TAIWANESE JOURNAL OF MATHEMATICS
Vol. 25, No. 5, pp. 959–979, October 2021
DOI: 10.11650/tjm/210501

A Pseudo-spectral Method for Time Distributed Order Two-sided Space Fractional Differential Equations

Shina Daniel Oloniiju*, Sicelo Praisegod Goqo and Precious Sibanda

Abstract. Time distributed order two-sided space differential equations of arbitrary order offer a robust approach to modelling complex dynamical systems. In this study, we describe a scheme for obtaining the numerical solutions of time distributed order multidimensional two-sided space fractional differential equations. The numerical discretization scheme is a hybrid scheme, comprising a Newton–Cotes quadrature formula and a spectral collocation method. The time distributed order fractional differential operator is approximated using the composite Simpson's rule, and the solution of the resulting differential equation is expressed as a linear combination of shifted Chebyshev polynomials in all variables. Convergence analysis of the numerical scheme is presented. Some one- and two-dimensional time distributed order two-sided space fractional differential equations, such as the fractional advection-dispersion and diffusion equations, are presented to demonstrate the accuracy and computational efficiency of the numerical scheme, and numerical solutions are compared with the exact solutions, where these are available.

1. Introduction

In the past few decades, the number of studies on fractional calculus has grown tremendously owing to the vast amount of physical phenomena that can be modelled using fractional differential, integral and integrodifferential equations. Several mathematical models such as diffusion [3,20,22], viscoelasticity [2,13,16,19], growth model [11] and flow in porous media [7] have been generalized to include fractional derivatives. Fractional models are, in general non-local, which make them ideal for modelling multi-scale phenomena. Some physical systems are complex and cannot be modelled by a fixed order law, and so using multiple fixed orders leads to distributed order fractional models. Distributed order fractional models have been regarded as a robust tool for modelling sophisticated dynamical systems.

Received July 14, 2020; Accepted April 29, 2021.

Communicated by Chiu-Yen Kao.

2020 *Mathematics Subject Classification.* 26A33, 65M70.

Key words and phrases. numerical method, Chebyshev–Gauss–Lobatto quadrature, two-sided space, time distributed order, fractional partial differential equation.

*Corresponding author.

In this study, we consider a time distributed order two-sided space fractional differential equation of the form

$$(1.1) \qquad \int_{\beta_l}^{\beta_u} \rho(\beta)\, {}_0^C D_t^\beta u(\mathbf{x},t)\, d\beta = {}_0^C D_{\mathbf{x}}^\alpha u(\mathbf{x},t) + {}_{\mathbf{x}}^C D_{\mathsf{L}}^\alpha u(\mathbf{x},t) + g(\mathbf{x},t)$$

with appropriate initial and boundary conditions on the finite spatial interval $[0,\mathsf{L}]$ and temporal domain $[0,\mathsf{T}]$, where $\alpha,\beta \in \mathbb{R}^+$. Here, ${}_0^C D_{\mathbf{x}}^\alpha$ and ${}_{\mathbf{x}}^C D_{\mathsf{L}}^\alpha$ respectively denote the left and right sided Caputo fractional differential operators defined as [16,19]

$${}_0^C D_{\mathbf{x}}^\alpha u(\mathbf{x},t) = \frac{1}{\Gamma(n-\alpha)} \int_0^{\mathbf{x}} \frac{u^n(\widetilde{\mathbf{x}},t)}{(\mathbf{x}-\widetilde{\mathbf{x}})^{\alpha+1-n}}\, d\widetilde{\mathbf{x}}, \quad \mathbf{x} > 0,\ n \in \mathbb{N}$$

and

$${}_{\mathbf{x}}^C D_{\mathsf{L}}^\alpha u(\mathbf{x},t) = \frac{1}{\Gamma(n-\alpha)} \int_{\mathbf{x}}^{\mathsf{L}} \frac{u^n(\widetilde{\mathbf{x}},t)}{(\widetilde{\mathbf{x}}-\mathbf{x})^{\alpha+1-n}}\, d\widetilde{\mathbf{x}}, \quad \mathbf{x} < \mathsf{L},\ n \in \mathbb{N}$$

provided $u(\mathbf{x},t)$ is a continuously differentiable function. The class of fractional differential equation, as in (1.1), has been proposed to have applications in the description of anomalous diffusion, advection-dispersion process and relaxation phenomena in which there is a temporal change in the diffusion exponent. We refer readers to the study in Cai and Li [4] for the regularity of the solution of a two-sided space fractional differential equation, and to Stojanović [21] for the regularity of the solution of a distributed order fractional differential equation.

One of the many challenges in solving distributed order differential equations is the huge computational cost involved compared to fixed order fractional differential equations. Among the few studies that have been dedicated to obtaining numerical solutions of time distributed order fractional partial differential equations are [9,12,14,15]. In Hu et al. [10], an implicit difference method was used to solve a one-dimensional differential equation of the form (1.1). A literature search reveals no studies where a pseudo-spectral method has been applied to time distributed order multidimensional two-sided space fractional differential equations. For this reason, we propose a pseudo-spectral numerical solution for this class of fractional differential equations. We consider equations with smooth and non-smooth solutions.

This study presents a numerical scheme to approximate the solution of (1.1) by approximating the integral using Simpson's rule. This leads to time multi-term fractional order two-sided space fractional differential equation which is approximated in terms of first kind shifted Chebyshev polynomials interpolated using Gauss–Lobatto quadrature. We illustrate the method using carefully chosen examples and compare the numerical results with exact solutions where available.

2. The numerical method

This section presents the numerical method of solution for (1.1). The section is divided into several subsections which describe the quadrature rule and the numerical discretization method used in approximating arbitrary order derivatives in the time multi-term two-sided space fractional differential equations.

2.1. Quadrature rule: Simpson's 1/3 rule

We approximate the distributed order derivative as a finite sum by discretizing the interval $[\beta_l, \beta_u]$ using the points $\beta_l = \beta_0 < \beta_1 < \cdots < \beta_Q = \beta_u$ and define $\Delta\beta = (\beta_u - \beta_l)/Q$, where the points are evenly spaced defined as $\beta_e = \beta_l + e\Delta\beta$, $e = 0, 1, 2, \ldots, Q-1, Q$. Then using the Simpson's 1/3 quadrature formula, we have

$$\begin{aligned}\int_{\beta_l}^{\beta_u} \rho(\beta)\, {}_0^C D_t^{\beta} u(\mathbf{x}, t)\, d\beta = \frac{\Delta\beta}{3}\Bigg[&\rho(\beta_0)\, {}_0^C D_t^{\beta_0} u(\mathbf{x}, t) + 4\sum_{e=1}^{Q/2} \rho(\beta_{2e-1})\, {}_0^C D_t^{\beta_{2e-1}} u(\mathbf{x}, t)\\ &+ 2\sum_{e=1}^{Q/2-1} \rho(\beta_{2e})\, {}_0^C D_t^{\beta_{2e}} u(\mathbf{x}, t) + \rho(\beta_Q)\, {}_0^C D_t^{\beta_Q} u(\mathbf{x}, t)\Bigg].\end{aligned}$$

The time distributed order two-sided space fractional differential equation (1.1) is now transformed into a time multi-term two-sided space fractional differential equation

$$\begin{aligned}&\frac{\Delta\beta}{3}\Bigg[\rho(\beta_0)\, {}_0^C D_t^{\beta_0} u(\mathbf{x}, t) + \rho(\beta_Q)\, {}_0^C D_t^{\beta_Q} u(\mathbf{x}, t) + 4\sum_{e=1}^{Q/2} \rho(\beta_{2e-1})\, {}_0^C D_t^{\beta_{2e-1}} u(\mathbf{x}, t)\\ &\quad + 2\sum_{e=1}^{Q/2-1} \rho(\beta_{2e})\, {}_0^C D_t^{\beta_{2e}} u(\mathbf{x}, t)\Bigg] + \mathcal{O}((\Delta\beta)^2)\\ &= {}_0^C D_{\mathbf{x}}^{\alpha} u(\mathbf{x}, t) + {}_{\mathbf{x}}^C D_{\mathsf{L}}^{\alpha} u(\mathbf{x}, t) + g(\mathbf{x}, t).\end{aligned} \tag{2.1}$$

Next, we approximate the functions in (2.1) and their derivatives in terms of the first kind shifted Chebyshev polynomials.

