Advances in Difference Equations
a SpringerOpen Journal

RESEARCH Open Access

A new parallel difference algorithm based on improved alternating segment Crank–Nicolson scheme for time fractional reaction–diffusion equation

Xiaozhong Yang[1*] and Xu Dang[1]

[*]Correspondence:
yxiaozh@ncepu.edu.cn
[1]School of Mathematics and Physics, North China Electric Power University, Beijing, China

Abstract

The fractional reaction–diffusion equation has profound physical and engineering background, and its rapid solution research is of important scientific significance and engineering application value. In this paper, we propose a parallel computing method of mixed difference scheme for time fractional reaction–diffusion equation and construct a class of improved alternating segment Crank–Nicolson (IASC–N) difference schemes. The class of parallel difference schemes constructed in this paper, based on the classical Crank–Nicolson (C–N) scheme and classical explicit and implicit schemes, combines with alternating segment techniques. We illustrate the unique existence, unconditional stability, and convergence of the parallel difference scheme solution theoretically. Numerical experiments verify the theoretical analysis, which shows that the IASC–N scheme has second order spatial accuracy and $2-\alpha$ order temporal accuracy, and the computational efficiency is greatly improved compared with the implicit scheme and C–N scheme. The IASC–N scheme has ideal computation accuracy and obvious parallel computing properties, showing that the IASC–N parallel difference method is effective for solving time fractional reaction–diffusion equation.

MSC: 65M06; 65M12; 65Y05

Keywords: Fractional reaction–diffusion equation; IASC–N difference scheme; Unconditional stability; Order of convergence; Parallel computing

1 Introduction

The fractional reaction–diffusion equation has a profound physical background and rich theoretical connotation. As the application of fractional calculus increases, the solution of fractional evolution equation has become an urgent research work (Baleanu et al. 2018; Mohammadi et al. 2018; Hajipour et al. 2019; Baleanu et al. 2019) [1–4]. The analytical solution of fractional reaction–diffusion equation is difficult to give explicitly. Even the analytical solution of linear fractional reaction–diffusion equation mostly contains special functions, such as Mittag-Leffler functions (Kumar et al. 2018; Kumar et al. 2018; Singh et al. 2019) [5–7]. The series corresponding to these functions converge slowly, and the calculation of these special functions is quite difficult in practical applications. Therefore, all mentioned above make the high-efficiency numerical simulation of fractional reaction–diffusion equation an urgent research problem (Uchaikin 2013; Chen et al. 2010) [8, 9]. Since fractional calculus has historical dependence and global correlation, the amount of computation and storage of numerical simulation for fractional differential equation is extremely large. Even with high-performance computers, it is difficult to simulate in long-term history (the amount of computation increases exponentially with the increase of time) or large computational domain (Guo et al. 2015; Sabatier et al. 2014) [10, 11]. Starting from the urgent need of scientific engineering computing in the era of big data, the finite difference parallel computing method for fractional reaction–diffusion equation has important scientific significance and engineering application value.

For the numerical computation of fractional evolution equation, more and more numerical computation methods are proposed (Singh et al. 2018; Kumar et al. 2018; Goswami et al. 2019) [12–14]. At present, the relatively more mature methods are still the finite difference method and the series method (mainly Adomian decomposition and variational iterative method). Among them, the finite difference method has a wide application range. The advantages of the finite difference method are reflected in solving problems in small spatial domain and short time history. The accuracy and stability of the algorithm can meet the needs of numerical simulation of small-scale problems (Sun and Gao 2015; Liu et al. 2015) [15, 16]. The theoretical analysis methods mainly include Fourier method, energy estimation, matrix method (eigenvalue), mathematical induction, and some other numerical methods, but most of them cannot be used as universal numerical methods or lack a perfect theoretical analysis. For time fractional diffusion equation, Lin and Xu (2007) [17] constructed a finite difference scheme in time domain and a Legendre spectrum method in spatial domain and proved the unconditional stability and convergence of the method. For time fractional fourth order reaction–diffusion equation with nonlinear reaction term, Liu et al. (2015) [18] proposed a finite difference approximation in time direction and a finite element approximation in spatial direction to obtain the numerical solution of the time fractional fourth order reaction–diffusion equation and analyzed the unconditional stability of the method. For time fractional reaction–diffusion equation, Liu et al. (2015) [19] proposed an H^1-Galerkin mixed finite element method and obtained the numerical results with optimal time and spatial convergence order. Chen et al. (2016) [20] discussed the numerical solution of distribution order time fractional reaction–diffusion equation in semi-infinite spatial domain and proposed a fully discrete scheme based on finite difference method in time domain and spectral approximation using Laguerre function in spatial domain. The numerical experiments verified the effectiveness of the proposed scheme. Zhang and Yang (2018) [21] gave a class of explicit–implicit (E–I) and implicit–explicit (I–E) difference methods for time fractional reaction–diffusion equation. The method had second order spatial precision and $2 - \alpha$ order time precision, and its computation time was nearly 41% lower than that of the classical implicit difference scheme.

With the rapid development of multi-core and clustering technologies, parallel algorithms have become one of the mainstream technologies for improving numerical computational efficiency (Bjorstad and Luskin 2000; Chi et al. 2015) [22, 23]. For integer diffusion equation, Evans and Abdullah (1983) [24] proposed the idea of group explicit (GE) and designed the alternating group explicit (AGE) scheme, which not only ensured the stability of numerical computation, but also had good parallel performance. Implicit scheme generally has good stability but is not suitable for parallelization. Inspired by the method of constructing AGE scheme, Zhang et al. (1994) [25] proposed the idea of constructing segment implicit schemes using Saul'yev asymmetric scheme, and appropriately used alternating techniques to establish a variety of explicit–implicit and pure implicit alternating parallel methods, which obtained research results with both stability and parallelism. Academician Zhou (1997) [26] called the mixed explicit and implicit schemes for the most general parabolic equation as a difference scheme with intrinsic parallelism. He studied the theoretical issues such as the existence, uniqueness, convergence, and stability of the difference decomposition, and established the basic theory of the difference methods with intrinsic parallelism for parabolic equation. Wang (2006) [27] constructed a class of alternating difference schemes with intrinsic parallelism for the KdV equation using Saul'yev asymmetric scheme combined with C–N difference scheme and proved the linear absolute stability of the scheme. Yuan et al. (2007) [28] proposed a class of parallel difference schemes with unconditional stability and second order spatial precision for nonlinear parabolic equation. The main advantage of these difference schemes with intrinsic parallelism is that the schemes can be directly applied to parallel computer systems with distributed memory and minimize the amounts of communication between processors. The algorithm only needs to transfer local messages between adjacent processors. The communication and computation involved are local so that it is easier to load balance between them, thus obtaining good precision and scalability of parallel computing.

In recent years, some progress has been made in the fast algorithm for fractional partial differential equation. Most of fast algorithms are parallel algorithms for studying algebraic equations from the perspective of numerical algebra. Wang et al. (2010) [29] gave a differential scheme $O(N\log^2 N)$ fast algorithm for the one-dimensional space fractional diffusion equation based on the special structure of the difference matrix in the constructed scheme, and further extended the fast algorithm to solve two-dimensional space fractional diffusion equation, which was an early attempt to apply a fast algorithm to numerical simulation of fractional differential equations. Diethelm (2011) [30] performed parallel computation on the second order Adams–Bashforth–Moulton method of fractional derivatives, and discussed the accuracy of the parallel algorithm. Wang et al. (2012) [31] constructed an improved conjugate gradient squared (CGS) method by decomposing a two-dimensional spatial fractional difference matrix into a class-like Toeplitz matrix. Moroney et al. (2013) [32] constructed a Newton–Krylov fast iterative method with Poisson preprocessing operator to solve nonlinear spatial fractional diffusion equations. Gong et al. (2013) [33] implemented parallel computation for the explicit difference scheme of the Riesz spatial fractional diffusion equation. The core content of parallelization was parallel computing of matrix and vector product, vector and vector addition. Sweilam et al. (2014) [34] constructed a class of parallel Crank–Nicolson difference schemes for time fractional parabolic equations. The core of the method is to solve the discrete algebraic equations in parallel by using preconditioned conjugate gradient method. Lu et al. (2015) [35] established a differential scheme for time fractional sub-diffusion equations and proposed a fast algorithm based on its special structure. Wang et al. (2016) [36] studied the parallel algorithm of the implicit difference scheme for Caputo fractional reaction–diffusion equation. After parallelization, the computational efficiency of the original scheme is improved. The purpose is to parallelize the serial algorithm under the principle of minimizing communication, properly allocating computational tasks, and trying not to change the original serial difference scheme. Wu et al. (2018) [37] proposed an alternating segment Crank–Nicolson parallel difference scheme for time fractional sub-diffusion equation, which had ideal computing accuracy and efficiency. For nonlinear time-space fractional parabolic partial differential equations, Biala and Khaliq (2018) [38] developed a time stepping scheme which was implemented in parallel using the distributed (MPI), shared memory systems (OpenMP), and a combination of both, and the scheme was shown to be convergent and of order $1+\alpha$. Fu and Wang (2019) [39] developed a fast parallel finite difference method for space-time fractional partial differential equations, which used a matrix-free preconditioned fast Krylov subspace iterative solver at each time step, significantly reducing computational complexity and memory requirement.

