CHAPTER 2

EFFICIENT NUMERICAL ALGORITHMS FOR RIESZ-SPACE FRACTIONAL PARTIAL DIFFERENTIAL EQUATIONS BASED ON FINITE DIFFERENCE/OPERATIONAL MATRIX

In this chapter, we have developed two efficient numerical algorithms to find the numerical solutions of Riesz fractional diffusion equations (RFDE) and RFADEs. For this purpose, we have constructed the numerical schemes by applying a finite difference scheme based on MTM in spatial direction and a meshfree OMM based on shifted Legendre polynomials in the time direction. To the best of our knowledge, this approach has not been applied so far to solve proposed RFDEs and RFADEs. In Section 2.2, we discuss the MTM for Riesz derivatives and function approximation by operational matrix approach. In Section 2.3, we give the construction of the numerical schemes for RFDE and RFADE. Optimal error bounds for the approximation are discussed in Section 2.4. In Section 2.5, the numerical stability has

been verified and two numerical examples of RFDE and RFADE are presented to show the effectiveness and accuracy of the proposed schemes followed by concluding remarks.

2.1 Introduction

In this chapter, we consider the following space fractional partial differential equation with Riesz-space fractional derivative

$$\frac{\partial}{\partial t}\mathrm{u}(x,t) = k_\alpha \frac{\partial^\alpha}{\partial |x|^\alpha}\mathrm{u}(x,t) + k_\beta \frac{\partial^\beta}{\partial |x|^\beta}\mathrm{u}(x,t), \quad 0 \le t \le T, \quad 0 \le x \le L, \tag{2.1}$$

with initial condition

$$\mathrm{u}(x,0) = f(x), \tag{2.2}$$

and Dirichlet boundary conditions

$$\mathrm{u}(0,t) = \mathrm{u}(L,t) = 0, \tag{2.3}$$

where, $1 < \alpha \le 2$, $0 < \beta < 1$, u is a solute concentration; k_α and k_β represent the dispersion coefficient and the average fluid velocity, respectively. Here, we take $k_\alpha > 0$ and $k_\beta \ge 0$. When $k_\beta = 0$, (2.1) reduces to RFDE otherwise, it is called RFADE.

In order to find the numerical solution of the proposed model (2.1)-(2.3), we have approximated the Riesz derivatives by using MTM, which converts the (2.1) into a system of time ordinary differential equations (TODEs). Yang et al. [72] used differential algebraic system solver (DASSL) to solve the system of TODEs. Joubert

et al. questions the credibility of ODE solvers in his paper [163] and suggested that the blind trust on these ODE solvers is not always suitable for every problem. The authors also gave the justification that for a large scale of time, these solvers are not reliable. The disadvantages of these ODE solvers also motivated us to develop this numerical approach to solve the system of TODEs without using any ODE solvers, which will work for a large scale of time. Therefore, we apply OMM based on shifted Legendre polynomials for approximating the time derivative.

The advantages of this proposed numerical schemes are manifold. The MTM leads to a system of TODEs with a spatial discretization matrix raised to fractional order and provides the best approximation to the analytical solution [72]. The choice of shifted Legendre polynomials is due to their orthogonal properties. It also saves the computational cost in calculating the unknown Legendre's coefficients. Furthermore, the operational matrix of integration for shifted Legendre polynomials is sparse in nature, which eases the transformation of the system of TODEs to a system of linear algebraic equations. It is meshfree in the time domain, and the numerical solution does not depend on the previous time levels. Therefore, one does not have to store the numerical value at any time level. We have observed that the convergence rate of the proposed numerical scheme is of second order in spatial direction for all values of $\alpha \in (1, 2)$ and derived the optimal error bound for the approximate solution. The accuracy of the schemes also depends on the optimum number of basis functions. The stability of the numerical scheme has also been verified numerically by introducing some noisy data in the initial condition. The effectiveness and accuracy of the developed numerical schemes have been tested on some numerical experiments. It is found that the schemes are simple, easy to implement and yield high accurate results.

2.2 Matrix transform method and the operational matrix

In this section, we discuss about the MTM for Riesz-space fractional derivative and function approximation by shifted Legendre polynomials. We have also derived the operational matrix of integration for solving the problem (2.1).

2.2.1 Matrix transform method for Riesz space fractional derivative

The MTM, for space fractional diffusion equation, is proposed by Illic et al. [71]. It says that the Riesz fractional derivative is equivalent to the fractional power of the Laplacian operator under Dirichlet boundary conditions. It provides the best approximation to the analytic solution.

Let $x_i = ih$, $i = 0, 1, 2, ..., N$ be the number of grid points in space direction and $h = L/N$ be the step size in space, where N is the total number of grid points and L is the length of the interval.

At first, we consider the following RFDE

$$\frac{\partial}{\partial t}\mathrm{u}(x,t) = k_\alpha \frac{\partial^\alpha}{\partial |x|^\alpha}\mathrm{u}(x,t). \tag{2.4}$$

Using Lemma (1.2.1), we get

$$\frac{\partial}{\partial t}\mathrm{u}(x,t) = -k_\alpha(-\Delta)^{\alpha/2}\mathrm{u}(x,t) = -k_\alpha\left(-\frac{\partial^2}{\partial x^2}\right)^{\alpha/2}\mathrm{u}(x,t). \tag{2.5}$$

Consider the standard diffusion equation

$$\frac{\partial}{\partial t}\mathrm{u}(x,t) = -k_\alpha \left(-\frac{\partial^2}{\partial x^2}\right)\mathrm{u}(x,t). \tag{2.6}$$

Let $\mathrm{u}_i(t) = \mathrm{u}(x_i,t)$, $i = 1,2,...,N-1$, be the numerical value of the $\mathrm{u}(x,t)$ at each grid points. Now, applying central difference formula on the R.H.S. of (2.6), we get

$$\frac{d}{dt}\mathrm{u}_i(t) = -k_\alpha \left(\frac{-\mathrm{u}_{i+1}(t) + 2\mathrm{u}_i(t) - \mathrm{u}_{i-1}(t)}{h^2}\right).$$

Then, the above equation can be written in matrix form as;

$$\frac{d}{dt}\mathrm{U}(t) = -k_\alpha \mathrm{T}_h \mathrm{U}(t) + \mathrm{b}, \tag{2.7}$$

where, $\mathrm{T}_h \in R^{N-1\times N-1}$ and $\mathrm{U}(t), \mathrm{b} \in R^{N-1}$ are given as;

$$\mathrm{T}_h = \frac{1}{h^2}\begin{bmatrix} 2 & -1 & 0 & 0 & \cdots & 0 \\ -1 & 2 & -1 & 0 & \cdots & 0 \\ 0 & -1 & 2 & -1 & \cdots & 0 \\ \vdots & & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & -1 & 2 & -1 \\ 0 & \cdots & \cdots & 0 & -1 & 2 \end{bmatrix}, \quad \mathrm{U}(t) = \begin{bmatrix} \mathrm{u}_1(t) \\ \vdots \\ \vdots \\ \mathrm{u}_{N-1}(t) \end{bmatrix}, \text{ and } \mathrm{b} = \frac{k_\alpha}{h^2}\begin{bmatrix} \mathrm{u}_0 \\ 0 \\ \vdots \\ 0 \\ \mathrm{u}_N \end{bmatrix}. \tag{2.8}$$

If $b = 0$ (using Dirichlet boundary condition), then (2.7) can be written as;

$$\frac{d}{dt}\mathrm{U}(t) = -k_\alpha \mathrm{T}_h \mathrm{U}(t). \tag{2.9}$$

Hence, we can write (2.5) as;

$$\frac{d}{dt}\mathrm{U}(t) = -k_\alpha(\mathrm{T}_h)^{\alpha/2}\mathrm{U}(t), \tag{2.10}$$

Since, the matrix T_h is a symmetric positive definite matrix, therefore, there exist a nonsingular matrix $\mathrm{V} \in R^{N-1\times N-1}$ such that,

$$\mathrm{T}_h = \mathrm{VDV}^T,$$

where, $\mathrm{D} = diag(\lambda_1, \lambda_2, ..., \lambda_{n-1})$ and λ_i, $i = 1, 2, ..., N-1$, be the eigenvalues of matrix T and V is the matrix containing the eigen vectors v_i of T corresponding to eigen values λ_i such that $\mathrm{V} = [\mathrm{v}_1, \mathrm{v}_2, ..., \mathrm{v}_{N-1}]$. Therefore, (2.10) can be written as,

$$\begin{aligned}\frac{d}{dt}\mathrm{U}(t) &= -k_\alpha\left(\mathrm{VD}^{\alpha/2}\mathrm{V}^T\right)\mathrm{U},\\ \frac{d}{dt}\mathrm{U}(t) &= -k_\alpha\mathrm{AU}(t),\end{aligned} \tag{2.11}$$

where $\mathrm{A} = \mathrm{VD}^{\alpha/2}\mathrm{V}^T$ is the symmetric as well as centro-symmetric matrix of order $(N-1)$.