2.2. First kind Chebyshev approximation

Definition 2.1. [1] The first kind Chebyshev polynomials are eigenvalue solutions of the Sturm–Liouville problem with weight function $1/\sqrt{1-x^2}$. Considering a change of variable $[-1, 1] \mapsto [0, \mathsf{L}]$, the shifted form of the polynomials is defined in series form as

$$T_{\mathsf{L},n}(x) = n\sum_{j=0}^{n} \frac{(-1)^{n-j}(n+j-1)!2^{2j}}{(n-j)!(2j)!\mathsf{L}^j} x^j = n\sum_{j=0}^{n} \frac{(n+j-1)!2^{2j}}{(n-j)!(2j)!\mathsf{L}^j}(x-\mathsf{L})^j.$$

We approximate the solution of (2.1) as a linear combination of the shifted first kind Chebyshev polynomials in both variables as

$$u(\mathbf{x},t) \approx U(\mathbf{x},t) = \sum_{n_1=0}^{N_\mathbf{x}} \sum_{n_2=0}^{N_t} \widehat{U}_{n_1,n_2} T_{\mathsf{L},n_1}(\mathbf{x}) T_{\mathsf{T},n_2}(t), \quad \mathbf{x} \in [0,\mathsf{L}],\ t \in [0,\mathsf{T}],$$

where $\widehat{U}_{n_1,n_2}$ satisfies the L^2 orthogonality condition written in discrete form as

$$\widehat{U}_{n_1,n_2} = \frac{1}{h_{n_1}} \frac{1}{h_{n_2}} \sum_{j_1=0}^{N_\mathbf{x}} \sum_{j_2=0}^{N_t} \varpi_{j_1} \varpi_{j_2} U(\mathbf{x}_{j_1}, t_{j_2}) T_{\mathsf{L},n_1}(\mathbf{x}_{j_1}) T_{\mathsf{T},n_2}(t_{j_2}).$$

Here, $h_n = c_n \pi/2$ with $c_0 = 2$, $c_n = 1$, $\forall\, n \geq 1$ and the Christoffel number $\varpi_j = \pi/c_j N$ with $c_0 = c_N = 2$, $c_j = 1$, $\forall\, 1 \leq j \leq N-1$. Therefore $U(\mathbf{x},t)$ is given as

$$U(\mathbf{x},t) = \sum_{j_1=0}^{N_\mathbf{x}} \sum_{j_2=0}^{N_t} \left[\varpi_{j_1} \varpi_{j_2} \sum_{n_1=0}^{N_\mathbf{x}} \sum_{n_2=0}^{N_t} \frac{1}{h_{n_1}} \frac{1}{h_{n_2}} T_{\mathsf{L},n_1}(\mathbf{x}_{j_1}) T_{\mathsf{L},n_1}(\mathbf{x}_{p_1}) T_{\mathsf{T},n_2}(t_{j_2}) T_{\mathsf{T},n_2}(t_{p_2}) \right] \times U(\mathbf{x}_{j_1}, t_{j_2}), \quad p_1 = 0,1,\ldots,N_\mathbf{x},\ p_2 = 0,1,\ldots,N_t.$$

Lemma 2.2. [8, 17, 18] *Let $\alpha > 0$ and $\mathbf{x} > 0$. Suppose that $u(\mathbf{x},t)$ is a continuously differentiable function, then the approximation of the left sided derivative with respect to $\mathbf{x}$ is given as*

$$\begin{aligned} &{}_0^C D_\mathbf{x}^\alpha U(\mathbf{x},t) \\ &= \sum_{j_1=0}^{N_\mathbf{x}} \sum_{j_2=0}^{N_t} \left[\varpi_{j_1} \varpi_{j_2} \sum_{n_1=0}^{N_\mathbf{x}} \sum_{n_2=0}^{N_t} \sum_{k=0}^{N_\mathbf{x}} \frac{1}{h_{n_1}} \frac{1}{h_{n_2}} T_{\mathsf{L},n_1}(\mathbf{x}_{j_1}) T_{\mathsf{T},n_2}(t_{j_2}) T_{\mathsf{T},n_2}(t_{p_2})\, {}^l D_{n_1,k}^{(\alpha)} T_{\mathsf{L},k}(\mathbf{x}_{p_1}) \right] \\ &\quad \times U(\mathbf{x}_{j_1}, t_{j_2}), \end{aligned}$$

where

$${}^l D_{n_1,k}^{(\alpha)} = n_1 \sum_{j_1=0}^{n_1} \frac{(-1)^{n_1-j_1}(n_1+j_1-1)!2^{2j_1}}{(n_1-j_1)!(2j_1)!\mathsf{L}^{j_1}} \frac{\Gamma(j_1+1)}{\Gamma(j_1-\alpha+1)} q_{j_1,k},$$

and $q_{j_1,k} = 0$ for $j_1 = 0,1,\ldots,\lceil\alpha\rceil - 1$ and

$$q_{j_1,k} = \frac{k\sqrt{\pi}}{h_k} \sum_{r=0}^{k} \frac{(-1)^{k-r}(k+r-1)!2^{2r}}{(k-r)!(2r)!} \mathsf{L}^{j_1-\alpha} \frac{\Gamma(j_1-\alpha+r+1/2)}{\Gamma(j_1-\alpha+r+1)}$$

for $j_1 = \lceil\alpha\rceil, \lceil\alpha\rceil+1,\ldots,N$, $k = 0,1,\ldots,N$.

The approximation in the above lemma can also be used for the temporal fractional derivatives.

Lemma 2.3. *For $\alpha > 0$ and $\mathbf{x} < \mathsf{L}$, and if $u(\mathbf{x}, t)$ is a smooth function, the right sided derivative of arbitrary order is discretized as*

$$\begin{aligned}
&{}^{C}_{\mathbf{x}}D^{\alpha}_{\mathsf{L}}U(\mathbf{x}, t)\\
&= \sum_{j_1=0}^{N_{\mathbf{x}}}\sum_{j_2=0}^{N_t}\left[\varpi_{j_1}\varpi_{j_2}\sum_{n_1=0}^{N_{\mathbf{x}}}\sum_{n_2=0}^{N_t}\sum_{k=0}^{N_{\mathbf{x}}}\frac{1}{h_{n_1}}\frac{1}{h_{n_2}}T_{\mathsf{L},n_1}(\mathbf{x}_{j_1})T_{\mathsf{T},n_2}(t_{j_2})T_{\mathsf{T},n_2}(t_{p_2})\,{}^{r}D^{(\alpha)}_{n_1,k}T_{\mathsf{L},k}(\mathbf{x}_{p_1})\right]\\
&\qquad \times U(\mathbf{x}_{j_1}, t_{j_2}).
\end{aligned}$$

Here,

$$\begin{aligned}
{}^{r}D^{(\alpha)}_{n_1,k} &= n_1\sum_{j_1=0}^{n_1}\frac{(-1)^{j_1}(n_1+j_1-1)!2^{2j_1}}{(n_1-j_1)!(2j_1)!\mathsf{L}^{\alpha}}\frac{\Gamma(j_1+1)}{\Gamma(j_1-\alpha+1)}\\
&\quad \times\frac{2k}{\pi c_k}\sum_{r=0}^{k}\frac{(-1)^r(k+r-1)!2^{2r}}{(k-r)!(2r)!}\frac{\sqrt{\pi}\Gamma(j_1-\alpha+r+1/2)}{\Gamma(j_1-\alpha+r+1)}.
\end{aligned}$$

The discretizations in (2.1), Lemmas 2.2 and 2.3 lead to a linear algebraic system which is evaluated using shifted Gauss–Lobatto quadrature. This can be extended to fractional differential equations in three variables.