We do not study parallel algorithms from the perspective of numerical algebra, but based on the parallelization of traditional differential schemes, we seek to explore another way of parallelization skipping the difficulty of numerical algebra. A class of mixed difference parallel computing methods for solving fractional reaction–diffusion equation is proposed in this paper. Based on the classical C–N scheme and classical explicit and implicit schemes, our proposed method combines alternating segment techniques to construct an improved alternating segment Crank–Nicolson (IASC–N) difference scheme. We analyze the existence, uniqueness, unconditional stability, and convergence of the IASC–N scheme solution theoretically, and numerical experiments verify the theoretical analysis. The computational efficiency of IASC–N scheme is greatly improved compared to the implicit scheme and the C–N scheme. The ideal computational accuracy and obvious parallel computing properties indicate that the IASC–N parallel difference scheme is effective for solving time fractional reaction–diffusion equation.

The structure of this paper is as follows. In Sect. 2, the time fractional reaction–diffusion equation is given and the IASC–N parallel difference scheme is constructed. In Sect. 3, the unique solvability, stability, and convergence of the method are proved rigorously. In Sect. 4, the specific numerical example is given which verifies the efficiency of the constructed scheme and supports theoretical results.

2 Alternating segment C–N parallel difference scheme for time fractional reaction–diffusion equation

2.1 Time fractional reaction–diffusion equation

Consider the time fractional reaction–diffusion equation defined in region $\Omega = \{0 \le x \le L, 0 \le t \le T\}$ (Uchaikin 2013; Chen et al. 2010; Guo et al. 2015) [8–10]:

$$\begin{cases} \frac{\partial^\alpha u(x,t)}{\partial t^\alpha} = \frac{\partial^2 u(x,t)}{\partial x^2} - pu(x,t) + f(x,t), \\ u(x,0) = u_0(x), \\ u(0,t) = u(L,t) = 0, \end{cases} \tag{1}$$

where $f(x,t)$ and $u_0(x)$ are given functions, p is a nonnegative constant, $0 < \alpha < 1$, and $\frac{\partial^\alpha u(x,t)}{\partial t^\alpha}$ is a Caputo-type fractional derivative defined by

$${}_0^C D_t^\alpha u(x,t) = \frac{\partial^\alpha u(x,t)}{\partial t^\alpha} = \frac{1}{\Gamma(1-\alpha)} \int_0^t \frac{\partial u(x,\xi)}{\partial \xi} \frac{d\xi}{(t-\xi)^\alpha}.$$

When $f(x,t) = 0$, the analytical solution of Eq. (1) in region $\Omega = \{0 \le x \le L, 0 \le t \le T\}$ can be obtained by finite sine and Laplace transforms as follows (Guo et al. 2015; Sabatier et al. 2014) [10, 11]:

$$u(x,t) = \frac{2}{L}\sum_{n=1}^{\infty} E_\alpha\bigl[-\bigl(p + a^2n^2t^\alpha\bigr)\bigr]\sin(anx)\int_0^L u_0(r)\sin(anr)\,\mathrm{d}r,$$

where $E_\alpha(z)$ is a Mittag-Leffler function, $E_\alpha(z) = \sum_{k=0}^{\infty}\frac{z^k}{\Gamma(\alpha k+1)}$, and $a = \frac{\pi}{L}$.

2.2 A class of IASC–N parallel difference schemes

With the space step h and the time step τ, the solution region Ω is divided into grids and the grid points are (x_i, t_k). Let space step $h = \frac{L}{M}$ and time step $\tau = \frac{T}{N}$, then $x_i = ih$, $i = 0,1,2,\ldots,M$, and $t_k = k\tau$, $k = 0,1,2,\ldots,N$. Let U_i^k be the numerical solution to $u(x_i,t_k)$ and f_i^k be the exact solution to $f(x_i,t_k)$.

The time fractional derivative of Eq. (1) can be discretized into the following form:

$$\begin{aligned}
\frac{\partial^\alpha u(x_i,t_{k+1})}{\partial t^\alpha} &= \frac{1}{\Gamma(1-\alpha)}\sum_{j=0}^{k}\int_{j\tau}^{(j+1)\tau}\frac{\partial u(x_i,\xi)}{\partial \xi}\frac{\mathrm{d}\xi}{(t_{k+1}-\xi)^\alpha} \\
&\approx \frac{1}{\Gamma(1-\alpha)}\sum_{j=0}^{k}\frac{u(x_i,t_{j+1}) - u(x_i,t_j)}{\tau}\int_{j\tau}^{(j+1)\tau}\frac{\mathrm{d}\xi}{(t_{n+1}-\xi)^\alpha} \\
&= \frac{\tau^{1-\alpha}}{\Gamma(2-\alpha)}\sum_{j=0}^{k}\frac{(j+i)^{1-\alpha} - j^{1-\alpha}}{\tau}\bigl[u(x_i,t_{k+1-j}) - u(x_i,t_{k-j})\bigr] \\
&= \frac{\tau^{-\alpha}}{\Gamma(2-\alpha)}\sum_{j=0}^{k} l_j\bigl[u(x_i,t_{k+1-j}) - u(x_i,t_{k-j})\bigr],
\end{aligned}$$

where $l_j = (j+1)^{(1-\alpha)} - j^{(1-\alpha)}$, $j = 0,1,2,\ldots,N$. Then there is

$$ {}_0^C D_t^\alpha u(x_i,t_{k+1}) = \frac{\tau^{-\alpha}}{\Gamma(2-\alpha)}\sum_{j=0}^{k} l_j\bigl[u(x_i,t_{k+1-j}) - u(x_i,t_{k-j})\bigr]. \tag{2}$$

To construct the IASC–N scheme of Eq. (1), give the following three discrete schemes first, where $c = \tau^\alpha\Gamma(2-\alpha)$, $a = \frac{c}{h^2}$, and $b = pc$.

(1) Classical explicit difference scheme

$$ {}_0^C D_t^\alpha u(x_i,t_{k+1}) = \frac{1}{h^2}\bigl(U_{i-1}^k - 2U_i^k + U_{i+1}^k\bigr) - pU_i^k + f_i^k.$$

In conjunction with Eq. (2), the above scheme can be rewritten as

$$\sum_{j=0}^{k} l_j\bigl(U_i^{k+1-j} - U_i^{k-j}\bigr) = a\bigl(U_{i-1}^k - 2U_i^k + U_{i+1}^k\bigr) - bU_i^k + cf_i^k. \tag{3}$$

(2) Classical implicit difference scheme

$$ {}_0^C D_t^\alpha u(x_i,t_{k+1}) = \frac{1}{h^2}\bigl(U_{i-1}^{k+1} - 2U_i^{k+1} + U_{i+1}^{k+1}\bigr) - pU_i^{k+1} + f_i^{k+1}.$$

In conjunction with Eq. (2), the above scheme can be rewritten as

$$\sum_{j=0}^{k} l_j\left(U_i^{k+1-j}-U_i^{k-j}\right)=a\left(U_{i-1}^{k+1}-2U_i^{k+1}+U_{i+1}^{k+1}\right)-bU_i^{k+1}+cf_i^{k+1}. \tag{4}$$

(3) C–N difference scheme

$$\begin{aligned}{}_0^C D_t^\alpha u(x_i,t_{k+1})={}&\frac{1}{2h^2}\left(U_{i-1}^{k+1}-2U_i^{k+1}+U_{i+1}^{k+1}+U_{i-1}^{k}-2U_i^{k}+U_{i+1}^{k}\right)\\&-\frac{p}{2}\left(U_i^{k+1}+U_i^k\right)+\frac{1}{2}\left(f_i^{k+1}+f_i^k\right).\end{aligned}$$