Lemma 2.2.1 ([164]). If T_h is a positive definite matrix and $\mathrm{T}_h = \mathrm{VDV}^T$, where V is the orthogonal matrix then for any arbitrary $\alpha/2$

$$\mathrm{T}_h^{\alpha/2} = \mathrm{V(D)}^{\alpha/2}\mathrm{V}^T, \tag{2.12}$$

and the eigenvalues of the real symmetric matrix $\mathrm{V(D)}^{\alpha/2}\mathrm{V}^T$ are given by;

$$\lambda_i = \left(\frac{4}{h^2}\sin^2\left(\frac{\pi i}{2N}\right)\right)^{\alpha/2}, \quad i = 1, 2, ..., N-1. \tag{2.13}$$

Lemma 2.2.2. If λ_i, $i = 1, 2, ..., N-1$, be the eigen values of the matrix T_h arranged

in such a manner that $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_{N-1}$ and let $\mathrm{v}_i = [\mathrm{v}_{1,i}, \mathrm{v}_{2,i}, ..., \mathrm{v}_{N-1,i}]^T$ be the eigen vectors corresponding to the eigen values λ_i such that $\mathrm{D} = diag[\lambda_1, \lambda_2, ..., \lambda_{N-1}]$ and $\mathrm{V} = [\mathrm{v}_1, \mathrm{v}_2, ..., \mathrm{v}_{N-1}]$ then the product matrix $\mathrm{A} = \left(\mathrm{V}(\mathrm{D})^{\alpha/2}\mathrm{V}^T\right)$ is the symmetric as well as centro-symmetric matrix of order $(N-1)$ and is of the form

$$\mathrm{A} = [a_{ij}] = \begin{bmatrix} a_{1,1} & a_{2,1} & a_{3,1} & \cdots & a_{N-3,1} & a_{N-2,1} & a_{N-1,1} \\ a_{2,1} & a_{2,2} & a_{3,2} & \cdots & a_{N-3,2} & a_{N-2,2} & a_{N-2,1} \\ a_{3,1} & a_{3,2} & a_{3,3} & \cdots & a_{N-3,3} & a_{N-3,2} & a_{N-3,1} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ a_{N-3,1} & a_{N-3,2} & a_{N-3,3} & \cdots & a_{3,3} & a_{3,2} & a_{3,1} \\ a_{N-2,1} & a_{N-2,2} & a_{N-3,2} & \cdots & a_{3,2} & a_{2,2} & a_{2,1} \\ a_{N-1,1} & a_{N-2,1} & a_{N-3,1} & \cdots & a_{3,1} & a_{2,1} & a_{1,1} \end{bmatrix}. \tag{2.14}$$

e.g.: In particular, for $N = 4$ and $N = 5$

$$\mathrm{A} = \begin{bmatrix} a_{11} & a_{21} & a_{31} \\ a_{21} & a_{22} & a_{21} \\ a_{31} & a_{21} & a_{11} \end{bmatrix}, \quad \mathrm{A} = \begin{bmatrix} a_{11} & a_{21} & a_{31} & a_{41} \\ a_{21} & a_{22} & a_{32} & a_{31} \\ a_{31} & a_{32} & a_{22} & a_{21} \\ a_{41} & a_{31} & a_{21} & a_{11} \end{bmatrix}.$$

2.2.2 Function approximation by orthogonal polynomials

2.2.2.1 Shifted Legendre polynomials

The shifted Legendre polynomials defined over the domain $[0, 1]$ are given by;

$$\psi_j(t) = P_j(2t-1), \quad j = 0, 1, 2, 3, ... \tag{2.15}$$

where $P_j(t)$ are the Legendre polynomials of order j defined on the interval $[-1, 1]$ and satisfy the following recursive formula;

$$P_{j+1}(t) = \left(\frac{2j+1}{j+1}\right) tP_j(t) - \left(\frac{j}{j+1}\right) P_{j-1}(t), \quad j = 1, 2, 3, ... \tag{2.16}$$

where $P_0(t) = 1$, $P_1(t) = t$, $P_2(t) = \frac{1}{2}(3t^2 - 1)$ and so on.

The shifted Legendre polynomials are orthogonal with respect to the weight function $w(t) = 1$ such that,

$$\int_0^1 w(t)P_i(t)P_j(t)dt = \begin{cases} \frac{1}{2i+1}\delta_{ij} & \text{for } i = j, \\ 0 & \text{otherwise.} \end{cases} \tag{2.17}$$

2.2.2.2 Shifted Chebyshev polynomial of second kind

The shifted Chebyshev polynomial of second kind on $[0, 1]$ are given by

$$\phi_j(t) = U_j(2t - 1), \quad j = 0, 1, 2, 3, ... \tag{2.18}$$

where $U_j(t)$ are the Chebyshev polynomials of second kind of order j defined on the interval $[-1, 1]$ and satisfy the following recursive formula,

$$U_{j+1}(t) = 2tU_j(t) - U_{j-1}(t), \quad j = 1, 2, 3, ... \tag{2.19}$$

where $U_0(t) = 1$, $U_1(t) = t$, $U_2(t) = 4t^2 - 1$ and so on. The shifted Chebyshev polynomials of second kind are orthogonal with respect to the weight function $w(t) =$

$\sqrt{t-t^2}$ such that

$$\int_0^1 w(t)\phi_i(t)\phi_j(t)dt = \begin{cases} \pi/8 & \text{for } i=j, \\ 0 & \text{otherwise.} \end{cases} \tag{2.20}$$

2.2.2.3 Function approximation

Let $\mathrm{u}(x_i,t)$ be the numerical solution of the RFDE (2.4) at each grid point therefore, we approximate the solution $\mathrm{u}(x_i,t)$ at each grid point x_i for $i=1,2,...,N-1$. For this, let us suppose that the time derivative of $\mathrm{u}(x_i,t)$ can be approximated as

$$\frac{d\mathrm{u}(x_i,t)}{dt} \approx \sum_{j=0}^{M} c_{ij}\psi_j(t) = \mathrm{C}_i^T\Psi(t), \quad i=1,2,...,N-1, \quad j=0,1,...,M. \tag{2.21}$$

Integrating (2.21) with respect to t, we get

$$\begin{aligned} \int_0^t \frac{d\mathrm{u}(x_i,t)}{dt}dt &\approx \int_0^t \mathrm{C}_i^T\Psi(t)dt, \\ \mathrm{u}(x_i,t) &\approx \mathrm{u}(x_i,0) + \mathrm{C}_i^T\mathrm{K}\Psi(t), \\ \mathrm{u}(x_i,t) &\approx f(x_i) + \mathrm{C}_i^T\mathrm{K}\Psi(t), \end{aligned} \tag{2.22}$$

where $f(x_i)$ is the initial condition at each grid point x_i and K is the operational matrix of integration, which we will discuss in the next subsection 2.2.2.4.

Now, approximating $f(x_i)$ as mentioned in (1.29), we get

$$f(x_i) \approx \sum_{j=0}^{M} f_{ij}\psi_j(t) = \mathrm{F}_i^T\Psi(t), \quad i=1,2,...,N-1, \quad j=0,1,...,M, \tag{2.23}$$

Using (2.23) in (2.22), we get

$$\begin{aligned} \mathrm{u}(x_i,t) &\approx \mathrm{F}_i^T\Psi(t) + \mathrm{C}_i^T\mathrm{K}\Psi(t), \\ \mathrm{u}(x_i,t) &\approx \left(\mathrm{F}_i^T + \mathrm{C}_i^T\mathrm{K}\right)\Psi(t), \end{aligned} \tag{2.24}$$

where,

$$\mathrm{F}_i = [f_{ij}] = \begin{bmatrix} f_{i0} \\ f_{i1} \\ \vdots \\ f_{iM} \end{bmatrix}, \quad \mathrm{C}_i = [c_{ij}] = \begin{bmatrix} c_{i0} \\ c_{i1} \\ \vdots \\ c_{iM} \end{bmatrix}, \quad \text{and } \Psi(t) = \begin{bmatrix} \psi_0(t) \\ \psi_1(t) \\ \vdots \\ \psi_M(t) \end{bmatrix}. \tag{2.25}$$

Here C_i's are the unknowns coefficients, which we need to find out. Since f_i's are known, therefore, f_{ij} are calculated by,

$$f_{ij} = \frac{\langle f_i(t), \psi_j(t)\rangle}{\langle \psi_j(t), \psi_j(t)\rangle} = \frac{\int_0^1 w(t) f_i(t)\psi_j(t)dt}{\int_0^1 w(t)\psi_j(t)\psi_j(t)dt}. \tag{2.26}$$

2.2.2.4 Operational matrix of integration

Let $\Psi(t)$ be the shifted Legendre polynomials defined in section 2.2.2, then

$$\int_0^t \Psi(\xi)d\xi = \begin{bmatrix} \int_0^t \psi_0(\xi)d\xi \\ \int_0^t \psi_1(\xi)d\xi \\ \vdots \\ \int_0^t \psi_M(\xi)d\xi \end{bmatrix} = \begin{bmatrix} k_0(t) \\ k_1(t) \\ \vdots \\ k_M(t) \end{bmatrix} = \begin{bmatrix} k_i(t) \end{bmatrix}. \tag{2.27}$$