3. Space of fractional derivatives and convergence analysis

Here, we give the convergence analysis of the numerical discretization scheme described above. To this end, we define the domain $\vartheta = \tau \times \chi$, where $\tau = [0, \mathsf{T}]$ and $\chi = [0, \mathsf{L}] \times [0, \mathsf{L}]$, and the $L^2_w(\vartheta)$ norm defined as $\|\cdot\|_{\vartheta}$. We introduce the fractional Sobolev space $H^{\beta}_w(\vartheta)$, $\beta \geq 0$ defined as

$$H^{\beta}_w(\vartheta) = \{u \in L^2_w(\vartheta) : {}^{C}D^{\beta}u \in L^2_w(\vartheta)\}.$$

Given $\alpha > 0$, we define the semi-norm $|u|_{l,\alpha}$ and associated norm $\|u\|_{l,\alpha}$ respectively as [23]

$$|u|_{l,\alpha} = \left(\|{}^{C}_{0}D^{\alpha}_{\mathbf{x}}\|^2_{L^2_w(\vartheta)}\right)^{1/2} \quad \text{and} \quad \|u\|_{l,\alpha} = \left(\|u\|^2_{L^2_w(\vartheta)} + |u|^2_{l,\alpha}\right)^{1/2},$$

and the semi-norm $|u|_{r,\alpha}$ and associated norm $\|u\|_{r,\alpha}$ respectively as

$$|u|_{r,\alpha} = \left(\|{}^{C}_{\mathbf{x}}D^{\alpha}_{\mathsf{L}}\|^2_{L^2_w(\vartheta)}\right)^{1/2} \quad \text{and} \quad \|u\|_{r,\alpha} = \left(\|u\|^2_{L^2_w(\vartheta)} + |u|^2_{r,\alpha}\right)^{1/2}.$$

Lemma 3.1. [5,6] *Let $u \in H^n_w(\tau)$ and P_N be the orthogonal projector into space of shifted Chebyshev polynomials up to degree $N+1$, then we have the following estimates*

$$\begin{aligned}
\|u - P_N u\|_{H^k_w(\tau)} &\leq CN^{2k-n}\|u\|_{H^n_w(\tau)},\\
\|u - P_N u\|_{L^2_w(\tau)} &\leq CN^{-m}\|u\|_{H^n_w(\tau)}
\end{aligned}$$

for k and n real numbers, such that $0 \leq k \leq n$ and C, a positive constant independent of N.

Lemma 3.2. [6] *For $u \in H_w^{m,n}(\vartheta)$, $m, n \geq 0$, we have*

$$\|u - P_{N,N_t}u\|_{L_w^2(\vartheta)} \leq C\big(N_t^{-n}\|u\|_{H_w^{0,n}(\vartheta)} + N^{-m}\|u\|_{H_w^{m,0}(\vartheta)}\big).$$

The exact solution of (1.1) is

$$\begin{aligned}
&\frac{\Delta\beta}{3}\Bigg[\rho(\beta_0)\,{}_0^C D_t^{\beta_0}u(\mathbf{x},t) + \rho(\beta_Q)\,{}_0^C D_t^{\beta_Q}u(\mathbf{x},t) + 4\sum_{e=1}^{Q/2}\rho(\beta_{2e-1})\,{}_0^C D_t^{\beta_{2e-1}}u(\mathbf{x},t)\\
&\qquad + 2\sum_{e=1}^{Q/2-1}\rho(\beta_{2e})\,{}_0^C D_t^{\beta_{2e}}u(\mathbf{x},t)\Bigg] + \mathcal{O}((\Delta\beta)^2)\\
&= {}_0^C D_{\mathbf{x}}^{\alpha}u(\mathbf{x},t) + {}_{\mathbf{x}}^C D_{\mathsf{L}}^{\alpha}u(\mathbf{x},t) + g(\mathbf{x},t)
\end{aligned} \tag{3.1}$$

provided $u(\mathbf{x}, t)$ is regular with respect to all $\beta \in [\beta_0, \beta_Q]$. The numerical approximation scheme is

$$\begin{aligned}
&\frac{\Delta\beta}{3}\Bigg[\rho(\beta_0)\,{}_0^C D_t^{\beta_0}U(\mathbf{x},t) + \rho(\beta_Q)\,{}_0^C D_t^{\beta_Q}U(\mathbf{x},t) + 4\sum_{e=1}^{Q/2}\rho(\beta_{2e-1})\,{}_0^C D_t^{\beta_{2e-1}}U(\mathbf{x},t)\\
&\qquad + 2\sum_{e=1}^{Q/2-1}\rho(\beta_{2e})\,{}_0^C D_t^{\beta_{2e}}U(\mathbf{x},t)\Bigg]\\
&= {}_0^C D_{\mathbf{x}}^{\alpha}U(\mathbf{x},t) + {}_{\mathbf{x}}^C D_{\mathsf{L}}^{\alpha}U(\mathbf{x},t) + G(\mathbf{x},t).
\end{aligned} \tag{3.2}$$

Theorem 3.3. *Let $u(\mathbf{x}, t)$ be the exact solution of* (1.1) *and $U(\mathbf{x}, t)$ be the solution given by the numerical scheme, such that $u \in H_w^{m,n}(\vartheta)$. Then, there exists a positive constant C independent of $\Delta\beta$, N and N_t such that*

$$\begin{aligned}
\|u - U\|_{L_w^2(\vartheta)} \leq C\Bigg(&(\Delta\beta)^4 + N_t^{2\beta_0 - n}\|\xi\|_{0,n} + N_t^{2\beta_Q - n}\|\xi\|_{0,n} + \sum_{e=1}^{Q/2}N_t^{2\beta_{2e-1}-n}\|\xi\|_{0,n}\\
&+ \sum_{e=1}^{Q/2-1}N_t^{2\beta_{2e}-n}\|\xi\|_{0,n} + N^{2\alpha - m}\|\xi\|_{m,0} + N^{-m}\|u\|_{m,0} + N_t^{-n}\|u\|_{0,n}\Bigg).
\end{aligned}$$

Proof. To prove this result, define $\widetilde{u} = P_{N,N_t}u$, $\xi = \widetilde{u} - U$ and note that $g = G$. Subtracting (3.2) from (3.1) yields the following error equation

$$\begin{aligned}
&\frac{\Delta\beta}{3}\Bigg[\rho(\beta_0)\,{}_0^C D_t^{\beta_0}(u-U) + \rho(\beta_Q)\,{}_0^C D_t^{\beta_Q}(u-U) + 4\sum_{e=1}^{Q/2}\rho(\beta_{2e-1})\,{}_0^C D_t^{\beta_{2e-1}}(u-U)\\
&\qquad + 2\sum_{e=1}^{Q/2-1}\rho(\beta_{2e})\,{}_0^C D_t^{\beta_{2e}}(u-U)\Bigg] + \mathcal{O}((\Delta\beta)^2)\\
&= {}_0^C D_{\mathbf{x}}^{\alpha}(u-U) + {}_{\mathbf{x}}^C D_{\mathsf{L}}^{\alpha}(u-U).
\end{aligned} \tag{3.3}$$

Projecting (3.3) into the space of shifted Chebyshev polynomials gives

$$\begin{aligned}&\frac{\Delta\beta}{3}\rho(\beta_0)\,{}_0^CD_t^{\beta_0}\xi+\frac{\Delta\beta}{3}\rho(\beta_Q)\,{}_0^CD_t^{\beta_Q}\xi+\frac{4\Delta\beta}{3}\sum_{e=1}^{Q/2}\rho(\beta_{2e-1})\,{}_0^CD_t^{\beta_{2e-1}}\xi\\ &\quad+\frac{2\Delta\beta}{3}\sum_{e=1}^{Q/2-1}\rho(\beta_{2e})\,{}_0^CD_t^{\beta_{2e}}\xi\\ &=({}_0^CD_{\mathsf{x}}^{\alpha}+{}_{\mathsf{x}}^CD_{\mathsf{L}}^{\alpha})\xi+\epsilon+\mathcal{O}((\Delta\beta)^2),\end{aligned}\tag{3.4}$$

where

$$\begin{aligned}\epsilon=-\frac{\Delta\beta}{3}\Bigg[&\rho(\beta_0)\,{}_0^CD_t^{\beta_0}(u-\widetilde{u})+\rho(\beta_Q)\,{}_0^CD_t^{\beta_Q}(u-\widetilde{u})+4\sum_{e=1}^{Q/2}\rho(\beta_{2e-1})\,{}_0^CD_t^{\beta_{2e-1}}(u-\widetilde{u})\\ &+2\sum_{e=1}^{Q/2-1}\rho(\beta_{2e})\,{}_0^CD_t^{\beta_{2e}}(u-\widetilde{u})\Bigg]+({}_0^CD_{\mathsf{x}}^{\alpha}+{}_{\mathsf{x}}^CD_{\mathsf{L}}^{\alpha})(u-\widetilde{u}).\end{aligned}$$