In conjunction with Eq. (2), the above scheme can be rewritten as

$$\begin{aligned}\sum_{j=0}^{k} l_j\left(U_i^{k+1-j}-U_i^{k-j}\right)={}&\frac{a}{2}\left(U_{i-1}^{k+1}-2U_i^{k+1}+U_{i+1}^{k+1}+U_{i-1}^{k}-2U_i^{k}+U_{i+1}^{k}\right)\\&-\frac{b}{2}\left(U_i^{k+1}+U_i^k\right)+\frac{c}{2}\left(f_i^{k+1}+f_i^k\right).\end{aligned} \tag{5}$$

Let $c_j = l_{j-1} - l_j$, $j = 1, 2, \ldots, N$.
When $k = 0$, we have

$$\begin{aligned}&U_i^1=aU_{i+1}^0+(1-2a-b)U_i^0+aU_{i-1}^0+cf_i^0,\\&-aU_{i-1}^1+(1+2a+b)U_i^1-aU_{i+1}^1=U_i^0+cf_i^1,\\&-\frac{a}{2}U_{i-1}^1+\left(1+a+\frac{b}{2}\right)U_i^1-\frac{a}{2}U_{i+1}^1=\frac{a}{2}U_{i-1}^0+\left(1-a-\frac{b}{2}\right)U_i^0+\frac{a}{2}U_{i+1}^0+\frac{c}{2}\left(f_i^1+f_i^0\right).\end{aligned}$$

When $k > 0$, we have

$$U_i^{k+1}=aU_{i+1}^k+(c_1-2a-b)U_i^k+aU_{i-1}^k+\sum_{j=1}^{k-1}c_{j+1}U_i^{k-j}+l_kU_i^0+cf_i^k, \tag{6}$$

$$-aU_{i-1}^{k+1}+(1+2a+b)U_i^{k+1}-aU_{i+1}^{k+1}=c_1U_i^k+\sum_{j=1}^{k-1}c_{j+1}U_i^{k-j}+l_kU_i^0+cf_i^{k+1}, \tag{7}$$

$$\begin{aligned}-\frac{a}{2}U_{i-1}^{k+1}+\left(1+a+\frac{b}{2}\right)U_i^{k+1}-\frac{a}{2}U_{i+1}^{k+1}={}&\frac{a}{2}U_{i-1}^k+\left(c_1-a-\frac{b}{2}\right)U_i^k+\frac{a}{2}U_{i+1}^k\\&+\sum_{j=1}^{k-1}c_{j+1}U_i^{k-j}+l_kU_i^0+\frac{c}{2}\left(f_i^{k+1}+f_i^k\right).\end{aligned} \tag{8}$$

Combine the three schemes (6), (7), (8) to construct IASC–N scheme as follows.

There are M intervals on each time numerical layer, and the points on the same time numerical layer are divided into B segments. Let B be an odd number (no loss of generality) and each segment have l points, then $M-1=Bl$, where B and l are positive integers, $B \geq 3$ and $l \geq 3$. For points on odd time layers, when $i_0 = l, 2l, \ldots, (B-1)l$, calculate the inner boundary points (x_{i_0}, t_{k+1}) by the classical explicit scheme (6); when $i_0 = l+1, 2l+1, \ldots, (B-1)l+1$, calculate the inner boundary points (x_{i_0}, t_{k+1}) by the classical implicit scheme (7);

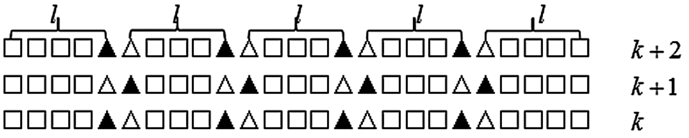

Figure 1 Schematic diagram of the IASC–N scheme ($B = 5$)

and calculate the inner points by the C–N scheme (8). For points on even time layers, when $i_0 = l, 2l, \dots, (B-1)l$, calculate the inner boundary points (x_{i_0}, t_{k+2}) by the classical implicit scheme (7); when $i_0 = l+1, 2l+1, \dots, (B-1)l+1$, calculate the inner boundary points (x_{i_0}, t_{k+2}) by the classical explicit scheme (6); and calculate the inner points by the C–N scheme (8). Particularly, use the C–N scheme at the boundary of both ends of the time numerical layers.

Let $M = 26$, $B = 5$, and $l = 5$. The construction principle of IASC–N scheme is shown in Fig. 1, where the classical explicit scheme is used in △ place, the classical implicit scheme is used in ▲ place, and the C–N scheme is used in □ place.

Above all, the IASC–N scheme of Eq. (1) can be constructed as follows:

$$\begin{cases} (I + A_1 G)U^1 = (I - A_2 G)U^0 + b^0 + cA_1 f^1 + cA_2 f^0, \\ (I + A_2 G)U^{k+1} = (c_1 I - A_1 G)U^k + b^k + \sum_{j=1}^{k-1} c_{j+1} U^{k-j} + l_k U^0 \\ \qquad\qquad\qquad + cA_2 f^{k+1} + cA_1 f^k, \\ (I + A_1 G)U^{k+2} = (c_1 I - A_2 G)U^{k+1} + b^{k+1} + \sum_{j=1}^{k} c_{j+1} U^{k-j+1} + l_{k+1} U^0 \\ \qquad\qquad\qquad + cA_1 f^{k+2} + cA_2 f^{k+1}, \end{cases}$$

$$k = 1, 3, \dots, \tag{9}$$

where

$$G = \begin{bmatrix} 2a+b & -a & & & \\ -a & 2a+b & -a & & \\ & & \ddots & & \\ & & -a & 2a+b & -a \\ & & & -a & 2a+b \end{bmatrix}_{(M-1)\times(M-1)},$$

$$\theta_i = \begin{cases} 0, & i = l, 2l, \dots, (B-1)l, \\ 1, & i = l+1, 2l+1, \dots, (B-1)l+1, \\ \frac{1}{2}, & \text{elsewhere}, \end{cases}$$

$U^k = (U_1^k, U_2^k, \dots, U_{M-1}^k)^T$, $b^k = (\frac{a}{2}(U_0^{k+1} + U_0^k), 0, \dots, 0, \frac{a}{2}(U_M^{k+1} + U_M^k))^T$, $f^k = (f_1^k, f_2^k, \dots, f_{M-1}^k)^T$, $k = 0, 1, 2, \dots, N$, $A_1 = \operatorname{diag}(\theta_1, \theta_2, \dots, \theta_{M-2}, \theta_{M-1})$, $A_1 + A_2 = I$, and I is an $(M-1)$th order identity matrix.

3 Numerical analysis of IASC–N scheme for time fractional reaction–diffusion equation

3.1 Existence and uniqueness of IASC–N scheme solution

Since

$$A_1G = \begin{bmatrix} \theta_1(2a+b) & -\theta_1 a & & & \\ -\theta_2 a & \theta_2(2a+b) & -\theta_2 a & & \\ & & \ddots & & \\ & & -\theta_{M-2}a & \theta_{M-2}(2a+b) & -\theta_{M-2}a \\ & & & -\theta_{M-1}a & \theta_{M-1}(2a+b) \end{bmatrix}_{(M-1)\times(M-1)}$$

is a weak diagonally dominant matrix, $I + A_1G$ is a strict diagonally dominant matrix. Similarly, $I + A_2G$ is also a strict diagonally dominant matrix. We have that the coefficient matrices $I + A_1G$ and $I + A_2G$ of IASC–N scheme (9) are non-singular matrices. So there is the following theorem.

Theorem 1 *The IASC–N parallel difference scheme* (9) *for time fractional reaction–diffusion equation has a unique solution.*

3.2 Unconditional stability of IASC–N scheme

Lemma 1 (Kellogg lemma [27]) *If matrix* $A + A^T$ *is nonnegative, then for* $\forall\theta > 0$, *there is* $\|(\theta I - A)(\theta I + A)^{-1}\|_2 \le 1$.