Now, approximating $k_i(t)$ as mentioned in (1.29)

$$k_i(t) \approx \sum_{j=0}^{M} k_{ij}\psi_j(t) = \begin{bmatrix} k_{00} & k_{01} & \cdots & k_{0M} \\ k_{10} & k_{11} & \cdots & k_{1M} \\ \vdots & & & \\ k_{M0} & k_{M1} & \cdots & k_{MM} \end{bmatrix} \begin{bmatrix} \psi_0(t) \\ \psi_1(t) \\ \vdots \\ \psi_M(t) \end{bmatrix} = \mathrm{K}\Psi(t), \tag{2.28}$$

where K is the operational matrix of integration of order $(M+1)$, whose coefficient k_{ij} are obtained by the formula

$$k_{ij} = \frac{\langle k_i(t), \psi_j(t)\rangle}{\langle \psi_j(t), \psi_j(t)\rangle} = \frac{\int_0^1 w(t)k_i(t)\psi_j(t)dt}{\int_0^1 w(t)\psi_j(t)\psi_j(t)dt}. \tag{2.29}$$

2.3 Construction of numerical scheme for RFDE and RFADE

2.3.1 Numerical scheme for Riesz fractional diffusion equation (RFDE)

In this section, we construct the efficient numerical scheme for the RFDE

$$\frac{\partial}{\partial t}\mathrm{u}(x,t) = k_\alpha \frac{\partial^\alpha}{\partial |x|^\alpha}\mathrm{u}(x,t), \quad 0 \le t \le T,\ 0 \le x \le L, \tag{2.30}$$

with the initial and boundary conditions given by

$$\mathrm{u}(x,0) = f(x), \tag{2.31}$$

$$\mathrm{u}(0,t) = \mathrm{u}(\pi,t) = 0, \tag{2.32}$$

where $1 < \alpha \leq 2$.

After applying the MTM as mentioned in section 2.2.1, the RFDE (2.30)-(2.32) converts in the form

$$\frac{d}{dt}\mathrm{U}(t) = -k_\alpha \mathrm{A}\mathrm{U}(t), \tag{2.33}$$

where $\mathrm{A} = \mathrm{V}(\mathrm{D})^{\alpha/2}\mathrm{V}^T$ is the symmetric as well as centro-symmetric matrix of order $(N-1)$ and $\mathrm{U}(t) = [\mathrm{u}_1(t), \mathrm{u}_2(t), ..., \mathrm{u}_{N-1}(t)]^T$.
We can write the above equation as

$$\frac{d\mathrm{u}_i(t)}{dt} = -k_\alpha \mathrm{A}\mathrm{u}_i(t). \tag{2.34}$$

Let us denote $\mathrm{u}(x_i, t) = \mathrm{u}_i(t)$, $i = 1, 2, ..., N-1$, and use the approximation of $\mathrm{u}_i(t)$, mentioned in (2.21) and (2.24) as

$$\frac{d\mathrm{u}_i(t)}{dt} \approx \mathrm{C}_i^T \Psi(t), \quad \text{and} \quad \mathrm{u}_i(t) \approx \left(\mathrm{F}_i^T + \mathrm{C}_i^T K\right) \Psi(t), \tag{2.35}$$

where F_i, C_i and $\Psi(t)$ are mentioned in (2.25) and K is mentioned in (2.28). Using (2.35) in (2.34), we get

$$[\mathrm{C}_i^T]\Psi(t) = -k_\alpha \mathrm{A}[\mathrm{F}_i^T + \mathrm{C}_i^T \mathrm{K}]\Psi(t). \tag{2.36}$$

The above equation can be written as

$$\begin{aligned} [\mathrm{C}_i^T] &= -k_\alpha \mathrm{A}[\mathrm{F}_i^T + \mathrm{C}_i^T \mathrm{K}], \\ [\mathrm{C}_i^T] &= -k_\alpha \mathrm{A}\mathrm{C}_i^T \mathrm{K} - k_\alpha \mathrm{A}\mathrm{F}_i^T, \end{aligned} \tag{2.37}$$

where,

$$\left[\mathrm{C}_i^T\mathrm{K}\right] = \left[\sum_{j=0}^{M} c_{ij}k_{j0},\ \sum_{j=0}^{M} c_{ij}k_{j1}, \cdots, \sum_{j=0}^{M} c_{ij}k_{jM}\right], \tag{2.38}$$

is a matrix of order $(N-1)(M+1)$.

Now, putting the value of $[\mathrm{F}_i^T]$, $[\mathrm{C}_i^T]$ from (2.25) and $[\mathrm{C}_i^T\mathrm{K}]$ from (2.38) in (2.37), and after rigorous calculations (see 2.6.1), comparing the coefficients of c_{ij} from (2.93), for $i = 1, 2, ..., N-1$ and $j = 0, 1, ..., M$, we get the $(N-1)(M+1)$ a system of algebraic equation, given as

$$c_{ij} = -k_\alpha \sum_{l=1}^{N-1} a_{il} \left(\sum_{m=0}^{M} c_{lm}k_{mj} \right) - k_\alpha \sum_{l=1}^{N-1} a_{il} \left(f_{lj}\right), \quad i = 1, 2, ..., N-1 \ \ j = 0, 1, ..., M, \tag{2.39}$$

where $a_{ij} = a_{ji}$ with $a_{ij} = a_{N-i+1,N-j+1}$ because A is symmetric as well as centro-symmetric matrix. Let '$\otimes$' denotes the kronecker product then, the numerical scheme in matrix form is given by

$$\begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_{N-1} \end{bmatrix} = -k_\alpha \begin{bmatrix} a_{1,1} & a_{2,1} & \cdots & a_{N-1,1} \\ a_{2,1} & a_{2,2} & \cdots & a_{N-2,1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N-1,1} & a_{N-2,1} & \cdots & a_{1,1} \end{bmatrix} \otimes \begin{bmatrix} k_{00} & k_{10} & \cdots & k_{M0} \\ k_{01} & k_{11} & \cdots & k_{M1} \\ \vdots & & & \\ k_{0M} & k_{1M} & \cdots & k_{MM} \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_{N-1} \end{bmatrix}$$

$$- k_\alpha \begin{bmatrix} a_{1,1} & a_{2,1} & \cdots & a_{N-1,1} \\ a_{2,1} & a_{2,2} & \cdots & a_{N-2,1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N-1,1} & a_{N-2,1} & \cdots & a_{1,1} \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_{N-1} \end{bmatrix}, \tag{2.40}$$

or, simply we can write

$$\begin{aligned} \mathrm{C} &= -k_\alpha(\mathrm{A}\otimes\mathrm{K}^T)\mathrm{C} - k_\alpha(\mathrm{A}\otimes\mathrm{I})\mathrm{F}, \\ \mathrm{C} &= -k_\alpha\mathrm{P}\mathrm{C} - k_\alpha\mathrm{Q}\mathrm{F}, \\ [\mathrm{I}+k_\alpha\mathrm{P}]\mathrm{C} &= -k_\alpha[\mathrm{Q}\mathrm{F}], \end{aligned} \tag{2.41}$$

where P and Q are the matrix of order $(N-1)(M+1)\times(N-1)(M+1)$ as given in (2.42) and (2.43).

$$\mathrm{P} = \mathrm{A}\otimes\mathrm{K}^T = \begin{bmatrix} a_{11}\mathrm{K}^T & a_{21}\mathrm{K}^T & \cdots & a_{N-1,1}\mathrm{K}^T \\ a_{21}\mathrm{K}^T & a_{22}\mathrm{K}^T & \cdots & a_{N-2,1}\mathrm{K}^T \\ \vdots & \vdots & \ddots & \vdots \\ a_{N-1,1}\mathrm{K}^T & a_{N-2,1}\mathrm{K}^T & \cdots & a_{1,1}\mathrm{K}^T \end{bmatrix}, \tag{2.42}$$

$$\mathrm{Q} = \mathrm{A}\otimes\mathrm{I} \quad = \begin{bmatrix} a_{11}\mathrm{I} & a_{21}\mathrm{I} & \cdots & a_{N-1,1}\mathrm{I} \\ a_{21}\mathrm{I} & a_{22}\mathrm{I} & \cdots & a_{N-2,1}\mathrm{I} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N-1,1}\mathrm{I} & a_{N-2,1}\mathrm{I} & \cdots & a_{1,1}\mathrm{I} \end{bmatrix}, \tag{2.43}$$

and C and F are vector of order $(N-1)(M+1)\times 1$ given in (2.44)

$$\mathrm{C}=\begin{bmatrix}\mathrm{C}_1\\ \mathrm{C}_2\\ \vdots\\ \mathrm{C}_{N-1}\end{bmatrix}=\begin{bmatrix}c_{10}\\ \vdots\\ c_{1M}\\ c_{20}\\ \vdots\\ c_{2M}\\ \vdots\\ c_{N-1,0}\\ \vdots\\ c_{N-1,M}\end{bmatrix},\quad \mathrm{F}=\begin{bmatrix}\mathrm{F}_1\\ \mathrm{F}_2\\ \vdots\\ \mathrm{F}_{N-1}\end{bmatrix}=\begin{bmatrix}f_{10}\\ \vdots\\ f_{1M}\\ f_{20}\\ \vdots\\ f_{2M}\\ \vdots\\ f_{N-1,0}\\ \vdots\\ f_{N-1,M}\end{bmatrix}. \tag{2.44}$$

Now, solving the system (2.41), we get the value of the coefficients C_i and putting the value of C_i^T and F_i^T in (2.35), one can obtain the numerical solution of the RFDE (2.30)-(2.32) at each grid points x_i.