Applying Young's inequality and properties of the Sobolev norm on the left-hand side of (3.4), we have

$$\begin{aligned}&\Bigg\|\frac{\Delta\beta}{3}\rho(\beta_0)\,{}_0^CD_t^{\beta_0}\xi+\frac{\Delta\beta}{3}\rho(\beta_Q)\,{}_0^CD_t^{\beta_Q}\xi+\frac{4\Delta\beta}{3}\sum_{e=1}^{Q/2}\rho(\beta_{2e-1})\,{}_0^CD_t^{\beta_{2e-1}}\xi\\ &\quad+\frac{2\Delta\beta}{3}\sum_{e=1}^{Q/2-1}\rho(\beta_{2e})\,{}_0^CD_t^{\beta_{2e}}\xi\Bigg\|_{L^2_w(\vartheta)}\\ &\le\Bigg\|\frac{\Delta\beta}{3}\rho(\beta_0)\,{}_0^CD_t^{\beta_0}\xi\Bigg\|_{L^2_w(\vartheta)}+\Bigg\|\frac{\Delta\beta}{3}\rho(\beta_Q)\,{}_0^CD_t^{\beta_Q}\xi\Bigg\|_{L^2_w(\vartheta)}\\ &\quad+\Bigg\|\frac{4\Delta\beta}{3}\sum_{e=1}^{Q/2}\rho(\beta_{2e-1})\,{}_0^CD_t^{\beta_{2e-1}}\xi\Bigg\|_{L^2_w(\vartheta)}+\Bigg\|\frac{2\Delta\beta}{3}\sum_{e=1}^{Q/2-1}\rho(\beta_{2e})\,{}_0^CD_t^{\beta_{2e}}\xi\Bigg\|_{L^2_w(\vartheta)}\\ &\le C\Bigg(N_t^{2\beta_0-n}\|\xi\|_{0,n}+N_t^{2\beta_Q-n}\|\xi\|_{0,n}+\sum_{e=1}^{Q/2}N_t^{2\beta_{2e-1}-n}\|\xi\|_{0,n}+\sum_{e=1}^{Q/2-1}N_t^{2\beta_{2e}-n}\|\xi\|_{0,n}\Bigg).\end{aligned}\tag{3.5}$$

Estimating the first term on the right-hand side of (3.4) gives

$$\big\|({}_0^CD_{\mathsf{x}}^{\alpha}+{}_{\mathsf{x}}^CD_{\mathsf{L}}^{\alpha})\xi\big\|_{L^2_w(\vartheta)}\le CN^{2\alpha-m}\|\xi\|_{m,0},\tag{3.6}$$

and for the second term, we get

$$\|\epsilon\|_{L^2_w(\vartheta)}\le C\big(N^{-m}\|u\|_{m,0}+N_t^{-n}\|u\|_{0,n}\big).\tag{3.7}$$

Applying the triangle inequality and substituting (3.5)–(3.7) into (3.4) yields

$$\|u-U\|_{L^2_w(\vartheta)}=\|\xi+\epsilon\|_{L^2_w(\vartheta)}\le\|\xi\|_{L^2_w(\vartheta)}+\|\epsilon\|_{L^2_w(\vartheta)}$$

$$\leq C\Bigg((\Delta\beta)^4 + N_t^{2\beta_0 - n}\|\xi\|_{0,n} + N_t^{2\beta_Q - n}\|\xi\|_{0,n} + \sum_{e=1}^{Q/2} N_t^{2\beta_{2e-1} - n}\|\xi\|_{0,n}$$

$$+ \sum_{e=1}^{Q/2-1} N_t^{2\beta_{2e} - n}\|\xi\|_{0,n} + N^{2\alpha - m}\|\xi\|_{m,0} + N^{-m}\|u\|_{m,0} + N_t^{-n}\|u\|_{0,n}\Bigg),$$

which completes the proof. □

4. Numerical examples

In this section, we experiment with the numerical scheme described earlier with some time distributed order two-sided space fractional partial differential equations. We consider both one- and two-spatial dimensional two-sided space differential equations of arbitrary orders. In the examples, we consider the fractional advection-dispersion and fractional diffusion equations. The accuracy of the method is investigated by comparing the exact solution of these equations with obtained numerical solutions. The results are presented in form of tables and graphs. The results emphasize computational efficiency in terms of CPU time and accuracy which is measured in terms of the absolute difference between the exact and numerical solutions. The error is defined as

$$E = \|u(\mathbf{x}_i, t_k) - U(\mathbf{x}_i, t_k)\|,$$

where u and U respectively represent the exact and numerical solutions. We define the order of convergence as

$$\text{Order} = \frac{\log(\|E(N_1)\|_\infty / \|E(N_2)\|_\infty)}{\log(N_2/N_1)},$$

where $N_1 \neq N_2$. The numerical scheme was implemented using the Python 3.6 programming language on the Spyder ide run on a computer with *Intel Core i5, 7th gen, CPU @ 2.50 GHz* and *8 GB DDR4 installed RAM.*

Example 4.1. [10] We consider the one dimensional time distributed order two-sided space fractional advection-dispersion equation

$$\int_0^1 \Gamma(3-\beta)\, {}_0^C D_t^\beta u(x,t)\, d\beta = -\frac{\partial u(x,t)}{\partial x} + \frac{1}{2}\, {}_0^C D_x^\alpha u(x,t) + \frac{1}{2}\, {}_x^C D_1^\alpha u(x,t) + g(x,t)$$

with $1 \leq \alpha < 2$ and boundary conditions

$$u(x,t)|_{\partial\Omega} = 0; \quad \Omega = [0,1],\ t \in [0, \mathsf{T}],$$

and initial condition $u(x,0) = 0$. If we assume that the exact solution that satisfies the equation is $u(x,t) = t^2 x^2 (1-x)^2$, then

$$g(x,t) = g_1(x,t) + g_2(x,t) + g_3(x,t) + g_4(x,t),$$

where

$$\begin{aligned} g_1(x,t) &= \frac{2x^2(1-x)^2(t^2-t)}{\ln t}, \\ g_2(x,t) &= 2x^2(1-x)(1-2x)t^2, \\ g_3(x,t) &= -\frac{x^{2-\alpha}t^2[(3-\alpha)(4-\alpha)-6(4-\alpha)x+12x^2]}{\Gamma(5-\alpha)}, \\ g_4(x,t) &= -\frac{(1-x)^{2-\alpha}t^2[(3-\alpha)(4-\alpha)-6(4-\alpha)(1-x)+12(1-x)^2]}{\Gamma(5-\alpha)}. \end{aligned}$$

Tables 4.1, 4.2 and 4.3 show the maximum absolute errors and computational time (in seconds) for Example 4.1 for different values of N_x, N_t and Q respectively. In Table 4.1, we present the order of convergence with respect to change in N_x for $\alpha = 1.2$ and 1.8, and we can see the geometric rate of convergence of the numerical scheme. The results show that the method is computationally accurate and efficient. In Tables 4.1 and 4.2, we can see that accuracy improves as the number of terms (both space and time) in the Chebyshev expansion increases.

	$\alpha = 1.2$			$\alpha = 1.8$		
N_x	$\lVert E\rVert_\infty$	Order	CPU time (secs)	$\lVert E\rVert_\infty$	Order	CPU time (secs)
2	0.04356	–	1.2176	0.01581	–	1.3592
4	0.00809	2.4288	1.1686	0.00689	1.1983	1.6072
8	9.0135×10^{-4}	3.1659	1.4856	0.00054	3.6735	1.6421
16	2.8783×10^{-5}	6.7067	1.6756	1.1027×10^{-5}	5.6138	2.2679

Table 4.1: Maximum errors, order of convergence and CPU time for different values of N_x with $\alpha = 1.2$, $\alpha = 1.8$, $N_t = 10$, $Q = 32$ and $T = 0.99$ for Example 4.1.

	$\alpha = 1.3$		$\alpha = 1.9$	
N_t	$\lVert E\rVert_\infty$	CPU time (secs)	$\lVert E\rVert_\infty$	CPU time (secs)
2	0.00278	0.0778	0.00184	0.0654
4	5.4704×10^{-5}	0.2430	4.2668×10^{-5}	0.1729
6	5.3122×10^{-5}	0.2276	4.1648×10^{-5}	0.2679
8	5.3082×10^{-5}	0.7049	4.1622×10^{-5}	0.6553
10	5.3146×10^{-5}	1.1849	4.1654×10^{-5}	1.1606

Table 4.2: Maximum errors and CPU time for different values of N_t with $\alpha = 1.3$, $\alpha = 1.9$, $N_x = 10$, $Q = 32$ and $T = 0.99$ for Example 4.1.

However, in Table 4.3, it can be seen that with 2 equal subintervals in Simpson's rule, accurate results are obtained. Obviously, it is expected that computation would become more expensive as the number of intervals in the quadrature formula and number of spatial and temporal grid points increase. This can be seen from the increasing CPU time as the quantities (N_x, N_t, Q) increase. Figure 4.1 shows the exact and numerical solutions of Example 4.1 at $t = 1.5$ and 2. It can be noted that the exact solutions are in agreement with the numerical approximations.