Lemma 2 $A_1G + (A_1G)^T$ *and* $A_2G + (A_2G)^T$ *defined in the IASC–N scheme* (9) *are nonnegative definite matrices.*

Proof Since

$$\begin{aligned} A_1G + (A_1G)^T &= A_1\bigl(G + G^T\bigr) \\ &= A_1 \begin{bmatrix} 4a+2b & -2a & & & \\ -2a & 4a+2b & -2a & & \\ & & \ddots & & \\ & & -2a & 4a+2b & -2a \\ & & & -2a & 4a+2b \end{bmatrix}_{(M-1)\times(M-1)} \end{aligned}$$

is a weak diagonally dominant matrix and the diagonal elements are nonnegative real numbers, $A_1G + (A_1G)^T$ is a nonnegative definite matrix. Similarly, $A_2G + (A_2G)^T$ is also a nonnegative definite matrix. The lemma is proved. □

Let U_i^k be the exact solution and $\widetilde{U}_i^k$ be the approximate solution of the difference scheme. Denote the error $\varepsilon_i^k = \widetilde{U}_i^k - U_i^k$ and $E^k = (\varepsilon_1^k, \varepsilon_2^k, \ldots, \varepsilon_{M-1}^k)$. Here $\varepsilon_0^k = \varepsilon_M^k = 0$ and $k = 0, 1, 2, \ldots, N$. Then scheme (9) satisfies

$$\begin{cases} (I + A_1G)E^1 = (I - A_2G)E^0, & \\ (I + A_2G)E^{k+1} = (c_1I - A_1G)E^k + c_2E^{k-1} + \cdots + c_kE^1 + l_kE^0, & k = 1, 3, 5, \ldots. \\ (I + A_1G)E^{k+2} = (c_1I - A_2G)E^{k+1} + c_2E^k + \cdots + c_{k+1}E^1 + l_{k+1}E^0, & \end{cases}$$

It is easy to get that the minimum eigenvalues of the matrices A_1G and A_2G are zero. Let λ_1 and λ_2 ($\lambda_1, \lambda_2 \geq 0$) be any eigenvalue of the matrix A_1G, there is $\|(c_1I - A_1G)(c_1I + A_1G)^{-1}\|_2 = \max\{\frac{|c_1-\lambda_1|}{c_1+\lambda_2}\} \leq 1$ by Lemma 1 and Lemma 2. Therefore, let $\lambda_2 = 0$ and λ_1 be taken as the eigenvalue of the matrix A_1G, which makes $|c_1 - \lambda_1|$ the largest, we have $\frac{|c_1-\lambda_1|}{c_1+\lambda_2} \leq 1$. Let λ_3 and λ_4 ($\lambda_3, \lambda_4 \geq 0$) be any eigenvalue of the matrix A_2G. Obtained by $\|(c_1I - A_2G)(c_1I + A_2G)^{-1}\|_2 = \max\{\frac{|c_1-\lambda_3|}{c_1+\lambda_4}\} \leq 1$, let $\lambda_4 = 0$ and λ_3 be taken as the eigenvalue of the matrix A_2G, which makes $|c_1 - \lambda_3|$ the largest, we get $\frac{|c_1-\lambda_3|}{c_1+\lambda_4} \leq 1$.

When $0 \leq \lambda_1 \leq c_1$, we have $\frac{|c_1-\lambda_1|+l_1}{1+\lambda_4} = \frac{c_1-\lambda_1+l_1}{1+\lambda_4} = 1 - \lambda_1 \leq 1$.

When $\lambda_1 > c_1$, let $\lambda_1 = r_1c_1$ ($r_1 > 1$), from $\frac{|c_1-\lambda_1|}{c_1+\lambda_2} \leq 1$, $r_1 \leq 2$ can be obtained. Therefore, there is $c_1 < \lambda_1 \leq 2c_1$, and $\frac{|c_1-\lambda_1|+l_1}{1+\lambda_4} = \frac{\lambda_1-c_1+l_1}{1+\lambda_4} \leq \frac{2c_1-c_1+l_1}{1+\lambda_4} = 1$ at this time.

Above all, $\frac{|c_1-\lambda_1|+l_1}{1+\lambda_4} \leq 1$. Similarly, we can get $\frac{|c_1-\lambda_3|+l_1}{1+\lambda_2} \leq 1$.

When $k = 1$ (k is the time layer),

$$\begin{aligned}\|E^1\|_2 &= \|(I + G_1)^{-1}(I - G_2)E^0\|_2 \\ &\leq \|(I + G_1)^{-1}\|_2\|(I - G_2)E^0\|_2 \\ &\leq \frac{|1 - \lambda_3|}{1 + \lambda_2}\|E^0\|_2 \\ &\leq \|E^0\|_2.\end{aligned}$$

When $k = 2$,

$$\begin{aligned}\|E^2\|_2 &= \|(I + G_2)^{-1}[(c_1I - G_1)E^1 + l_1E^0]\|_2 \\ &\leq \|(I + G_2)^{-1}\|_2\|(c_1I - G_1)E^1 + l_1E^0\|_2 \\ &\leq \frac{|c_1 - \lambda_1| + l_1}{1 + \lambda_4}\|E^0\|_2 \\ &\leq \|E^0\|_2.\end{aligned}$$

It is assumed that when $k \leq 2n$, $\|E^k\|_2 \leq \|E^0\|_2$ holds. Then there are

$$\begin{aligned}\|E^{2n+1}\|_2 &= \|(I + G_1)^{-1}[(c_1I - G_2)E^{2n} + c_2E^{2n-1} + \cdots + c_{2n}E^1 + l_{2n}E^0]\|_2 \\ &\leq \|(I + G_1)^{-1}\|_2\|(c_1I - G_2)E^{2n} + c_2E^{2n-1} + \cdots + c_{2n}E^1 + l_{2n}E^0\|_2 \\ &\leq \frac{|c_1 - \lambda_3| + c_2 + \cdots + c_{2n} + l_{2n}}{1 + \lambda_2}\|E^0\|_2 \\ &= \frac{|c_1 - \lambda_3| + l_1}{1 + \lambda_2}\|E^0\|_2 \\ &\leq \|E^0\|_2, \\ \|E^{2n+2}\|_2 &= \|(I + G_2)^{-1}[(c_1I - G_1)E^{2n+1} + c_2E^{2n} + \cdots + c_{2n+1}E^1 + l_{2n+1}E^0]\|_2 \\ &\leq \|(I + G_2)^{-1}\|_2\|(c_1I - G_1)E^{2n+1} + c_2E^{2n} + \cdots + c_{2n+1}E^1 + l_{2n+1}E^0\|_2 \\ &\leq \frac{|c_1 - \lambda_1| + c_2 + \cdots + c_{2n+1} + l_{2n+1}}{1 + \lambda_4}\|E^0\|_2 \\ &= \frac{|c_1 - \lambda_1| + l_1}{1 + \lambda_4}\|E^0\|_2 \\ &\leq \|E^0\|_2.\end{aligned}$$

Above all, we can get $\|E^k\|_2 \leq \|E^0\|_2$, where $k = 1, 2, \ldots, N$.

Theorem 2 *The IASC–N parallel difference scheme* (9) *for time fractional reaction–diffusion equation is unconditionally stable.*

3.3 Convergence of IASC–N scheme

It is known that ${}_0^C D_t^\alpha u(x_i, t_{k+1})$ has $2 - \alpha$ order numerical accuracy (Lin et al. 2011; Gao et al. 2014) [40, 41]. Let $C_U = \max_{0\leq t\leq t_{k+1}}\{|\frac{\partial^2 U(x_i,t)}{\partial t^2}|\}$, then we have

$$\begin{aligned}
\frac{\partial^{\alpha+1} U(x_i, t_{k+1})}{\partial t^{\alpha+1}} &= \frac{1}{\Gamma(1-\alpha)} \int_0^t \frac{\partial^2 U(x_i,\xi)}{\partial \xi^2} \frac{\mathrm{d}\xi}{(t_{k+1}-\xi)^\alpha} \\
&= \frac{1}{\Gamma(1-\alpha)} \sum_{j=0}^{k} \int_{j\tau}^{(j+1)\tau} \frac{\partial^2 U(x_i,\xi)}{\partial \xi^2} \frac{\mathrm{d}\xi}{(t_{k+1}-\xi)^\alpha} \\
&\leq \frac{1}{\Gamma(1-\alpha)} \max_{0\leq t\leq t_{k+1}} \left\{ \left| \frac{\partial^2 U(x_i,t)}{\partial t^2} \right| \right\} \sum_{j=0}^{k} \int_{j\tau}^{(j+1)\tau} \frac{\mathrm{d}\xi}{(t_{k+1}-\xi)^\alpha} \\
&= \frac{C_U}{\Gamma(1-\alpha)} \sum_{j=0}^{k} \int_{j\tau}^{(j+1)\tau} \frac{\mathrm{d}\xi}{(t_{k+1}-\xi)^\alpha} \\
&= \frac{C_U \tau^{1-\alpha}}{\Gamma(2-\alpha)} \sum_{j=0}^{k} \bigl[(k+1-j)^{1-\alpha} - (k-j)^{1-\alpha}\bigr] \\
&\leq \frac{(k+1)^{1-\alpha} C_U \tau^{1-\alpha}}{\Gamma(2-\alpha)}.
\end{aligned}$$