The algorithm for solving RFDE (2.30)-(2.32) by the efficient numerical scheme 2.3.1 is given below.

Algorithm 1: Approximating the numerical solution of RFDE (2.30)-(2.32)

Input: The constant k_α, M, N, $f(x)$, $\Psi(t)$, $1 < \alpha \leq 2$.

Output: The approximate solutions at each space grid point

$$\mathrm{u}(x_i, t) \approx \left(([\mathrm{F}_i]^\psi)^T + ([\mathrm{C}_i]^\psi)^T [\mathrm{K}]^\psi\right) \Psi(t).$$

for Numerical solution of RFDE (2.30)-(2.32) by the efficient numerical scheme in subsection 2.3.1 **do**

Step-1.1 Generate the basis function $\psi_j(t)$; $j = 0, \ldots, M$ by using shifted Legendre polynomials as given in Section 2.2.2.3.

Step-1.2 Assuming $\mathrm{u}_t(x_i, t) \approx ([\mathrm{C}_i]^\psi)^T \Psi(t)$, $i = 1, \ldots, N$.

Step-1.3 Compute the operational matrix of integration $[\mathrm{K}]^\psi$ of order $(M+1)$ as in Section 2.2.2.4.

Step-1.4 Compute the operational vector $[\mathrm{F}_i]^\psi$, by approximating the for initial condition at each grid point, as mentioned in (2.26)

Step-1.5 Compute the matrix $[\mathrm{A}]$ of order $(N-1)$ for the Laplacian operator $(-\Delta)^{\alpha/2}$ by using the Matrix Transform Method for uniform mesh $\Omega_h = \{x_i : x_i = ih,\ i = 0, 1, \ldots, N\}$ as mentioned in Lemma 2.2.2.

Step-1.6 Evaluate the matrices $[\mathrm{P}] = \left([\mathrm{A}] \otimes ([\mathrm{K}]^\psi)^T\right)$ and $[\mathrm{Q}] = ([\mathrm{A}] \otimes [\mathrm{I}])$ of order $(N-1)(M+1)$.

Step-1.7 Solve the system of equations $([\mathrm{I}] + k_\alpha[\mathrm{P}])\,[\mathrm{C}]^\psi = -k_\alpha[\mathrm{Q}][\mathrm{F}]^\psi$ get the value of unknown vector $[\mathrm{C}]^\psi$.

Step-1.8 Put the value of $[\mathrm{F}]^\psi, [\mathrm{C}]^\psi$ and $[\mathrm{K}]^\psi$ in $\mathrm{u}(x_i, t) \approx \left(([\mathrm{F}_i]^\psi)^T + ([\mathrm{C}_i]^\psi)^T [\mathrm{K}]^\psi\right) \Psi(t)$, $i = 1, \ldots, N-1$, we get the approximate solution at each grid point x_i.

end

2.3.2 Numerical method for Riesz fractional advection dispersion equation (RFADE)

In this section, we construct an efficient numerical scheme for RFADE

$$\frac{\partial}{\partial t}\mathrm{u}(x,t) = k_\alpha \frac{\partial^\alpha}{\partial |x|^\alpha}\mathrm{u}(x,t) + k_\beta \frac{\partial^\beta}{\partial |x|^\beta}\mathrm{u}(x,t),\ 0 \le t \le T,\ 0 \le x \le L, \tag{2.45}$$

with the initial and boundary conditions given by

$$\mathrm{u}(x,0) = f(x), \tag{2.46}$$

$$\mathrm{u}(0,t) = \mathrm{u}(\pi,t) = 0, \tag{2.47}$$

where $1 < \alpha \le 2$ and $0 < \beta < 1$.

After applying the MTM as mentioned in section 2.2.1, the RFADE (2.45)-(2.47) is converted in the form

$$\frac{d}{dt}\mathrm{U}(t) = -k_\alpha \mathrm{AU} - k_\beta \mathrm{BU}. \tag{2.48}$$

where $\mathrm{A} = \mathrm{V(D)}^{\alpha/2}\mathrm{V}^T$ and $\mathrm{B} = \mathrm{V(D)}^{\beta/2}\mathrm{V}^T$ are the symmetric as well as centro-symmetric matrix of order $(N-1)$ and $\mathrm{U}(t) = [\mathrm{u}_1(t), \mathrm{u}_2(t) \cdots \mathrm{u}_{N-1}(t)]^T$.
We can write the above equation as,

$$\frac{d\mathrm{u}_i(t)}{dt} = -k_\alpha \mathrm{Au}_i(t) - k_\beta \mathrm{Bu}_i(t) \tag{2.49}$$

Following the same process as mentioned in Section 2.3.1, we use the approximation of $\frac{d\mathrm{u}_i(t)}{dt}$ and $\mathrm{u}_i(t)$, from (2.35),

$$\frac{d\mathrm{u}_i(t)}{dt} \approx \mathrm{C}_i^T \Psi(t), \quad \text{and} \quad \mathrm{u}_i(t) \approx \left(\mathrm{F}_i^T + \mathrm{C}_i^T \mathrm{K}\right) \Psi(t), \tag{2.50}$$

where F_i, C_i and $\Psi(t)$ are mentioned in equation (2.25) and K is mentioned in (2.28). Now, using (2.50) in the (2.49), we get

$$\begin{aligned}
[\mathrm{C}_i^T]\Psi(t) &= -k_\alpha \mathrm{A}[\mathrm{F}_i^T + \mathrm{C}_i^T \mathrm{K}]\Psi(t) - k_\beta \mathrm{B}[\mathrm{F}_i^T + \mathrm{C}_i^T \mathrm{K}]\Psi(t), \\
[\mathrm{C}_i^T] &= -k_\alpha \mathrm{A}[\mathrm{F}_i^T + \mathrm{C}_i^T \mathrm{K}] - k_\beta \mathrm{B}[\mathrm{F}_i^T + \mathrm{C}_i^T \mathrm{K}], \\
[\mathrm{C}_i^T] &= -k_\alpha \mathrm{A}[\mathrm{C}_i^T \mathrm{K}] - k_\alpha \mathrm{A}[\mathrm{F}_i^T] - k_\beta \mathrm{B}[\mathrm{F}_i^T] - k_\beta \mathrm{B}[\mathrm{C}_i^T \mathrm{K}].
\end{aligned} \tag{2.51}$$

Now, putting the value of $[\mathrm{F}_i^T]$, $[\mathrm{C}_i^T]$ from (2.25) and $[\mathrm{C}_i^T \mathrm{K}]$ from (2.38) in 2.51 and after rigorous calculation (see 2.6.2) and comparing the coefficients of c_{ij} from (2.95) for $i = 1, 2, ..., N-1$ and $j = 0, 1, ..., M$, we get the $(N-1)(M+1)$ system of algebraic equation which is given below

$$c_{ij} = -k_\alpha \sum_{l=1}^{N-1} a_{il} \left(\sum_{m=0}^{M} c_{lm} k_{mj} \right) - k_\alpha \sum_{l=1}^{N-1} a_{il} f_{lj} - k_\beta \sum_{l=1}^{N-1} b_{il} \left(\sum_{m=0}^{M} c_{lm} k_{mj} \right) - k_\beta \sum_{l=1}^{N-1} b_{il} f_{lj}, \tag{2.52}$$

where $a_{ij} = a_{ji}$ with $a_{ij} = a_{N-i+1,N-j+1}$ for A and $b_{ij} = b_{ji}$ with $b_{ij} = b_{N-i+1,N-j+1}$ for B because A and B are symmetric as well as centro-symmetric matrix. Then, the matrix form of the numerical scheme is given by

$$\begin{aligned}
\mathrm{C} &= -k_\alpha(\mathrm{A} \otimes \mathrm{K}^T)\mathrm{C} - k_\alpha(\mathrm{A} \otimes \mathrm{I})\mathrm{F} - k_\beta(\mathrm{B} \otimes \mathrm{K}^T)\mathrm{C} - k_\beta(\mathrm{B} \otimes \mathrm{I})\mathrm{F}, \\
\mathrm{C} &= -k_\alpha \mathrm{PC} - k_\alpha \mathrm{QF} - k_\beta \mathrm{RC} - k_\beta \mathrm{SF}, \\
[\mathrm{I} + k_\alpha \mathrm{P} + k_\beta \mathrm{R}]\mathrm{C} &= -[k_\alpha \mathrm{QF} + k_\beta \mathrm{SF}],
\end{aligned} \tag{2.53}$$

where '$\otimes$' denotes the kronecker product. $\mathrm{P}, \mathrm{Q}, \mathrm{R}, \mathrm{S}$ are the matrices of order $(N-1)(M+1) \times (N-1)(M+1)$. R, S can be calculated in the same way as P, Q mentioned in (2.42, 2.43) and C, F are the vectors of order $(N-1)(M+1)\times 1$ as mentioned in (2.44), where

$$\mathrm{P} = \mathrm{A} \otimes \mathrm{K}^T, \ \mathrm{Q} = \mathrm{A} \otimes \mathrm{I}, \ \mathrm{R} = \mathrm{B} \otimes \mathrm{K}^T, \ \mathrm{S} = \mathrm{B} \otimes \mathrm{I}. \tag{2.54}$$

Now, solving the system (2.53) we get the values of the coefficients C_i and putting the value of C_i^T and F_i^T in (2.50) one can obtain the numerical solution of the RFADE (2.45)-(2.47) at each grid points x_i.