	$\alpha = 1.3$		$\alpha = 1.9$	
Q	$\|E\|_\infty$	CPU time (secs)	$\|E\|_\infty$	CPU time (secs)
2	5.3148×10^{-5}	0.1311	4.1692×10^{-5}	0.1589
4	5.3146×10^{-5}	0.1904	4.1656×10^{-5}	0.2746
8	5.3146×10^{-5}	0.3028	4.1654×10^{-5}	0.3129
16	5.3146×10^{-5}	0.5816	4.1654×10^{-5}	0.8234
32	5.3146×10^{-5}	1.2644	4.1654×10^{-5}	1.0751

Table 4.3: Maximum errors and CPU time for different values of Q with $\alpha = 1.3$, $\alpha = 1.9$, $N_x = 10$, $N_t = 10$ and $T = 0.99$ for Example 4.1.

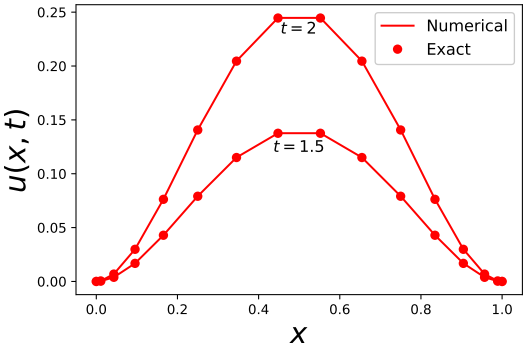


Figure 4.1: Numerical and exact solutions of Example 4.1 at $t = 1.5$, 2 with $N_t = N_x = 15$, $Q = 32$ and $\alpha = 1.5$.

Example 4.2. Consider the two-sided space fractional diffusion equation with time fractional distributed order

$$\int_0^2 \frac{\Gamma(6-\beta)}{120}\, {}_0^CD_t^\beta u(x,t)\,d\beta = \frac{1}{2}\,{}_0^CD_x^\alpha u(x,t) + \frac{1}{2}\,{}_x^CD_1^\alpha u(x,t) + g(x,t), \qquad 1 \le \alpha \le 2$$

with exact solution $u(x,t) = t^5 x^2 (1-x)^2$, such that the boundary conditions are given as

$$u(x,t)|_{\partial\Omega} = 0; \qquad \Omega = [0,1],\ t \in [0, \mathsf{T}],$$

initial conditions given by

$$u(x,0) = u_t(x,0) = 0, \quad x \in \Omega,$$

and

$$g(x,t) = g_1(x,t) + g_2(x,t) + g_3(x,t), \quad g_1(x,t) = \frac{x^2(1-x)^2(t^5-t^3)}{\ln t},$$
$$g_2(x,t) = -\frac{x^{2-\alpha}t^5[(3-\alpha)(4-\alpha) - 6(4-\alpha)x + 12x^2]}{\Gamma(5-\alpha)},$$
$$g_3(x,t) = -\frac{(1-x)^{2-\alpha}t^5[(3-\alpha)(4-\alpha) - 6(4-\alpha)(1-x) + 12(1-x)^2]}{\Gamma(5-\alpha)}.$$

In Tables 4.4, 4.5 and 4.6, we present the maximum error and computational time for Example 4.2 for different values of N_x, N_t and Q respectively, and the convergence order is presented in Table 4.4.

	$\alpha = 1.2$			$\alpha = 1.8$		
N_x	$\|E\|_\infty$	Order	CPU time (secs)	$\|E\|_\infty$	Order	CPU time (secs)
2	0.04212	–	1.2820	0.01532	–	1.3556
4	0.00783	2.4293	1.2753	0.00649	1.2391	1.1975
8	6.2391×10^{-4}	3.6478	1.2452	0.00063	3.3648	1.5205
16	1.0245×10^{-5}	5.9283	1.5589	1.0838×10^{-5}	5.8612	1.5520

Table 4.4: Maximum errors, order of convergence and CPU time for different values of N_x with $\alpha = 1.2$, $\alpha = 1.8$, $N_t = 10$, $Q = 32$ and $T = 0.99$ for Example 4.2.

	$\alpha = 1.2$		$\alpha = 1.8$	
N_t	$\|E\|_\infty$	CPU time (secs)	$\|E\|_\infty$	CPU time (secs)
2	0.04700	0.2015	0.01346	0.0658
4	0.00239	0.1227	0.00093	0.1107
6	8.1716×10^{-5}	0.4807	0.00014	0.1845
8	8.1689×10^{-5}	0.4049	0.00014	0.4408
10	8.1689×10^{-5}	0.8637	0.00014	0.9845

Table 4.5: Maximum errors and CPU time for different values of N_t with $\alpha = 1.2$, $\alpha = 1.8$, $N_x = 10$, $Q = 32$ and $T = 1.2$ for Example 4.2.

The results show that the numerical method gives accurate results. The computational efficiency of the numerical scheme in terms of CPU time is also given. The maximum computational time is approximately $1.2s$, which is the time required to solve the fractional differential equation with $N_x = N_t = 10$ terms of the shifted Chebyshev polynomials and approximate the distributed order with 32 intervals of Simpson's rule. In Figure 4.2, the numerical and exact solutions of Example 4.2 for $t = 0.8,\ 1.36,\ 1.6$ are shown to be in agreement.

	$\alpha = 1.2$		$\alpha = 1.8$	
Q	$\|E\|_\infty$	CPU time (secs)	$\|E\|_\infty$	CPU time (secs)
2	8.1659×10^{-5}	0.1626	0.00014	0.1526
4	8.1684×10^{-5}	0.1766	0.00014	0.2244
8	8.1689×10^{-5}	0.2683	0.00014	0.2773
16	8.1689×10^{-5}	0.5156	0.00014	0.8082

Table 4.6: Maximum errors and CPU time for different values of Q with $\alpha = 1.2$, $\alpha = 1.8$, $N_x = 10$, $N_t = 10$ and $T = 1.2$ for Example 4.1.

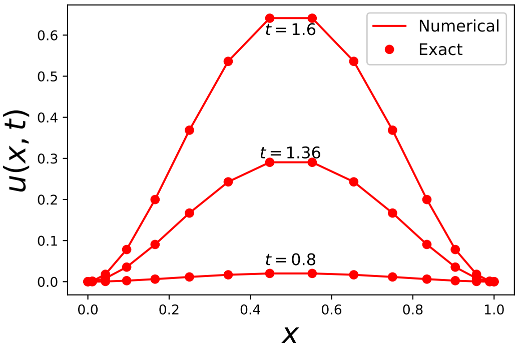


Figure 4.2: Approximate and exact solutions of Example 4.2 at $t = 0.8,\ 1.36,\ 1.6$ with $N_t = 16$, $N_x = 15$, $Q = 32$ and $\alpha = 1.4$.

Example 4.3. In this example, we consider a two dimensional time distributed order two-sided space fractional diffusion equation with constant coefficients

$$\int_0^1 \Gamma(3-\beta)\, {}_0^C D_t^{\beta} u \, d\beta = \frac{1}{2}\left[{}_0^C D_x^{\alpha_1} u + {}_x^C D_1^{\alpha_1} u + {}_0^C D_y^{\alpha_2} u + {}_y^C D_1^{\alpha_2} u\right] + g(x,y,t)$$

whose exact solution is assumed to be $u(x,y,t) = t^2 x^2 (1-x)^2 y^2 (1-y)^2$, such that the boundary condition is given as

$$u(x,y,t)|_{\partial\Omega} = 0,$$

and initial condition

$$u(x,y,0) = 0,$$

where Ω is the rectangular domain $[0,1] \times [0,1]$. In this case the function

$$g(x,y,t) = g_1(x,y,t) + g_2(x,y,t) + g_3(x,y,t) + g_4(x,y,t) + g_5(x,y,t),$$

where

$$
\begin{aligned}
g_1(x,y,t) &= \frac{2x^2(1-x)^2y^2(1-y)^2(t^2-t)}{\ln t},\\
g_2(x,y,t) &= -\frac{x^{2-\alpha_1}t^2y^2(1-y)^2[(3-\alpha_1)(4-\alpha_1)-6(4-\alpha_1)x+12x^2]}{\Gamma(5-\alpha_1)},\\
g_3(x,y,t) &= -\frac{(1-x)^{2-\alpha_1}t^2y^2(1-y)^2[(3-\alpha_1)(4-\alpha_1)-6(4-\alpha_1)(1-x)+12(1-x)^2]}{\Gamma(5-\alpha_1)},\\
g_4(x,y,t) &= -\frac{y^{2-\alpha_2}t^2x^2(1-x)^2[(3-\alpha_2)(4-\alpha_2)-6(4-\alpha_2)y+12y^2]}{\Gamma(5-\alpha_2)},\\
g_5(x,y,t) &= -\frac{(1-y)^{2-\alpha_2}t^2x^2(1-x)^2[(3-\alpha_2)(4-\alpha_2)-6(4-\alpha_2)(1-y)+12(1-y)^2]}{\Gamma(5-\alpha_2)}.
\end{aligned}
$$

In Table 4.7, we show the dynamics of the absolute error norms when the length of the time domain $[0,\mathsf{T}]$ is varied. The number of grid points in the spatial variables (N_x and N_y) are kept constant at $N_x = N_y = 15$ and the temporal grid points at $N_t = 10$. We used 32 equal subintervals for the quadrature formula. We also report the computational time, which was used as a measure of the efficiency of the numerical scheme.