Consider the explicit scheme (10) on $k+1$ time layer and the implicit scheme (11) on $k+2$ time layer:

$$ {}_0^C D_t^\alpha u(x_i, t_{k+1}) = \frac{1}{h^2}\bigl(U_{i-1}^k - 2U_i^k + U_{i+1}^k\bigr) - pU_i^k + f_i^k, \qquad (10)$$

$$ {}_0^C D_t^\alpha u(x_i, t_{k+2}) = \frac{1}{h^2}\bigl(U_{i-1}^{k+2} - 2U_i^{k+2} + U_{i+1}^{k+2}\bigr) - pU_i^{k+2} + f_i^{k+2}. \qquad (11)$$

Perform the Taylor expansion of schemes (10) and (11) at U_i^{k+1} respectively, and obtain the truncation errors $T_1(\tau, h)$ and $T_2(\tau, h)$ as follows:

$$\begin{aligned}
T_1(\tau,h) &= \frac{\partial^\alpha U(x_i,t_{k+1})}{\partial t^\alpha} - \frac{1}{h^2}\bigl(U_{i-1}^k - 2U_i^k + U_{i+1}^k\bigr) + pU_i^k \\
&= \frac{\partial^\alpha U(x_i,t_{k+1})}{\partial t^\alpha} - U_{xx} + pU + \tau U_{xxt} - p\tau U_t - \frac{h^2}{12} U_{xxxx} + O\bigl(\tau + h^2\bigr), \\
T_2(\tau,h) &= \frac{\partial^\alpha U(x_i,t_{k+2})}{\partial t^\alpha} - \frac{1}{h^2}\bigl(U_{i-1}^{k+2} - 2U_i^{k+2} + U_{i+1}^{k+2}\bigr) + pU_i^{k+2} \\
&= \frac{\partial^\alpha U(x_i,t_{k+1})}{\partial t^\alpha} + \tau \frac{\partial^{\alpha+1} U(x_i,t_{k+1})}{\partial t^{\alpha+1}} - U_{xx} + pU - \tau U_{xxt} + p\tau U_t \\
&\quad - \frac{h^2}{12} U_{xxxx} + O\bigl(\tau + h^2\bigr).
\end{aligned}$$

Consider C–N scheme (12) on $k+1$ time layer and C–N scheme (13) on $k+2$ time layer:

$$\begin{aligned}
{}_0^C D_t^\alpha u(x_i,t_{k+1}) &= \frac{1}{2h^2}\bigl(U_{i-1}^{k+1} - 2U_i^{k+1} + U_{i+1}^{k+1} + U_{i-1}^{k} - 2U_i^{k} + U_{i+1}^{k}\bigr) \\
&\quad - \frac{p}{2}\bigl(U_i^{k+1} + U_i^{k}\bigr) + \frac{1}{2}\bigl(f_i^{k+1} + f_i^{k}\bigr), \qquad (12) \\
{}_0^C D_t^\alpha u(x_i,t_{k+2}) &= \frac{1}{2h^2}\bigl(U_{i-1}^{k+2} - 2U_i^{k+2} + U_{i+1}^{k+2} + U_{i-1}^{k+1} - 2U_i^{k+1} + U_{i+1}^{k+1}\bigr) \\
&\quad - \frac{p}{2}\bigl(U_i^{k+2} + U_i^{k+1}\bigr) + \frac{1}{2}\bigl(f_i^{k+2} + f_i^{k+1}\bigr). \qquad (13)
\end{aligned}$$

Perform the Taylor expansion of schemes (12) and (13) at U_i^{k+1} respectively, and obtain the truncation errors $T_3(\tau,h)$ and $T_4(\tau,h)$ as follows:

$$\begin{aligned}
T_3(\tau,h) &= \frac{\partial^\alpha U(x_i,t_{k+1})}{\partial t^\alpha} - \frac{1}{2h^2}\bigl(U_{i-1}^{k+1} - 2U_i^{k+1} + U_{i+1}^{k+1} + U_{i-1}^{k} - 2U_i^{k} + U_{i+1}^{k}\bigr) \\
&\quad + \frac{p}{2}\bigl(U_i^{k+1} + U_i^{k}\bigr) \\
&= \frac{\partial^\alpha U(x_i,t_{k+1})}{\partial t^\alpha} - U_{xx} + pU + \frac{\tau}{2}U_{xxt} - \frac{p\tau}{2}U_t - \frac{h^2}{12}U_{xxxx} + O\bigl(\tau + h^2\bigr), \\
T_4(\tau,h) &= \frac{\partial^\alpha U(x_i,t_{k+2})}{\partial t^\alpha} - \frac{1}{2h^2}\bigl(U_{i-1}^{k+2} - 2U_i^{k+2} + U_{i+1}^{k+2} + U_{i-1}^{k+1} - 2U_i^{k+1} + U_{i+1}^{k+1}\bigr) \\
&\quad + \frac{p}{2}\bigl(U_i^{k+2} + U_i^{k+1}\bigr) \\
&= \frac{\partial^\alpha U(x_i,t_{k+1})}{\partial t^\alpha} + \tau\frac{\partial^{\alpha+1} U(x_i,t_{k+1})}{\partial t^{\alpha+1}} - U_{xx} + pU - \frac{\tau}{2}U_{xxt} + \frac{p\tau}{2}U_t \\
&\quad - \frac{h^2}{12}U_{xxxx} + O\bigl(\tau + h^2\bigr).
\end{aligned}$$

When explicit and implicit schemes are used alternately in different time layers, the τU_{xxt} and $p\tau U_t$ terms in the truncation errors $T_1(\tau,h)$ and $T_2(\tau,h)$ are canceled. $\tau\frac{\partial^{\alpha+1}U(x_i,t_{k+1})}{\partial t^{\alpha+1}} \le \widetilde{C}\tau^{2-\alpha}$ and $\widetilde{C} = \frac{(k+1)^{1-\alpha}C_U}{\Gamma(2-\alpha)} > 0$ are known. Therefore we can get that the scheme accuracy at the inner boundary points is $O(\tau^{2-\alpha} + h^2)$. Similarly, when C–N schemes are used alternately in different time layers, the $\frac{\tau}{2}U_{xxt}$ and $\frac{p\tau}{2}U_t$ terms of the truncation errors $T_3(\tau,h)$ and $T_4(\tau,h)$ are canceled. So the scheme accuracy at the inner points and the boundary at both ends is still $O(\tau^{2-\alpha} + h^2)$.

Theorem 3 *The computation accuracy of the IASC–N parallel difference scheme* (9) *for time fractional reaction–diffusion equation is* $O(\tau^{2-\alpha} + h^2)$.

4 Numerical experiments

The numerical experiments are based on Intel Core i5-3230 CPU and run in MatlabR2016a environment (Liu 2012) [42]. We consider the following fractional reaction–diffusion equation (Lin and Xu 2007; Jiang and Ma 2011) [17, 43]:

$$\begin{cases}
\frac{\partial^\alpha u(x,t)}{\partial t^\alpha} = \frac{\partial^2 u(x,t)}{\partial x^2} - pu(x,t) + f(x,t), & \\
u(x,0) = u_0(x), & 0 \le x \le 1, 0 \le t \le 1, \\
u(0,t) = u(2,t) = 0, &
\end{cases} \qquad (14)$$

where $f(x,t) = \frac{2}{\Gamma(3-\alpha)}t^{2-\alpha}\sin(2\pi x) + 4\pi^2 t^2 \sin(2\pi x)$, $u_0(x) = 0$, $p = 0$, and $0 < \alpha < 1$. It is easy to get its analytical solution as follows (Lin and Xu 2007; Jiang and Ma 2011) [17, 43]:

$$u(x,t) = t^2 \sin(2\pi x). \tag{15}$$

Take $\alpha = 0.7$, $M = 81$, $N = 1000$, $B = 5$, and $l = \frac{M-1}{B} = 16$, and give surface plots of the analytical solution (15), implicit scheme solution, C–N scheme solution, and IASC–N scheme solution (9) as follows. As can be seen from Fig. 2, Fig. 3, Fig. 4, and Fig. 5, the shape of the three scheme solution surfaces is consistent with the shape of the analytical solution surface, and the surface of the IASC–N scheme solution is smooth.

Let $t = 0.4$, and compare the IASC–N scheme solution, implicit scheme solution, and C–N scheme solution with the analytical solution. The scheme solution is very close to the analytical solution. The computation results are shown in Table 1.