The algorithm for solving RFADE (2.45)-(2.47) by the efficient numerical scheme 2.3.2 is given below.

Algorithm 2: Approximating the numerical solution of RFADE (2.45)-(2.47)

Input: The constants $k_\alpha, k_\beta,\ M, N,\ 1 < \alpha \leq 2, 0 < \beta < 1,\ f(x),\ \Psi(t)$.

Output: The approximate solutions at each space grid point

$$\mathrm{u}(x_i, t) \approx \left(([\mathrm{F}_i]^\psi)^T + ([\mathrm{C}_i]^\psi)^T [\mathrm{K}]^\psi\right) \Psi(t).$$

for Numerical solution of RFADE (2.45)-(2.47) by efficient numerical scheme in subsection 2.3.2 **do**

Step-2.1 Generate the basis function $\psi_j(t);\ j = 0, \ldots, M$ by using shifted Legendre polynomials as given in Section 2.2.2.3.

Step-2.2 Assuming $\mathrm{u_t}(x_i, t) \approx ([\mathrm{C}_i]^\psi)^T \Psi(t),\ \ i = 1, \ldots, N-1$.

Step-2.3 Compute the operational matrix of integration $[\mathrm{K}]^\psi$ of order $(M+1)$ as defined in Section 2.2.2.4.

Step-2.4 Compute the operational vector $[\mathrm{F}_i]^\psi$ by approximating the for initial condition at each grid point as mentioned in (2.26).

Step-2.5 Compute the matrix [A] and [B] of order $(N-1)$ for the Laplacian operator $(-\Delta)^{\alpha/2}$ and $(-\Delta)^{\beta/2}$ respectively by using the Matrix Transform Method for uniform mesh $\Omega_h = \{x_i : x_i = ih,\ \ i = 0, 1, \ldots, N\}$ as mentioned in Lemma 2.2.2.

Step-2.6 Evaluated matrices $[\mathrm{P}] = \left([\mathrm{A}] \otimes ([\mathrm{K}]^\psi)^T\right)$, $[\mathrm{Q}] = ([\mathrm{A}] \otimes [\mathrm{I}])$, $[\mathrm{R}] = \left([\mathrm{B}] \otimes ([\mathrm{K}]^\psi)^T\right)$ and $[\mathrm{S}] = ([\mathrm{B}] \otimes [\mathrm{I}])$ of order $(N-1)(M+1)$.

Step-2.7 Solve the system of equations $([\mathrm{I}] + k_\alpha[\mathrm{P}] + k_\beta[\mathrm{R}])\,[\mathrm{C}]^\psi = -\left(k_\alpha[\mathrm{Q}][\mathrm{F}]^\psi + k_\beta[\mathrm{S}][\mathrm{F}]^\psi\right)$ get the value of unknown vector $[\mathrm{C}]^\psi$.

Step-2.8 Put the value of $[\mathrm{F}]^\psi, [\mathrm{C}]^\psi$ and $[\mathrm{K}]^\psi$ in $\mathrm{u}(x_i, t) \approx \left(([\mathrm{F}_i]^\psi)^T + ([\mathrm{C}_i]^\psi)^T [\mathrm{K}]^\psi\right) \Psi(t),\ \ i = 1, \ldots, N-1$, we get the approximate solution at each grid point x_i.

end

2.4 Optimal error bounds of numerical approximation

Lemma 2.4.1. Let $\mathrm{u}(x_i,t)$ be sufficiently smooth function in $L_2[0,1]$ and $\dfrac{\partial \tilde{\mathrm{u}}_i(t)}{\partial t}$ be the numerial approximation of $\dfrac{\partial \mathrm{u}_i(t)}{\partial t}$ obtained by using (M+1) elements of the shifted Legendre polynomial. Assuming that the third derivative of $\dfrac{\partial \mathrm{u}_i(t)}{\partial t}$ is bounded by a constant $\mathscr{K}$ i.e. $\left|\dfrac{\partial^3 \mathrm{u}_i(t)}{\partial t^3}\right| \leq \mathscr{K}$, then we have the following upper bound of error

$$\left\| \frac{\partial \mathrm{u}_i(t)}{\partial t} - \frac{\partial \tilde{\mathrm{u}}_i(t)}{\partial t} \right\|_{L^2}^2 \leq \sum_{j=M+1}^{\infty} \frac{\mathscr{K}^2}{(2j-3)^4}. \tag{2.55}$$

Proof. Let us suppose that the time derivative of the function $\mathrm{u}(x_i,t)$ is approximated by the basis $\Psi(t)$ of $L_2[0,1]$, then

$$\frac{\partial \mathrm{u}_i(t)}{\partial t} = \sum_{j=0}^{\infty} c_{ij}\psi_j(t), \tag{2.56}$$

where the coefficient c_{ij} are calculated by

$$c_{ij} = \frac{\left\langle \dfrac{\partial \mathrm{u}_i(t)}{\partial t}, \psi_j(t) \right\rangle}{\langle \psi_j(t), \psi_j(t) \rangle}. \tag{2.57}$$

Truncating the summation upto M basis point, (2.56) can be written as

$$\frac{\partial \tilde{\mathrm{u}}_i(t)}{\partial t} = \sum_{j=0}^{M} c_{ij}\psi_j(t). \tag{2.58}$$

Subtracting (2.58) to (2.56),

$$\frac{\partial \mathrm{u}_i(t)}{\partial t} - \frac{\partial \tilde{\mathrm{u}}_i(t)}{\partial t} = \sum_{j=M+1}^{\infty} c_{ij}\psi_j(t). \tag{2.59}$$

By definition of L_2-norm

$$\begin{aligned}\left\| \frac{\partial \mathrm{u}_i(t)}{\partial t} - \frac{\partial \tilde{\mathrm{u}}_i(t)}{\partial t} \right\|_{L^2}^2 &= \int_0^1 \left(\frac{\partial \mathrm{u}_i(t)}{\partial t} - \frac{\partial \tilde{\mathrm{u}}_i(t)}{\partial t} \right)^2 dt \\ &= \int_0^1 \left(\sum_{j=M+1}^{\infty} c_{ij}\psi_j(t) \right)^2 dt \\ &= \sum_{j=M+1}^{\infty} c_{ij}^2 \frac{1}{(2j+1)} dt, \end{aligned} \tag{2.60}$$

where, $\psi_j(t)$ is the shifted Legendre polynomials of order j, i.e.

$$\psi_j(t) = P_j(2t-1), \quad j = 0, 1, 2, ..., \tag{2.61}$$

Now, the coefficients c_{ij} are calculated by the formula (2.57), we get

$$\begin{aligned} c_{ij} &= (2j+1) \int_0^1 \frac{\partial \mathrm{u}_i(t)}{\partial t} \psi_j(t) dt \\ &= (2j+1) \int_0^1 \frac{\partial \mathrm{u}_i(t)}{\partial t} P_j(2t-1) dt \\ &= (2j+1) \int_{-1}^1 \frac{\partial \mathrm{u}_i(\frac{s+1}{2})}{\partial s} P_j(s) ds, \end{aligned} \tag{2.62}$$

where $s = 2t - 1$. By the property of Legendre polynomials

$$(2j+1)P_j(t) = \frac{d}{dt}(P_{j+1}(t) - P_{j-1}(t)). \tag{2.63}$$

Using above property in (2.62),

$$\begin{aligned}
c_{ij} &= (2j+1)\int_{-1}^{1} \frac{\partial \mathrm{u}_i(\frac{s+1}{2})}{\partial s}\left[\frac{(P_{j+1}(s)-P_{j-1}(s))'}{(2j+1)}\right]ds,\\
&= \int_{-1}^{1} \frac{\partial \mathrm{u}_i(\frac{s+1}{2})}{\partial s}[P_{j+1}(s)-P_{j-1}(s)]'ds,\\
&= -\frac{1}{2}\int_{-1}^{1} \frac{\partial^2 \mathrm{u}_i(\frac{s+1}{2})}{\partial s^2}[P_{j+1}(s)-P_{j-1}(s)]ds,\\
&= -\frac{1}{2}\int_{-1}^{1} \frac{\partial^2 \mathrm{u}_i(\frac{s+1}{2})}{\partial s^2}\left[\frac{(P_{j+2}(s)-P_j(s))'}{(2j+3)}-\frac{(P_j(s)-P_{j-2}(s))'}{(2j-1)}\right]ds,\\
&= \frac{1}{4}\int_{-1}^{1} \frac{\partial^3 \mathrm{u}_i(\frac{s+1}{2})}{\partial s^3}\left[\frac{(P_{j+2}(s)-P_j(s))}{(2j+3)}-\frac{(P_j(s)-P_{j-2}(s))}{(2j-1)}\right]ds,\\
&= \frac{1}{4}\int_{-1}^{1} \frac{\partial^3 \mathrm{u}_i(\frac{s+1}{2})}{\partial s^3}\sigma_j(s)ds, \qquad (2.64)
\end{aligned}$$

where $\sigma_j(s) = \dfrac{(P_{j+2}(s)-P_j(s))}{(2j+3)} - \dfrac{(P_j(s)-P_{j-2}(s))}{(2j-1)}$.

