M. Meerschaert and C. Tadjeran, "Finite difference approximations for fractional advection-dispersion flow equations," *Journal of Computational and Applied Mathematics*, vol. 172, no. 1, pp. 65–77, 2004.
- [29] M. M. Meerschaert, H.-P. Scheffler, and C. Tadjeran, "Finite difference methods for two-dimensional fractional dispersion equation," *Journal of Computational Physics*, vol. 211, no. 1, pp. 249–261, 2006.
- [30] C. Tadjeran and M. M. Meerschaert, "A second-order accurate numerical method for the two-dimensional fractional diffusion equation," *Journal of Computational Physics*, vol. 220, no. 2, pp. 813–823, 2007.
- [31] S. Chen, F. Liu, I. Turner, and V. Anh, "An implicit numerical method for the two-dimensional fractional percolation equation," *Applied Mathematics and Computation*, vol. 219, no. 9, pp. 4322–4331, 2013.
- [32] V. J. Ervin, N. Heuer, and J. P. Roop, "Numerical approximation of a time dependent, nonlinear, space-fractional diffusion equation," *SIAM Journal on Numerical Analysis*, vol. 45, no. 2, pp. 572–591, 2007.
- [33] H. Zhang, F. Liu, and V. Anh, "Galerkin finite element approximation of symmetric space-fractional partial differential equations," *Applied Mathematics and Computation*, vol. 217, no. 6, pp. 2534–2545, 2010.
- [34] Y. Y. Zheng, C. P. Li, and Z. G. Zhao, "A note on the finite element method for the space-fractional advection diffusion equation," *Computers & Mathematics with Applications*, vol. 59, no. 5, pp. 1718–1726, 2010.
- [35] W. Y. Liao, "A compact high-order finite difference method for unsteady convection-diffusion equation," *International Journal for Computational Methods in Engineering Science and Mechanics*, vol. 13, no. 3, pp. 135–145, 2012.
- [36] J. Douglas Jr. and S. Kim, "Improved accuracy for locally one-dimensional methods for parabolic equations," *Mathematical Models & Methods in Applied Sciences*, vol. 11, no. 9, pp. 1563–1579, 2001.