Take $\alpha = 0.3, 0.5, 0.7$ respectively, and compare the scheme solutions with the analytical solutions at $t = 0.4$. The computation result is shown in Fig. 6. The scheme solution curve is very close to the analytical solution curve, indicating that the fractional order parameter α has little effect on the computation accuracy, that is, the fractional order does not have

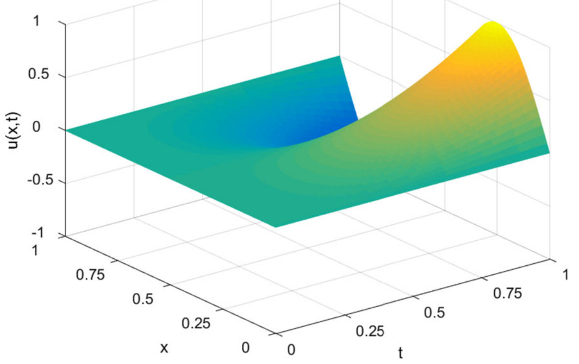

Figure 2 Surface of the analytical solution ($\alpha = 0.7$)

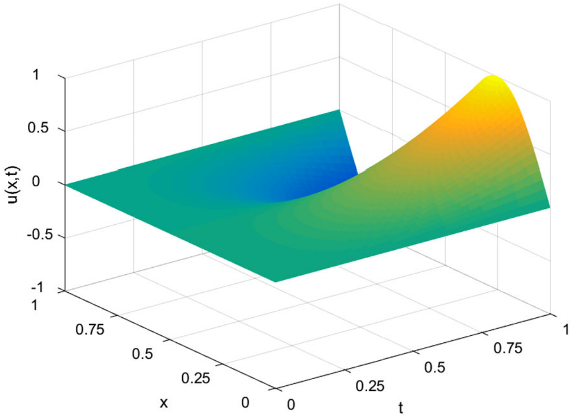

Figure 3 Surface of the implicit scheme solution ($\alpha = 0.7$)

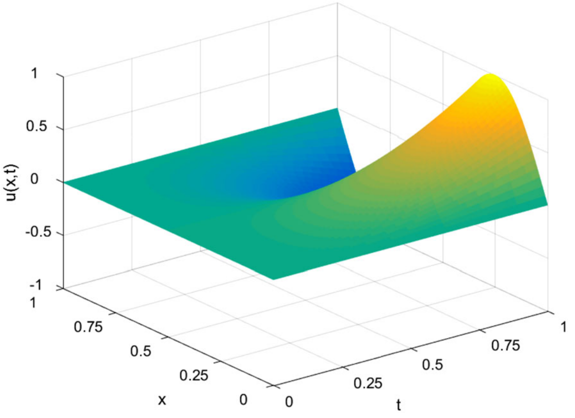

Figure 4 Surface of the C–N scheme solution ($\alpha = 0.7$)

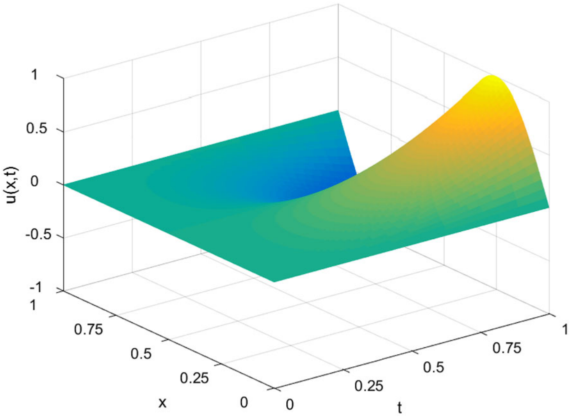

Figure 5 Surface of the IASC–N scheme solution ($\alpha = 0.7$)

Table 1 Comparison of scheme solutions and analytical solutions ($\alpha = 0.7$)

x	0.2	0.4	0.6	0.8
Analytical solution	0.152169	0.094046	−0.094046	−0.152169
Implicit scheme solution	0.152168	0.094023	−0.094023	−0.152168
C–N scheme solution	0.152184	0.094011	−0.094011	−0.152184
IASC–N scheme solution	0.151216	0.087818	−0.087822	−0.151209

much influence on the dynamic behavior of the system. Therefore, the IASC–N method is a high precision difference method for solving fractional reaction–diffusion equation.

In order to verify the stability and computation accuracy of the IASC–N scheme, take $\alpha = 0.7$, $M = 81$, and $N = 1000$, and give the change of relative error with time and space changing. Let the equation analytical solution u_i^k be the control solution and the scheme solution U_i^k be the disturbance solution. The sum of relative error for every time level (SRET) and the difference total energy (DTE) are defined as follows:

$$\mathrm{SRET}(k) = \sum_{i=1}^{M} \frac{|u_i^k - U_i^k|}{U_i^k}, \qquad \mathrm{DTE}(i) = \frac{1}{2}\sum_{k=1}^{N}\bigl(u_i^k - U_i^k\bigr)^2.$$

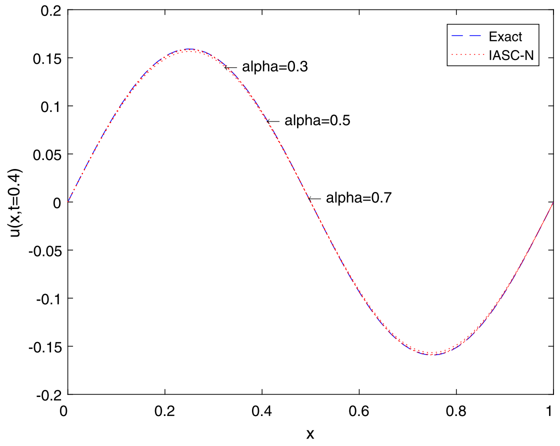

Figure 6 Comparison of the IASC–N scheme solution and the analytical solution

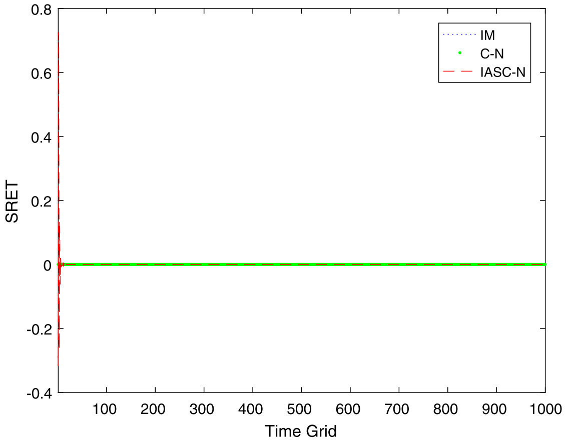

Figure 7 SRET curves of three schemes ($\alpha = 0.7$)

The SRET curve in Fig. 7 shows that although the error of the scheme solution is slightly larger when the number of time layers is small, as the number of time layers increases, SRET decreases rapidly and tends to zero, indicating that the IASC–N scheme for time fractional reaction–diffusion equation is computationally stable.

The DTE curve in Fig. 8 shows that the IASC–N scheme solution better approximates the analytical solution compared to the implicit scheme solution and the C–N scheme solution. The DTE curves of the three schemes have similar trends and are all less than 2.5×10^{-5}, indicating that the IASC–N scheme for time fractional reaction–diffusion equation has good computational accuracy.

The time convergence order and spatial convergence order of the IASC–N scheme are verified below. Define $E_2(h,\tau)$ as the L^2 mode, $Order1$ as the time convergence order, and $Order2$ as the spatial convergence order as follows (Jiang and Ma 2011) [43]:

$$E_2(h,\tau) = \max_{0\le k\le N} \left\| u_i^k - U_i^k \right\|_2,$$

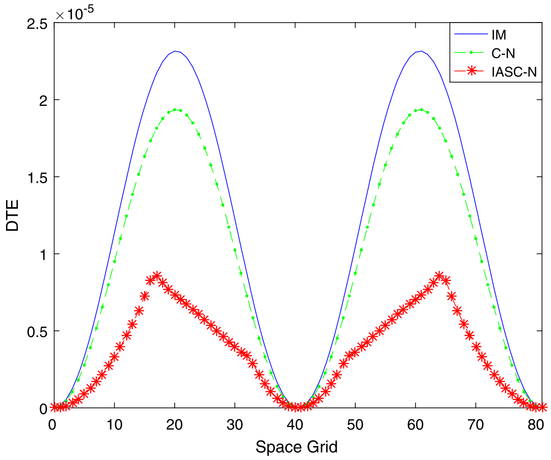

Figure 8 DTE curves of three schemes ($\alpha = 0.7$)

Table 2 Scheme errors and time convergence orders ($\alpha = 0.5$, $h = \frac{1}{80}$)