Taking square of the absolute of (2.64), one can obtain

$$\begin{aligned}
|c_{ij}|^2 &= \left|\frac{1}{4}\int_{-1}^{1} \frac{\partial^3 \mathrm{u}_i(\frac{s+1}{2})}{\partial s^3}\sigma_j(s)ds\right|^2\\
&= \frac{1}{16}\left|\int_{-1}^{1} \frac{\partial^3 \mathrm{u}_i(\frac{s+1}{2})}{\partial s^3}\sigma_j(s)ds\right|^2\\
&\le \frac{1}{16}\int_{-1}^{1}\left|\frac{\partial^3 \mathrm{u}_i(\frac{s+1}{2})}{\partial s^3}\right|^2|\sigma_j(s)ds|^2\\
&\le \frac{\mathscr{K}^2}{16}\int_{-1}^{1}|\sigma_j(s)ds|^2\\
&\le \frac{\mathscr{K}^2}{16}\int_{-1}^{1}\left(\frac{(P_{j+2}(s)-P_j(s))}{(2j+3)}-\frac{(P_j(s)-P_{j-2}(s))}{(2j-1)}\right)^2 ds\\
&\le \frac{\mathscr{K}^2}{16}\int_{-1}^{1}\left(\frac{(2j-1)P_{j+2}(s)-(4j+2)P_j(s)+(2j-3)P_{j-2}(s)}{(2j+3)(2j-1)}\right)^2 ds\\
&\le \frac{\mathscr{K}^2}{16}\int_{-1}^{1}\left(\frac{(2j-1)^2P_{j+2}^2(s)+(4j+2)^2P_j^2(s)+(2j-3)^2P_{j-2}^2(s)}{(2j+3)^2(2j-1)^2}\right)ds
\end{aligned}$$

$$|c_{ij}|^2 \leq \frac{\mathscr{K}^2}{16}\frac{1}{(2j+3)^2(2j-1)^2}\frac{12(2j+3)^2}{(2j-3)}$$
$$\leq \frac{\mathscr{K}^2}{16}\frac{12}{(2j-1)^2(2j-3)}, \tag{2.65}$$

Substituting (2.65) in (2.60), we obtain

$$\left\|\frac{\partial \mathrm{u}_i(t)}{\partial t} - \frac{\partial \tilde{\mathrm{u}}_i(t)}{\partial t}\right\|_{L^2}^2 = \sum_{j=M+1}^{\infty} c_{ij}^2 \frac{1}{(2j+1)}dt$$
$$\leq \sum_{j=M+1}^{\infty} \frac{3}{4}\frac{\mathscr{K}^2}{(2j-1)^2(2j-3)}\frac{1}{(2j+1)}$$
$$\leq \sum_{j=M+1}^{\infty} \frac{\mathscr{K}^2}{(2j-1)^3(2j-3)}$$
$$\leq \sum_{j=M+1}^{\infty} \frac{\mathscr{K}^2}{(2j-3)^4}.$$

This proves the Lemma. □

Lemma 2.4.2. Let $\mathrm{u}(x_i,t)$ be sufficiently smooth function in $L_2[0,1]$ and $\frac{\partial \tilde{\mathrm{u}}_i(t)}{\partial t}$ be the numerial approximation of $\frac{\partial \mathrm{u}_i(t)}{\partial t}$ obtained by using (M+1) elements of the shifted Chebyshev polynomial of second kind. Assuming that the third derivative of $\frac{\partial \mathrm{u}_i(t)}{\partial t}$ is bounded by a constant ξ i.e. $\left|\frac{\partial^3 \mathrm{u}_i(t)}{\partial t^3}\right| \leq \xi$, then we have the following upper bound of error

$$\left\|\frac{\partial \mathrm{u}_i(t)}{\partial t} - \frac{\partial \tilde{\mathrm{u}}_i(t)}{\partial t}\right\|_{L^2}^2 \leq \frac{\pi\xi^2}{3072}\mathrm{F}_3(1+M). \tag{2.66}$$

where $\mathrm{F}_n(z)$ is the Poly Gamma function defined by

$$\mathrm{F}_n(z) = (-1)^{n+1}n!\sum_{k=0}^{\infty}\frac{1}{(z+k)^{n+1}} \tag{2.67}$$

Proof. Proof is similar to Theorem 3 of [165] when $\alpha = 1$ □

2.5 Numerical examples

In this section, we give the numerical examples of the RFDE and RFADE to demonstrate the effectiveness of the numerical schemes. The accuracy of the proposed schemes for both cases is demonstrated by the following error norms

$$\|\mathrm{u}-\mathrm{U}\| = \begin{cases} \left(\sum_{i=1}^{N} h|\mathrm{u}(x_i,T)-\mathrm{U}(x_i,T)|^2\right)^{\frac{1}{2}}, & L_2\text{-Norm}, \\ \max_{0\le i\le N}|\mathrm{u}(x_i,T)-\mathrm{U}(x_i,T)|, & L_\infty\text{-Norm}, \end{cases} \tag{2.68}$$

where $\mathrm{u}(x,t)$ and $\mathrm{U}(x,t)$ are exact and numerical solutions of the FDEs. The spatial and temporal order of convergence is calculated by the following formula

$$\text{Rate of convergence in spacial direction}= log_{\frac{N_1}{N_2}}\frac{\|\text{error}(N_1)\|}{\|\text{error}(N_2)\|}, \tag{2.69}$$

where, error(N_i) denotes the error corresponds to the grid points N_i.

Apart from this, the numerical stability and robustness of our numerical algorithms are verified numerically by adding some noisy data in the initial condition, as discussed in [166, 167]. Let $f^{\delta}(x,0)$ denotes the initial condition with noise δ i.e.

$$f^{\delta}(x_i,0) = f(x_i,0)+\delta\omega, \tag{2.70}$$

where, ω is the uniform random variable whose values lies in $[-1,1]$, such that

$$\max_{1\le i\le N}(f(x_i,0)-f^{\delta}(x_i,0)) \le \delta, \tag{2.71}$$

we have choosen two different value of noise δ^i as $\delta^1 = m\%$ of μ^N and $\delta^2 = \sigma^N$, where μ_k^N is the mean of the initial values at each grid points defined as

$$\mu^N = \frac{1}{N+1}\sum_{i=0}^{N} f(x_i, 0), \tag{2.72}$$

and σ^N is standard deviation (sometimes called root mean square) and defined as

$$\sigma^N = \left(\sum_{i=1}^{N} h|\mathrm{u}(x_i, T) - \mathrm{U}(x_i, T)|^2\right)^{\frac{1}{2}}. \tag{2.73}$$

Remark 2.5.1. In all test examples, calculation of σ^N is performed by taking $N = 160$ and $m = 0.1$.

Remark 2.5.2. Numerical experiments are carried out on a Laptop with (a) Intel(R) Core (TM) i5-2430M CPU @ 2.40 GHz (b) RAM 8GB and (c) System type 64-bit operating system running on MATLAB 2015a (The MathWorks, Inc., Natick, Massachusetts) programming.

2.5.1 Numerical examples for Riesz-space fractional diffusion equation

Example 2.5.1. Consider the following RFDE

$$\frac{\partial}{\partial t}u(x,t) = k_\alpha \frac{\partial^\alpha}{\partial |x|^\alpha}u(x,t), \quad 1 < \alpha \leq 2, \tag{2.74}$$

with initial and boundary condition

$$u(x,0) = x^2(\pi - x), \quad 0 \leq x \leq \pi, \tag{2.75}$$

$$u(0,t) = u(\pi,t) = 0, \quad 0 \leq t \leq T. \tag{2.76}$$

From [72], we know the exact solution of RFDE in Ex. 2.5.1 is given by,

$$u(x,t) = \sum_{n=1}^{\infty} \left[\frac{8}{n^3}(-1)^{n+1} - \frac{4}{n^3} \right] \sin(nx) e^{-\left[(n^2)^{\frac{\alpha}{2}} k_\alpha\right] t}. \tag{2.77}$$

The outcomes of Ex. 2.5.1 are presented as:

- Fig. 2.1 and Fig. 2.2 show approximate solution, exact solution and absolute errors in Ex. 2.5.1, respectively, for $\alpha = 1.5$, $k_\alpha = 0.25$ and $M = 10$ SLP basis function. It can be observed from these figures that, approximate solutions by the proposed scheme (2.41) are in good agreement with the exact solutions at each time level.

- Fig. 2.3 reflects the behavior of fractional order α in the solution profile of Ex. 2.5.1 at the various time levels. It can be observed that the numerical scheme beautifully captures the shifting property of fractional diffusion process which is rightwards than standard diffusion process (in red color) at each time levels.

- Fig. 2.4 and Fig 2.5 verifies the numerical stability of our scheme with respect to SLP and SCP basis function respectively. The behavior of absolute errors at final time $T = 1$ for $\alpha = 1.5$ without noise and with two different noisy inputs, $\delta^1 = 0.1\%\mu^{160}$ and $\delta^2 = \sigma^{160}$, in the initial data is shown in both the figures. We see that the variation in the absolute error with noisy data is negligible as compared to without noisy data. Therefore, it can be concluded that the scheme (2.41) is numerically stable with respect to both the basis function.