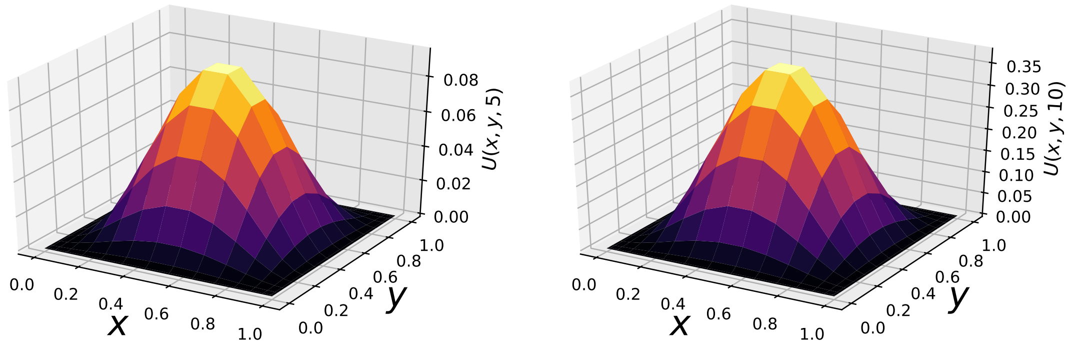
	$\alpha_1=\alpha_2=1.5$		$\alpha_1=\alpha_2=1.9$	
T	$\|E\|_\infty$	CPU time (secs)	$\|E\|_\infty$	CPU time (secs)
0.5	5.0793×10^{-7}	3.3520	5.3510×10^{-8}	3.3909
1.2	3.1589×10^{-6}	3.3291	3.1941×10^{-7}	3.4827
1.9	8.1299×10^{-6}	3.4079	8.1058×10^{-7}	3.5286
2.6	1.5442×10^{-5}	3.4996	1.5284×10^{-6}	3.4398
3.3	2.5106×10^{-5}	3.4538	2.4736×10^{-6}	3.3391
4.0	3.7131×10^{-5}	3.4428	3.6467×10^{-6}	3.4458

Table 4.7: Absolute error norms and CPU time obtained from the approximation of Example 4.3 for different time intervals $[0,\mathsf{T}]$ with $N_x = N_y = 15$, $N_t = 10$ and $Q = 32$.

Table 4.7 shows that the orders of magnitude of the errors are small indicating that accurate results are obtained for this example. However, the accuracy diminishes as the length of the time interval $[0,\mathsf{T}]$ increases. Although, the accuracy of the scheme diminishes for increasing length of time interval, the effect on CPU time is negligible, putting the CPU time at approximately 3.4142 s and 3.4378 s for $\alpha_1 = \alpha_2 = 1.5$ and $\alpha_1 = \alpha_2 = 1.9$ respectively. In Table 4.8, we present the infinity error norm, convergence order and computational time with respect to change in N_x and N_y for $\alpha_1 = \alpha_2 = \{1.2, 1.8\}$ for Example 4.3. Figure 4.3 shows the numerical solution of Example 4.3 for $t = 5$, 10 and $\alpha_1 = \alpha_2 = 1.5$ with $N_x = N_y = 15$, $N_t = 10$ and $Q = 32$. The corresponding error distribution over the spatial domain $[0,1] \times [0,1]$ are shown in Figure 4.4. These surface plots show that the numerical scheme is accurate on the entire spatial domain.

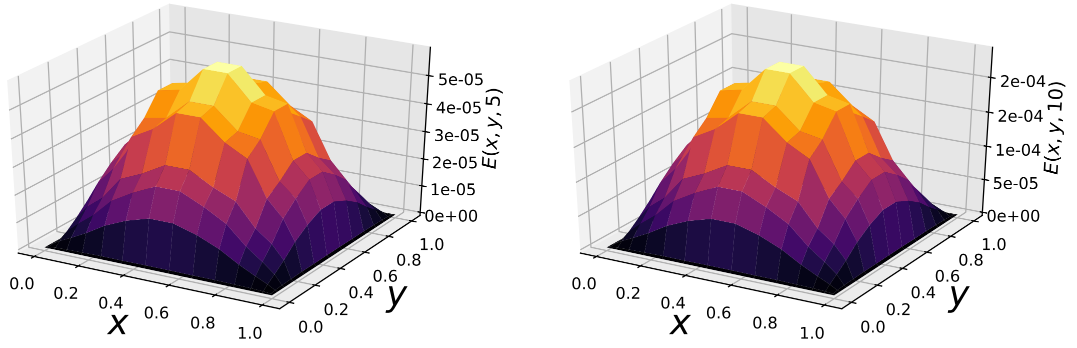
	$\alpha_1 = \alpha_2 = 1.2$			$\alpha_1 = \alpha_2 = 1.8$		
$N_x = N_y$	$\|E\|_\infty$	Order	CPU time (secs)	$\|E\|_\infty$	Order	CPU time (secs)
2	0.00298	–	1.1017	0.00105	–	1.2632
4	0.00059	2.3365	1.3942	0.00033	1.6699	1.6317
8	7.3152×10^{-5}	3.0117	1.3342	1.2885×10^{-5}	4.6787	1.7981
16	9.8584×10^{-7}	6.2134	3.8784	3.5353×10^{-7}	5.1877	5.5946

Table 4.8: Maximum errors, order of convergence and CPU time for different values of N_x, N_y with $\alpha_1 = \alpha_2 = 1.2$, $\alpha = 1.8$, $N_t = 10$, $Q = 32$ and $T = 0.99$ for Example 4.3.



(a) $t = 5$ (b) $t = 10$

Figure 4.3: Approximate solutions obtained for Example 4.3 for $\alpha_1 = \alpha_2 = 1.5$, $N_x = N_y = 15$ and $N_t = 10$ at different values of t.



(a) $t = 5$ (b) $t = 10$

Figure 4.4: Error distribution for Example 4.3 on the spatial domain $[0,1] \times [0,1]$ for $\alpha_1 = \alpha_2 = 1.5$ with $N_x = N_y = 15$ and $N_t = 10$ at different values of t.

Example 4.4. Consider the two dimensional time distributed order two-sided space fractional diffusion equation

$$\int_0^1 \Gamma(7/2-\beta)\,{}_0^CD_t^{\beta} u\,d\beta = \frac{1}{2}\left[{}_0^CD_x^{\alpha_1}u + {}_x^CD_1^{\alpha_1}u + {}_0^CD_y^{\alpha_2}u + {}_y^CD_1^{\alpha_2}u\right] + g(x,y,t)$$

with exact solution $u(x,y,t) = t^{\frac{5}{2}}x^2(1-x)^2y^2(1-y)^2$. The initial condition is given as