N	Implicit scheme		C–N scheme		IASC–N scheme	
	$E_2(h,\tau)$	*Order*1	$E_2(h,\tau)$	*Order*1	$E_2(h,\tau)$	*Order*1
5600	6.0126e–05	–	6.2566e–05	–	1.1149e–04	–
5800	5.7052e–05	1.4952	5.9409e–05	1.4756	1.0583e–04	1.4843
6000	5.4184e–05	1.5217	5.6462e–05	1.5008	1.0059e–04	1.4981
6200	5.1500e–05	1.5492	5.3705e–05	1.5269	9.5721e–05	1.5124
6400	4.8984e–05	1.5778	5.1120e–05	1.5538	9.1191e–05	1.5271

Table 3 Scheme errors and spatial convergence orders ($\alpha = 0.5$, $\tau = h^2$)

M	Implicit scheme		C–N scheme		IASC–N scheme	
	$E_2(h,\tau)$	*Order*2	$E_2(h,\tau)$	*Order*2	$E_2(h,\tau)$	*Order*2
101	3.2314e–04	–	3.3097e–04	–	6.2148e–04	–
201	8.0660e–05	2.0167	8.2618e–05	2.0166	1.5433e–04	2.0242
301	3.5826e–05	2.0098	3.6697e–05	2.0097	6.8421e–05	2.0144
401	2.0146e–05	2.0070	2.0635e–05	2.0069	3.8438e–05	2.0103
501	1.2891e–05	2.0054	1.3204e–05	2.0054	2.4571e–05	2.0097

$$Order1 = \frac{\ln(\frac{E_2(h,\tau_1)}{E_2(h,\tau_2)})}{\ln(\frac{\tau_1}{\tau_2})}, \qquad Order2 = \frac{\ln(\frac{E_2(h_1,\tau)}{E_2(h_2,\tau)})}{\ln(\frac{h_1}{h_2})}.$$

Take $\alpha = 0.5$ and $N = 5600, 5800, 6000, 6200, 6400$, and let $h = \frac{1}{80}$, that is, take $M = 81$. The computation results of the time convergence order are shown in Table 2. The accuracy of IASC–N scheme is close to $2 - \alpha$ order in time, which is consistent with the IASC–N scheme accuracy of $O(\tau^{2-\alpha} + h^2)$ in the theoretical analysis.

Take $\alpha = 0.5$ and $M = 101, 201, 301, 401, 501$, and let $\tau = h^2$, that is, take $N \approx \frac{M^2}{4}$. The computation results of the spatial convergence order are shown in Table 3. The accuracy of IASC–N scheme is spatially close to 2 order, which is similar to the error of the implicit scheme and the C–N scheme.

Finally, take $\alpha = 0.7$, $M = 3001, 4001, 5001, 6001, 7001$, and $N = 100$. Give the computing time and speedup (S_p) of the scheme solution in Table 4, where S_{p1} is the speedup of IASC–

Table 4 Computing time and speedup of scheme solution ($\alpha = 0.7$)

M	Implicit scheme/s	C–N scheme/s	IASC–N scheme/s	S_{p1}	S_{p2}
3001	2.3513	2.5916	1.9731	1.1916	1.3135
4001	5.1555	5.3499	3.6795	1.4011	1.4540
5001	8.9145	9.1686	5.7039	1.5629	1.6074
6001	19.7645	20.4187	8.5569	2.3098	2.3862
7001	48.9845	49.8756	11.5971	4.2238	4.3007

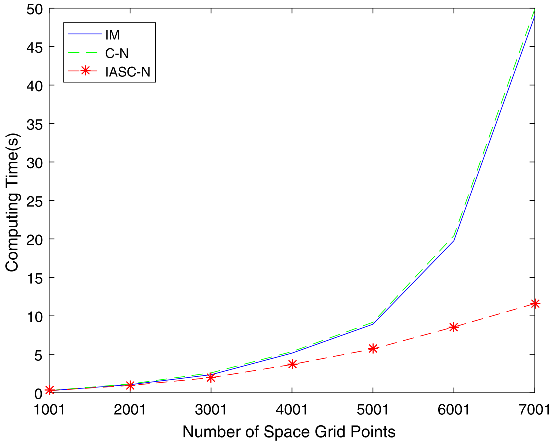

Figure 9 Comparison of computing time between the three schemes ($\alpha = 0.7$)

N scheme relative to the implicit scheme, and S_{p2} is the the speedup of IASC–N scheme relative to the C–N scheme.

Table 4 shows that the computing time of IASC–N scheme is much smaller than that of the implicit scheme and C–N scheme, and as the number of spatial grids increases, S_{p1} and S_{p2} also increase, indicating that the computing time growth rate of IASC–N scheme is smaller than the computing time growth rate of the implicit scheme and C–N scheme. It is valid to verify that the IASC–N scheme for time fractional reaction–diffusion equation is effective.

In order to more clearly compare the computational efficiency of IASC–N scheme, implicit scheme, and C–N scheme, take $\alpha = 0.7$ and $N = 100$, and give the variation of the computing time of the three schemes with the increase of the number of spatial grids as Fig. 9.

Figure 9 shows that when the number of spatial grids is small, the computing time of implicit scheme and C–N scheme is slightly smaller than the computing time of IASC–N scheme. When the number of spatial grids is larger than 2001, the parallel computing advantages of IASC–N scheme become more and more obvious, and the computational efficiency is higher than the serial scheme. This is due to data communication issues when using parallel computing programs (Zhu 1994) [44]. When the amount of data is small, the impact of data communication on the loop will greatly reduce the computational efficiency. As the amount of data increases, the superiority of parallel computing becomes significant gradually since the impact of execution of the program loop is much greater than the impact of data communication.

5 Conclusions

For a long time, a large number of parallel schemes have been designed to be conditionally stable or unconditionally stable but with only first order spatial accuracy (Yuan et al. 2015) [45]. In order to obtain the parallel difference scheme with higher accuracy and looser stability condition, this paper proposes a parallel computing method of IASC–N difference scheme for time fractional reaction–diffusion equation. Theoretically, we analyze the unique solvability, unconditional stability, convergence, second order spatial accuracy, and $2-\alpha$ order temporal accuracy of the IASC–N scheme. Numerical experiments verify the theoretical analysis, indicating that the proposed IASC–N difference scheme is more accurate and efficient than that given in [17, 43]. In particular, when the number of spatial grids is large enough, the IASC–N method has obvious localization features in terms of computation and communication and is suitable for operating in massive parallel computing systems. The parallel computing method of IASC–N difference scheme in this paper can be extended to high dimensional equations to solve the numerical solution of high dimensional fractional reaction–diffusion problems. We will also consider more numerical methods (Hajipour et al. 2018; Hajipour et al. 2018) [46, 47] applying to numerical solutions for other fractional evolution equations.

Acknowledgements
The authors would like to deeply thank Dr. Lifei Wu (School of Mathematics and Physics, North China Electric Power University) for his valuable suggestion and constructive comments during the preparation.

Funding
The research is supported by the National Natural Science Foundation of China (Grant No. 11371135) and the National Major Scientific and Technological Special Project (Grant No. 2017ZX07101001-01).

Competing interests
The authors declare that they have no competing interests.

Authors' contributions
Each of the authors contributed to each part of this work equally and read and approved the final version of the manuscript.

Publisher's Note

Received: 14 May 2019 Accepted: 21 September 2019 Published online: 01 October 2019

References

1. Baleanu, D., Asad, J.H., Jajarmi, A.: New aspects of the motion of a particle in a circular cavity. Proc. Rom. Acad., Ser. A **19**(2), 361–367 (2018)
2. Mohammadi, F., Moradi, L., Baleanu, D., Jajarmi, A.: A hybrid functions numerical scheme for fractional optimal control problems: application to non-analytic dynamical systems. J. Vib. Control **24**(21), 5030–5043 (2018)
3. Hajipour, M., Jajarmi, A., Baleanu, D., Sun, H.G.: On an accurate discretization of a variable-order fractional reaction–diffusion equation. Commun. Nonlinear Sci. Numer. Simul. **69**, 119–133 (2019)
4. Baleanu, D., Sajjadi, S.S., Jajarmi, A., Asad, J.H.: New features of the fractional Euler–Lagrange equations for a physical system within non-singular derivative operator. Eur. Phys. J. Plus **134**, 181 (2019). https://doi.org/10.1140/epjp/i2019-12561-x
5. Kumar, D., Singh, J., Baleanu, D., Sushila: Analysis of regularized long-wave equation associated with a new fractional operator with Mittag-Leffler type kernel. Physica A **492**, 155–167 (2018)
6. Kumar, D., Singh, J., Baleanu, D.: A new analysis of the Fornberg–Whitham equation pertaining to a fractional derivative with Mittag-Leffler-type kernel. Eur. Phys. J. Plus **133**, 70 (2018). https://doi.org/10.1140/epjp/i2018-11934-y
7. Singh, J., Kumar, D., Baleanu, D.: New aspects of fractional Biswas–Milovic model with Mittag-Leffler law. Math. Model. Nat. Phenom. **14**(3), 303 (2019). https://doi.org/10.1051/mmnp/2018068
8. Uchaikin, V.V.: Fractional Derivatives for Physicists and Engineers: Volume II: Applications. Higher Education Press, Beijing (2013)
9. Chen, W., Sun, H.G., Li, X.C.: Fractional Derivative Modeling for Mechanical and Engineering Problems. Science Press, Beijing (2010) (in Chinese)
10. Guo, B.L., Pu, X.K., Huang, F.H.: Fractional Partial Differential Equations and Their Numerical Solutions. Science Press, Beijing (2015)