- Tables 2.1-2.4 demonstrate the role of basis function, L_2-error, CPU time (in seconds) and spacial order of convergence. We have calculated the convergence order in space of the proposed scheme with respect to L_2-norm at time $T = 0.4$ by using two different set of elements in the basis function i.e. $M = 5$ and

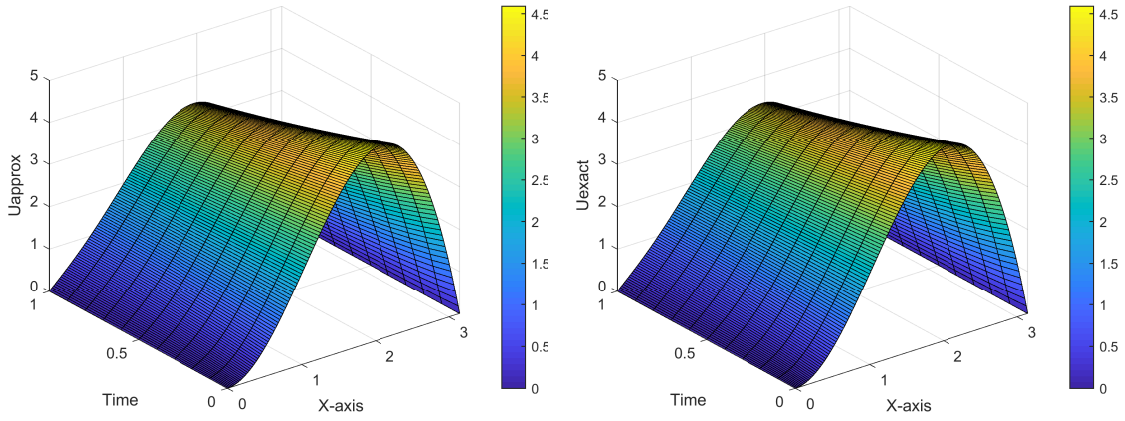

FIGURE 2.1: Surface plot of approximate solution (left) and exact solution (right) of Ex. 2.5.1 for $\alpha = 1.5$, $h = \pi/160$ and $M = 10$ SLP basis.

$M = 10$ for both the SLP/SCP basis functions. It is clear from Tables 2.1-2.4 that an increase in the number of basis function will reduce the absolute error at higher grid points when α increase. It is also found that the spacial order of convergence is of the second order for all values of $\alpha \in (1, 2]$.

- It can be seen from the Tables 2.1-2.4 that L_2 error obtained by using SCP basis is almost same as obtained by SLP basis but the CPU time taken by SLP basis function is very less as compared to SCP basis function. Therefore the proposed numerical scheme is more efficient with SLP basis function.

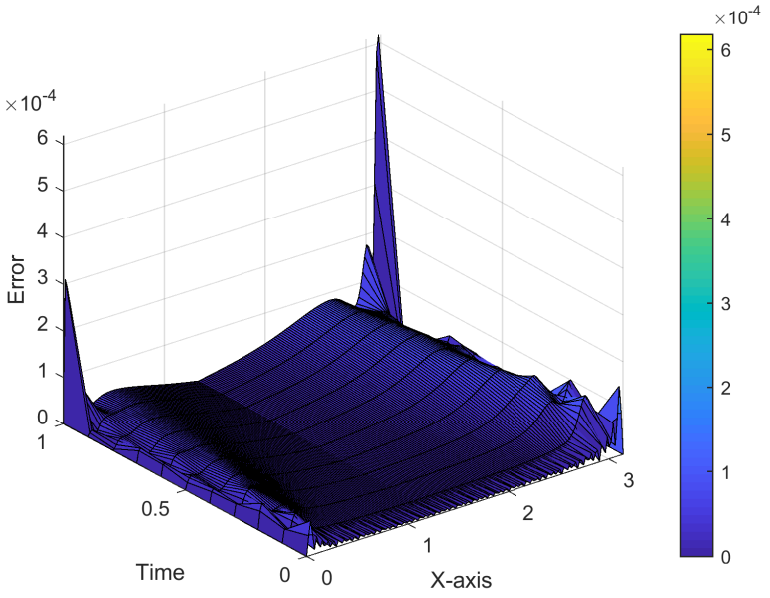

FIGURE 2.2: Surface plot of absolute error of Ex. 2.5.1 for $\alpha = 1.5$, $h = \pi/160$ and $M = 10$ SLP basis.

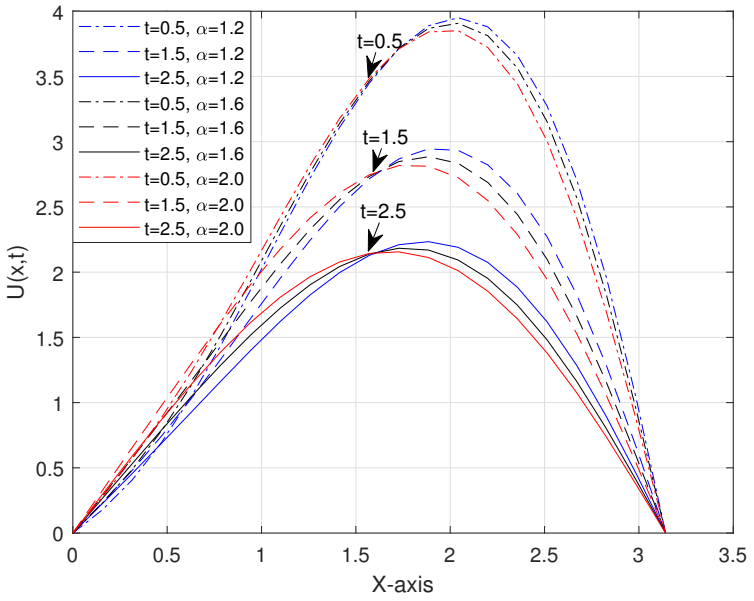

FIGURE 2.3: Approximate solution of Ex. 2.5.1 at various α and at various time level, when $h = \pi/20$ and $M = 10$ SLP basis.

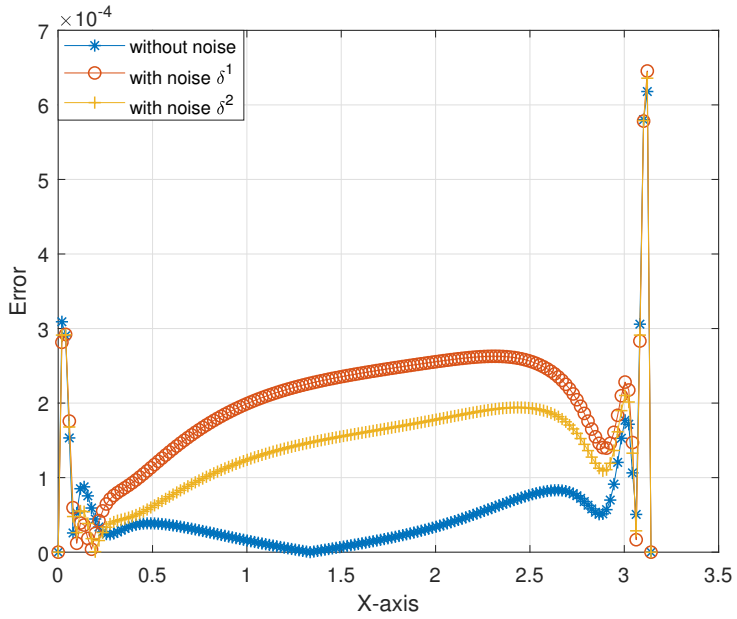

FIGURE 2.4: Absolute error of Ex. 2.5.1 for $\alpha = 1.5$, $h = \pi/160$, at $T = 1.0$, with different noisy input δ^i in the initial data, when $M = 10$ SLP basis.