$$u(x,y,0) = 0, \quad x,y \in [0,1]\times[0,1],$$

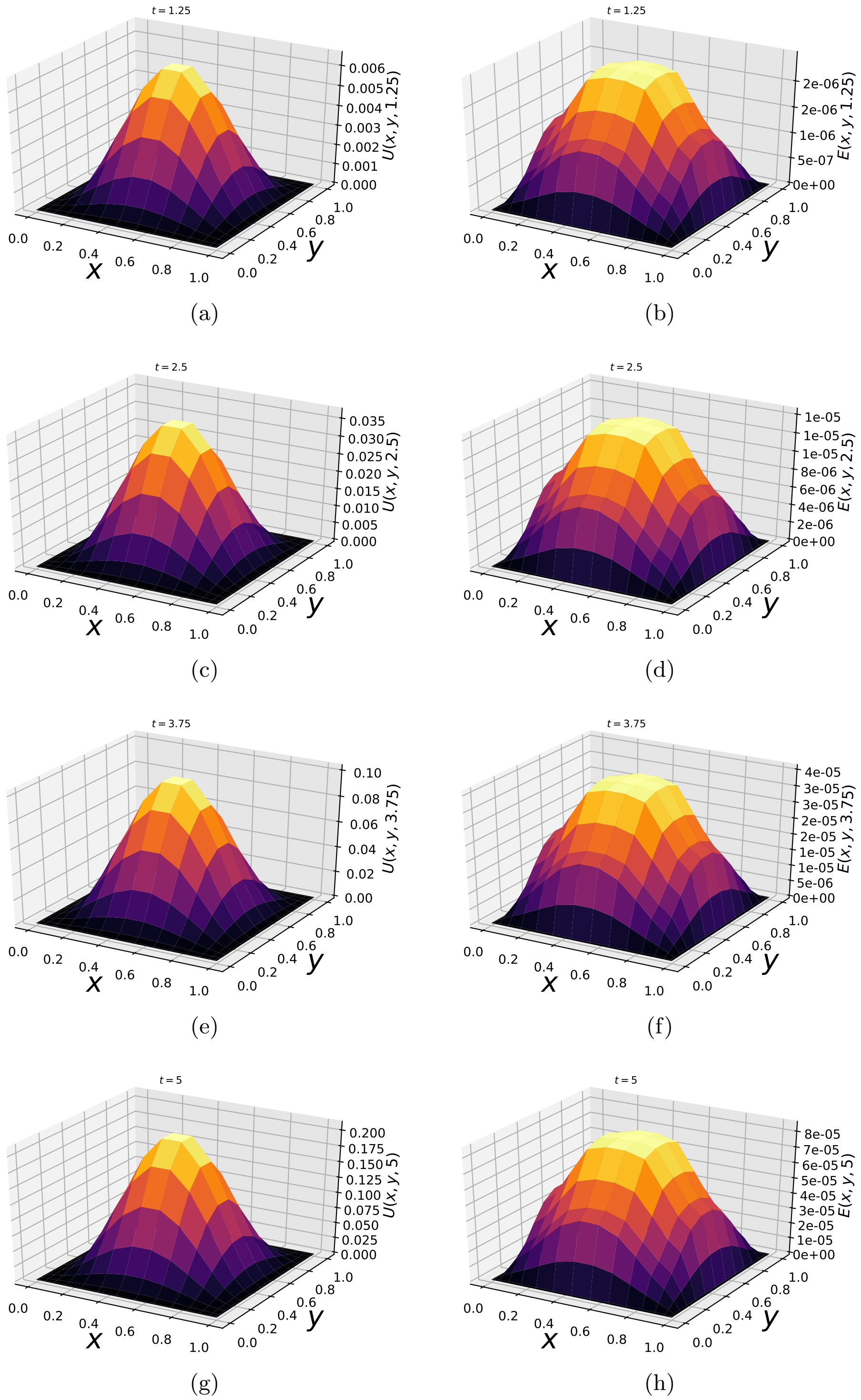
and boundary conditions as

$$u(x,y,t)|_{\partial\Omega} = 0, \quad t \in [0,\mathsf{T}],\ \Omega = [0,1]\times[0,1].$$

The function $g(x,y,t)$ is defined by

$$\begin{aligned}
g(x,y,t) &= g_1(x,y,t)+g_2(x,y,t)+g_3(x,y,t)+g_4(x,y,t)+g_5(x,y,t),\\
g_1(x,y,t) &= \frac{15\sqrt{\pi}x^2(1-x)^2y^2(1-y)^2(t-1)t^{\frac{3}{2}}}{8\ln t},\\
g_2(x,y,t) &= -\frac{x^{2-\alpha_1}t^{\frac{5}{2}}y^2(1-y)^2[(3-\alpha_1)(4-\alpha_1)-6(4-\alpha_1)x+12x^2]}{\Gamma(5-\alpha_1)},\\
g_3(x,y,t) &= -\frac{(1-x)^{2-\alpha_1}t^{\frac{5}{2}}y^2(1-y)^2[(3-\alpha_1)(4-\alpha_1)-6(4-\alpha_1)(1-x)+12(1-x)^2]}{\Gamma(5-\alpha_1)},\\
g_4(x,y,t) &= -\frac{y^{2-\alpha_2}t^{\frac{5}{2}}x^2(1-x)^2[(3-\alpha_2)(4-\alpha_2)-6(4-\alpha_2)y+12y^2]}{\Gamma(5-\alpha_2)},\\
g_5(x,y,t) &= -\frac{(1-y)^{2-\alpha_2}t^{\frac{5}{2}}x^2(1-x)^2[(3-\alpha_2)(4-\alpha_2)-6(4-\alpha_2)(1-y)+12(1-y)^2]}{\Gamma(5-\alpha_2)}.
\end{aligned}$$

Figure 4.5 shows the evolution of the numerical solutions and corresponding error distribution on the spatial domain for Example 4.4 for different time levels with $\alpha_1 = \alpha_2 = 1.4$, $N_x = N_y = 15$, $N_t = 12$ and $Q = 32$. The surface plots of the error on the domain $x,y \in [0,1]\times[0,1]$ show that the numerical solutions are accurate at every points of x and y for all values of t presented. Table 4.9 shows the variation of the maximum error norm for different length of time interval $[0,\mathsf{T}]$. The computational time for each time interval are also obtained. The table shows that the numerical results are accurate for all time interval considered. However, as in the case of Example 4.3 (Table 4.7), the order of magnitude of the errors reduces as the length of the time interval increases, but the effect on computational time is trivial. The CPU times are respectively averaged at 3.3675s and 3.4654s for $\alpha_1 = \alpha_2 = 1.2$ and $\alpha_1 = \alpha_2 = 1.7$. In Table 4.10, we present the error norm, order of convergence and computational time with respect to change in N_x and N_y for $\alpha_1 = \alpha_2 = \{1.2, 1.8\}$ for Example 4.4.



(a) (b)

(c) (d)

(e) (f)

(g) (h)

Figure 4.5: Graphs of numerical solution and error distribution on the spatial domain $[0, 1] \times [0, 1]$ for Example 4.4 with $\alpha_1 = \alpha_2 = 1.4$, $N_x = N_y = 15$ and $N_t = 12$ at different values of t.

	$\alpha_1 = \alpha_2 = 1.2$		$\alpha_1 = \alpha_2 = 1.7$	
T	$\|E\|_\infty$	CPU time (secs)	$\|E\|_\infty$	CPU time (secs)
0.5	1.6666×10^{-7}	3.4308	2.1362×10^{-7}	3.5734
1.2	1.8109×10^{-6}	3.2503	2.1005×10^{-6}	3.5425
1.9	6.1596×10^{-6}	3.3480	6.8529×10^{-6}	3.2892
2.6	1.4079×10^{-5}	3.4857	1.5287×10^{-5}	3.4667
3.3	2.6287×10^{-5}	3.1785	2.8072×10^{-5}	3.2513
4.0	4.3405×10^{-5}	3.5116	4.5789×10^{-5}	3.6692

Table 4.9: Maximum error norms and CPU time obtained from the approximation of Example 4.4 for different time intervals $[0, \mathsf{T}]$ with $N_x = N_y = 15$, $N_t = 10$ and $Q = 32$.

	$\alpha_1 = \alpha_2 = 1.2$			$\alpha_1 = \alpha_2 = 1.8$		
$N_x = N_y$	$\|E\|_\infty$	Order	CPU time (secs)	$\|E\|_\infty$	Order	CPU time (secs)
2	0.00279	–	1.1691	0.00100	–	1.2043
4	0.00059	2.2415	1.4967	0.00033	1.5995	1.8864
8	6.8054×10^{-5}	3.1159	1.5094	6.2492×10^{-5}	4.7234	1.3505
16	8.3009×10^{-7}	6.3573	4.2552	7.0864×10^{-7}	6.4625	4.0479

Table 4.10: Maximum errors, order of convergence and CPU time for different values of N_x, N_y with $\alpha_1 = \alpha_2 = 1.2$, $\alpha = 1.8$, $N_t = 10$, $Q = 32$ and $T = 0.99$ for Example 4.4.