11. Sabatier, J., Agrawal, O.P., Tenreiro Machado, J.A. (eds.): Advances in Fractional Calculus: Theoretical Developments and Applications in Physics and Engineering. World Publishing Corporation, Beijing (2014)
12. Singh, J., Kumar, D., Baleanu, D., Rathore, S.: An efficient numerical algorithm for the fractional Drinfeld–Sokolov–Wilson equation. Appl. Math. Comput. **335**, 12–24 (2018)
13. Kumar, D., Singh, J., Baleanu, D., Rathore, S.: Analysis of a fractional model of the Ambartsumian equation. Eur. Phys. J. Plus **133**, 259 (2018). https://doi.org/10.1140/epjp/i2018-12081-3
14. Goswami, A., Singh, J., Kumar, D., Sushila: An efficient analytical approach for fractional equal width equations describing hydro-magnetic waves in cold plasma. Physica A **524**, 563–575 (2019)
15. Sun, Z.Z., Gao, G.H.: Finite Difference Methods for Fractional Differential Equations. Science Press, Beijing (2015) (in Chinese)
16. Liu, F.W., Zhuang, P.H., Liu, Q.X.: Numerical Methods for Fractional Partial Differential Equations and Their Applications. Science Press, Beijing (2015) (in Chinese)
17. Lin, Y.M., Xu, C.J.: Finite difference/spectral approximations for the time-fractional diffusion equation. J. Comput. Phys. **225**(2), 1533–1552 (2007)
18. Liu, Y., Du, Y.W., Li, H., He, S., Gao, W.: Finite difference/finite element method for a nonlinear time-fractional fourth-order reaction–diffusion problem. Comput. Math. Appl. **70**(4), 573–591 (2015)
19. Liu, Y., Du, Y.W., Li, H., Wang, J.F.: An H^1-Galerkin mixed finite element method for time fractional reaction–diffusion equation. J. Appl. Math. Comput. **47**(1–2), 103–117 (2015)
20. Chen, H., Lu, S.J., Chen, W.P.: Finite difference/spectral approximations for the distributed order time fractional reaction–diffusion equation on an unbounded domain. J. Comput. Phys. **315**, 84–97 (2016)
21. Zhang, J.X., Yang, X.Z.: A class of efficient difference method for time fractional reaction–diffusion equation. Comput. Appl. Math. **37**(4), 4376–4396 (2018)
22. Bjorstad, P., Luskin, M.: Parallel Solution of Partial Differential Equations. Springer, Berlin (2000)
23. Chi, X.B., Wang, Y.W., Wang, J., Liu, F.: Parallel Computation and Implementation Technology. Science Press, Beijing (2015) (in Chinese)
24. Evans, D.J., Abdullah, A.R.B.: Group explicit methods for parabolic equations. Int. J. Comput. Math. **14**(1), 7–105 (1983)
25. Zhang, B.L., Yuan, G.X., Liu, X.P., Chen, J.: Parallel Finite Difference Methods for Partial Differential Equations. Science Press, Beijing (1994) (in Chinese)
26. Zhou, Y.L.: A finite difference scheme with intrinsic parallelism for quasilinear parabolic systems. Sci. China Ser. A, Math. **40**(1), 43–48 (1997) (in Chinese)
27. Wang, W.Q.: Difference schemes with intrinsic parallelism for the KdV equation. Acta Math. Appl. Sin. **29**(6), 995–1003 (2006) (in Chinese)
28. Yuan, G.W., Sheng, Z.Q., Hang, X.D.: The unconditional stability of parallel difference schemes with second order convergence for nonlinear parabolic system. J. Partial Differ. Equ. **20**(1), 45–64 (2007)
29. Wang, H., Wang, K.X., Sircar, T.: A direct $O(N \log^2 N)$ finite difference method for fractional diffusion equations. J. Comput. Phys. **229**(21), 8095–8104 (2010)
30. Diethelm, K.: An efficient parallel algorithm for the numerical solution of fractional differential equations. Fract. Calc. Appl. Anal. **14**(3), 475–490 (2011)
31. Wang, H., Basu, T.S.: A fast finite difference method for two-dimensional space-fractional diffusion equations. SIAM J. Sci. Comput. **34**, 2444–2458 (2012)
32. Moroney, T., Yang, Q.Q.: Efficient solution of two-sided nonlinear space-fractional diffusion equations using fast Poisson preconditioners. J. Comput. Phys. **246**(246), 304–317 (2013)
33. Gong, C.Y., Bao, W.M., Tang, G.J.: A parallel algorithm for the Riesz fractional reaction–diffusion equation with explicit finite difference method. Fract. Calc. Appl. Anal. **16**(3), 654–669 (2013)
34. Sweilam, N.H., Moharram, H., Moniem, N.K.A., Ahmed, S.: A parallel Crank–Nicolson finite difference method for time fractional parabolic equation. J. Numer. Math. **22**(4), 363–382 (2014)
35. Lu, X., Pang, H.K., Sun, H.W.: Fast approximate inversion of a block triangular Toeplitz matrix with applications to fractional sub-diffusion equations. Numer. Linear Algebra Appl. **22**(4), 866–882 (2015)
36. Wang, Q.L., Liu, J., Gong, C.Y., Tang, X.T., Fu, G.T., Xing, Z.C.: An efficient parallel algorithm for Caputo fractional reaction–diffusion equation with implicit finite-difference method. Adv. Differ. Equ. **2016**(1), 207 (2016). https://doi.org/10.1186/s13662-016-0929-9
37. Wu, L.F., Yang, X.Z., Cao, Y.H.: An alternating segment Crank–Nicolson parallel difference scheme for the time fractional sub-diffusion equation. Adv. Differ. Equ. **2018**(1), 287 (2018). https://doi.org/10.1186/s13662-018-1749-x
38. Biala, T.A., Khaliq, A.Q.M.: Parallel algorithms for nonlinear time-space fractional parabolic PDEs. J. Comput. Phys. **375**, 135–154 (2018)
39. Fu, H.F., Wang, H.: A preconditioned fast parallel finite difference method for space-time fractional partial differential equation. J. Sci. Comput. **78**(3), 1724–1743 (2019)
40. Lin, Y.M., Li, X.J., Xu, C.J.: Finite difference/spectral approximations for the fractional cable equation. Math. Comput. **80**(275), 1369–1396 (2011)
41. Gao, G.H., Sun, Z.Z., Zhang, H.W.: A new fractional numerical differentiation formula to approximate the Caputo fractional derivative and its applications. J. Comput. Phys. **259**(2), 33–50 (2014)
42. Liu, W.: The Actual Combat Matlab Parallel Programming. Beihang University Press, Beijing (2012) (in Chinese)
43. Jiang, Y.J., Ma, J.T.: High-order finite element methods for time-fractional partial differential equations. J. Comput. Appl. Math. **235**(11), 3285–3290 (2011)
44. Zhu, J.P.: Solving Partial Differential Equations on Parallel Computers. World Scientific, Singapore (1994)
45. Yuan, G.W., Sheng, Z.Q., Hang, X.D., Yao, Y.Z., Chang, L.N., Yue, J.Y.: Computation Methods for Diffusion Equations. Science Press, Beijing (2015) (in Chinese)
46. Hajipour, M., Jajarmi, A., Malek, A., Baleanu, D.: Positivity-preserving sixth-order implicit finite difference weighted essentially non-oscillatory scheme for the nonlinear heat equation. Appl. Math. Comput. **325**, 146–158 (2018)
47. Hajipour, M., Jajarmi, A., Baleanu, D.: On the accurate discretization of a highly nonlinear boundary value problem. Numer. Algorithms **79**(3), 679–695 (2018)