Example 4.5. Consider a one-dimensional time distributed two-sided space fractional differential equation

$$\int_0^1 \Gamma(7/2 - \beta)\, {}_0^C D_t^\beta u \, d\beta = {}_0^C D_x^{\alpha_1} u + {}_x^C D_2^\alpha u + g(x,t)$$

with a non-smooth exact solution $u(x,t) = (xt(2-x))^{5/2}$, and initial and boundary conditions are given respectively as

$$u(x,0) = 0, \quad x \in [0,2] \quad \text{and} \quad u(x,t)|_{\partial\Omega} = 0, \quad t \in [0, \mathsf{T}], \ \Omega = [0,2].$$

In Example 4.5, we assume that the analytical solution $u(x,t) = ((2x - x^2)t)^{5/2} \notin C^3([0,2])$, which is a three times continuously differentiable function at the spatial endpoints. Table 4.11 shows the maximum errors, convergence order and computational time for different values of N_t for Example 4.5, while Table 4.12 presents the maximum error for different values of N_x. Although, the solution does not satisfy the smoothness assumption in the spatial dimension, the numerical result is computationally accurate. The numerical scheme for Example 4.5 can also be extended to higher dimensional two-sided space time distributed fractional differential equations.

	$\alpha = 1.2$			$\alpha = 1.8$		
N_t	$\|E\|_\infty$	Order	CPU time (secs)	$\|E\|_\infty$	Order	CPU time (secs)
4	0.00467	–	0.1237	0.00296	–	1.1336
5	0.00181	4.2476	0.3949	0.00094	5.1405	0.1776
6	0.00084	4.2106	0.3899	0.00044	4.1636	0.2972
7	0.00044	4.1948	0.5469	0.00027	3.1680	0.4199
8	0.00025	4.2336	0.6124	0.00016	3.9185	0.5865
9	0.00015	4.3370	0.8757	0.00010	3.9904	0.8609
10	9.8735×10^{-5}	3.9692	1.2008	6.5021×10^{-5}	4.0856	1.6037
11	6.7162×10^{-5}	4.0429	1.6186	4.3713×10^{-5}	4.1660	2.4175

Table 4.11: Maximum errors, order of convergence and CPU time for different values of N_t with $\alpha = \{1.2, 1.8\}$, $N_x = 10$, $Q = 32$ and $T = 2$ for Example 4.5.

	$\alpha = 1.2$		$\alpha = 1.8$	
N_x	$\|E\|_\infty$	CPU time (secs)	$\|E\|_\infty$	CPU time (secs)
4	1.1209×10^{-4}	1.3893	6.5962×10^{-5}	1.1549
6	9.6827×10^{-5}	1.0312	6.4739×10^{-5}	1.1249
8	9.8545×10^{-5}	1.0692	6.5015×10^{-5}	1.1031
10	9.8735×10^{-5}	1.1119	6.5021×10^{-5}	1.0732

Table 4.12: Maximum errors and CPU time for different values of N_x with $\alpha = \{1.2, 1.8\}$, $N_t = 10$, $Q = 32$ and $T = 2$ for Example 4.5.

5. Conclusion

Fractional partial differential equations are a special and important class of differential equations. It is well known that fractional differential equation with distributed order is a robust mathematical instrument for describing complex dynamical systems, especially systems that occur at multiscale stages. Thus, constructing accurate numerical schemes for this class of differential equations is essential. In this study, we have presented a numerical method for the discretization of a general time distributed order two-sided multi-dimensional fractional differential equation. The time distributed order fractional differential operator was approximated using a Newton–Cotes formula. We presented both the left and right sided fractional differentiation matrices using approximation in terms of shifted Chebyshev polynomials and shifted Chebyshev–Gauss–Lobatto quadrature. The convergence of the numerical scheme for the aforementioned class of differential equation was shown. We demonstrated the numerical discretization method on some one- and two-dimensional time distributed order two-sided space fractional differential equations such as the fractional advection-dispersion and fractional diffusion equations. The numerical results showed that the method is computationally efficient and accurate.

References

[1] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions, with Formulas, Graphs, and Mathematical Tables*, Third printing, with corrections National Bureau of Standards Applied Mathematics Series **55**, Superintendent of Documents, U.S. Government Printing Office, Washington, D.C., 1965.

[2] T. M. Atanacković, S. Pilipović, B. Stanković and D. Zorica, *Fractional Calculus with Applications in Mechanics: Wave propagation, impact and variational principles*, John Wiley & Sons, New York, 2014.

[3] L. Beghin, *Fractional diffusion-type equations with exponential and logarithmic differential operators*, Stochastic Process. Appl. **128** (2018), no. 7, 2427–2447.

[4] M. Cai and C. Li, *Regularity of the solution to Riesz-type fractional differential equation*, Integral Transforms Spec. Funct. **30** (2019), no. 9, 711–742.

[5] C. Canuto, M. Y. Hussaini, A. Quarteroni and T. A. Zang, *Spectral Methods in Fluid Dynamics*, Springer Series in Computational Physics, Springer-Verlag, New York, 1988.

[6] ______, *Spectral Methods: Fundamentals in single domains*, Scientific Computation, Springer-Verlag, Berlin, 2006.

[7] M. Caputo, *Diffusion of fluids in porous media with memory*, Geothermics **28** (1999), no. 1, 113–130.

[8] E. H. Doha, A. H. Bhrawy and S. S. Ezz-Eldien, *Efficient Chebyshev spectral methods for solving multi-term fractional orders differential equations*, Appl. Math. Model. **35** (2011), no. 12, 5662–5672.

[9] S. Guo, L. Mei, Z. Zhang and Y. Jiang, *Finite difference/spectral-Galerkin method for a two-dimensional distributed-order time-space fractional reaction-diffusion equation*, Appl. Math. Lett. **85** (2018), 157–163.

[10] X. Hu, F. Liu, I. Turner and V. Anh, *An implicit numerical method of a new time distributed-order and two-sided space-fractional advection-dispersion equation*, Numer. Algorithms **72** (2016), no. 2, 393–407.

[11] G. Jumarie, *New stochastic fractional models for Malthusian growth, the Poissonian birth process and optimal management of populations*, Math. Comput. Modelling **44** (2006), no. 3-4, 231–254.

[12] X. Y. Li and B. Y. Wu, *A numerical method for solving distributed order diffusion equations*, Appl. Math. Lett. **53** (2016), 92–99.

[13] K. S. Miller and B. Ross, *An Introduction to the Fractional Calculus and Fractional Differential Equations*, A Wiley-Interscience Publication, John Wiley & Sons, New York, 1993.

[14] M. L. Morgado and M. Rebelo, *Numerical approximation of distributed order reaction-diffusion equations*, J. Comput. Appl. Math. **275** (2015), 216–227.

[15] M. L. Morgado, M. Rebelo, L. L. Ferrás and N. J. Ford, *Numerical solution for diffusion equations with distributed order in time using a Chebyshev collocation method*, Appl. Numer. Math. **114** (2017), 108–123.

[16] K. B. Oldham and J. Spanier, *The Fractional Calculus: Theory and applications of differentiation and integration to arbitrary order*, Mathematics in Science and Engineering **111**, Academic Press, New York, 1974.

[17] S. D. Oloniiju, S. P. Goqo and P. Sibanda, *A Chebyshev spectral method for heat and mass transfer in MHD nanofluid flow with space fractional constitutive model*, Front. Heat Mass Transf. **13** (2019), 8 pp.

[18] ———, *A Chebyshev pseudo-spectral method for the multi-dimensional fractional Rayleigh problem for a generalized Maxwell fluid with Robin boundary conditions*, Appl. Numer. Math. **152** (2020), 253–266.

[19] I. Podlubny, *Fractional Differential Equations: An introduction to fractional derivatives, fractional differential equations, to methods of their solution and some of their applications*, Mathematics in Science and Engineering **198**, Academic Press, San Diego, CA, 1999.

[20] W. R. Schneider and W. Wyss, *Fractional diffusion and wave equations*, J. Math. Phys. **30** (1989), no. 1, 134–144.

[21] M. Stojanović, *Regularity of solutions to nonlinear time fractional differential equation*, Acta Math. Sci. Ser. B (Engl. Ed.) **33** (2013), no. 6, 1721–1735.

[22] W. Wyss, *The fractional diffusion equation*, J. Math. Phys. **27** (1986), no. 11, 2782–2785.

[23] F. Zeng, F. Liu, C. Li, K. Burrage, I. Turner and V. Anh, *A Crank–Nicolson ADI spectral method for a two-dimensional Riesz space fractional nonlinear reaction-diffusion equation*, SIAM J. Numer. Anal. **52** (2014), no. 6, 2599–2622.

Shina Daniel Oloniiju, Sicelo Praisegod Goqo and Precious Sibanda
School of Mathematics, Statistics and Computer Science, University of KwaZulu-Natal,
Private Bag X01, Scottsville, Pietermaritzburg, 3209, South Africa
E-mail addresses: shina@aims.edu.gh, goqos@ukzn.ac.za, sibandap@ukzn.ac.za