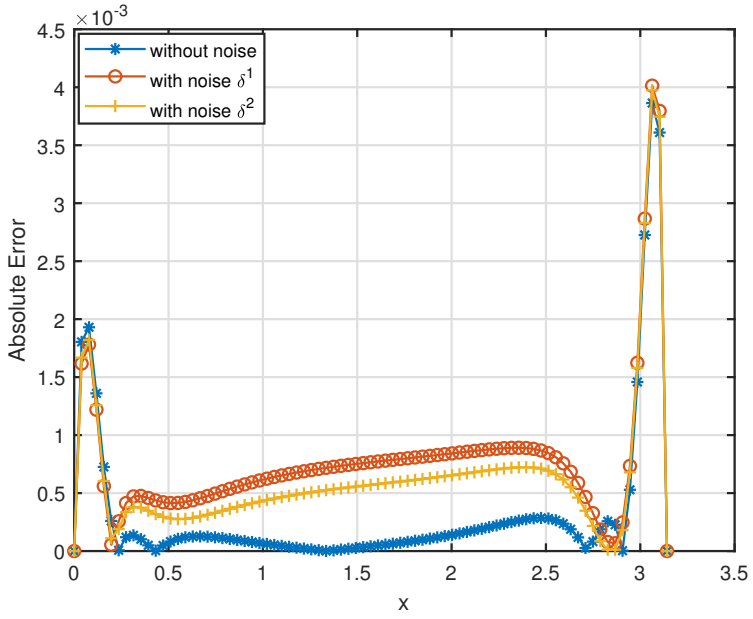

FIGURE 2.5: Absolute error of Ex. 2.5.1 for $\alpha = 1.5$, $h = \pi/80$, at $T = 1.0$, with different noisy input δ^i in the initial data, when $M = 5$ SCP basis.

h	$M=5,\ \alpha=1.1$			$M=5,\ \alpha=1.5$			$M=5,\ \alpha=1.9$		
	$\|u-U\|_2$	CO	CPU time	$\|u-U\|_2$	CO	CPU time	$\|u-U\|_2$	CO	CPU time
$\pi/5$	1.667e-02	-	3.45	2.815e-02	-	3.34	4.961e-02	-	3.38
$\pi/10$	7.055e-03	1.23	4.36	1.292e-02	1.12	4.21	1.834e-02	1.44	4.80
$\pi/20$	2.353e-03	1.58	6.72	3.635e-03	1.83	6.38	4.662e-03	1.98	6.74
$\pi/40$	6.334e-04	1.89	14.29	8.834e-04	2.04	14.27	1.135e-03	2.04	14.76
$\pi/80$	1.517e-04	2.06	42.46	2.307e-04	1.94	44.55	4.982e-04	1.18	48.54
$\pi/160$	3.691e-05	2.04	245.02	1.651e-04	0.48	248.42	4.991e-04	0.00	232.59

TABLE 2.1: Errors, Convergence Order (CO) and CPU time (in seconds) by using SLP basis for Ex. 2.5.1 at time $T=0.4$ with $dt=0.1$ and different values of α.

h	$M=5,\ \alpha=1.1$			$M=5,\ \alpha=1.5$			$M=5,\ \alpha=1.9$		
	$\|u-U\|_2$	CO	CPU time	$\|u-U\|_2$	CO	CPU time	$\|u-U\|_2$	CO	CPU time
$\pi/5$	1.659e-02	-	9.47	2.809e-02	-	9.70	4.957e-02	-	9.98
$\pi/10$	7.054e-03	1.23	10.12	1.289e-02	1.12	10.00	1.830e-02	1.44	10.08
$\pi/20$	2.352e-03	1.58	18.32	3.613e-03	1.83	18.41	4.622e-03	1.99	18.58
$\pi/40$	6.299e-04	1.90	45.89	8.699e-04	2.05	43.44	1.134e-03	2.03	45.47
$\pi/80$	1.484e-04	2.08	251.68	2.349e-04	1.89	234.11	5.775e-04	0.97	226.65
$\pi/160$	3.758e-05	1.98	4585.94	1.889e-04	0.31	4311.81	5.954e-04	0.45	4401.33

TABLE 2.2: Errors, Convergence Order (CO) and CPU time (in seconds) by using SCP basis for Ex. 2.5.1 at time $T=0.4$ with $dt=0.1$ and different values of α.

h	$M = 10,\ \alpha = 1.1$			$M = 10,\ \alpha = 1.5$			$M = 10,\ \alpha = 1.9$		
	$\|u - U\|_2$	CO	CPU time	$\|u - U\|_2$	CO	CPU	$\|u - U\|_2$	CO	CPU time
$\pi/5$	1.667e-02	-	4.70	2.815e-02	-	4.70	4.961e-02	-	4.96
$\pi/10$	7.055e-03	1.23	6.26	1.292e-02	1.12	6.57	1.853e-02	1.43	6.64
$\pi/20$	2.361e-03	1.58	12.86	3.693e-03	1.81	12.78	4.814e-03	1.94	10.21
$\pi/40$	6.475e-04	1.87	36.72	9.415e-04	1.97	39.38	1.212e-03	1.99	37.86
$\pi/80$	1.643e-04	1.98	186.51	2.363e-04	1.99	180.89	3.091e-04	1.97	168.32
$\pi/160$	4.132e-05	1.99	1039.77	5.982e-05	1.98	1216.99	9.523e-05	1.97	1078.45

TABLE 2.3: Errors, Convergence Order (CO) and CPU time (in seconds) by using SLP basis for Ex. 2.5.1 at time $T = 0.4$ with $dt = 0.1$ and different values of α.

h	$M = 10,\ \alpha = 1.1$			$M = 10,\ \alpha = 1.5$			$M = 10,\ \alpha = 1.9$		
	$\|u - U\|_2$	CO	CPU time	$\|u - U\|_2$	CO	CPU time	$\|u - U\|_2$	CO	CPU time
$\pi/5$	1.659e-02	-	24.42	2.809e-02	-	25.91	4.959e-02	-	24.41
$\pi/10$	7.057e-03	1.23	26.98	1.293e-02	1.12	30.10	1.846e-02	1.43	29.81
$\pi/20$	2.362e-03	1.58	62.11	3.689e-03	1.81	61.17	4.812e-03	1.94	60.67
$\pi/40$	6.467e-04	1.87	267.95	9.405e-04	1.97	256.15	1.217e-03	1.98	255.86
$\pi/80$	1.644e-04	1.98	5628.09	2.367e-04	1.99	5489.89	3.291e-04	1.44	5461.44

TABLE 2.4: Errors, Convergence Order (CO) and CPU time (in seconds) by using SCP basis for Ex. 2.5.1 at time $T = 0.4$ with $dt = 0.1$ and different values of α.

Now, we will test the numerical scheme to demonstrate the role of fractional order α in a more effective manner. Let's consider another example.

Example 2.5.2. Consider the following RFDE

$$\frac{\partial}{\partial t}u(x,t) = k_\alpha \frac{\partial^\alpha}{\partial |x|^\alpha}u(x,t), \qquad 1 < \alpha \leq 2, \tag{2.78}$$

with initial and boundary condition

$$u(x,0) = \sin(4x), \qquad 0 \le x \le \pi, \tag{2.79}$$

$$u(0,t) = u(\pi,t) = 0, \quad 0 \le t \le T. \tag{2.80}$$

From [72], we know the exact solution of the RFDE in Ex. 2.5.2 is given by

$$u(x,t) = \sum_{n=1}^{\infty} b_n \sin(nx) e^{-\left[(n^2)^{\frac{\alpha}{2}} k_\alpha\right] t}, \tag{2.81}$$

where

$$b_n = \frac{2}{\pi} \int_0^{\pi} f(\xi) \sin(n\xi) d\xi. \tag{2.82}$$

The outcomes of Ex. 2.5.2 are presented as:

- Fig. 2.6 shows approximate solution of Ex. 2.5.2 for $\alpha = 1.5$, $k_\alpha = 0.25$ and $M = 10$ SLP basis functions. The absolute errors shown in Fig. 2.7 verifies that approximate solution by scheme (2.41) are in good agreement with the exact solution at each time level.

- Fig. 2.8 demonstrate the case of standard diffusion ($\alpha = 2$) of Ex. 2.5.2. From Figs. 2.6, 2.8 and 2.9 we can say that the phenomena of fractional diffusion process is slower than the standard diffusion process.

- Fig. 2.9 demonstrates the behavior of α in the solution profile of Ex. 2.5.2 at $T = 0.5$. It can be seen that by increasing the value of fractional order α, the amplitude of sine wave decreases. Therefore, the given scheme verifies that the diffusion process is proportional to Riesz-space fractional derivative α.

- Table 2.5 shows the L_2 error and verifies that the second order convergent rate in space with respect to L_2-norm at time $T = 0.4$ for all values of $1 < \alpha \leq 2$.

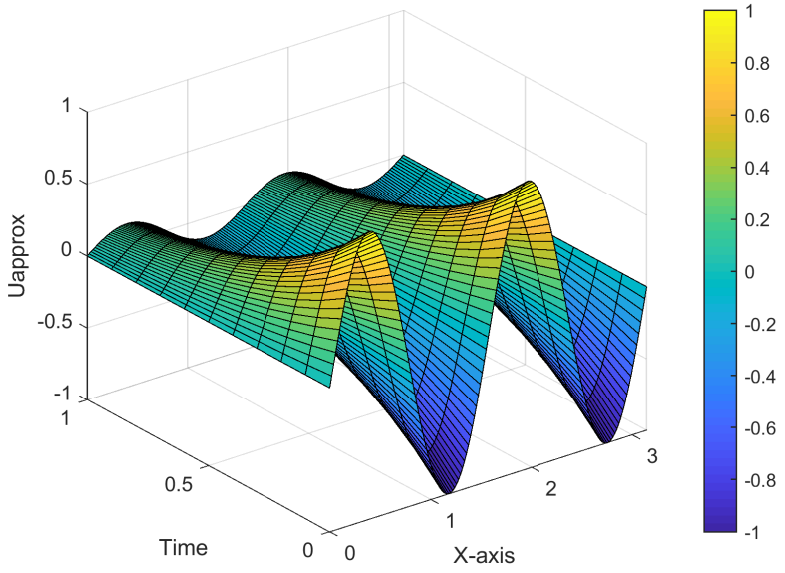

FIGURE 2.6: Surface plot of approximate solution of Ex. 2.5.2 for $\alpha = 1.5$, $h = \pi/160$ and $M = 10$ SLP basis.

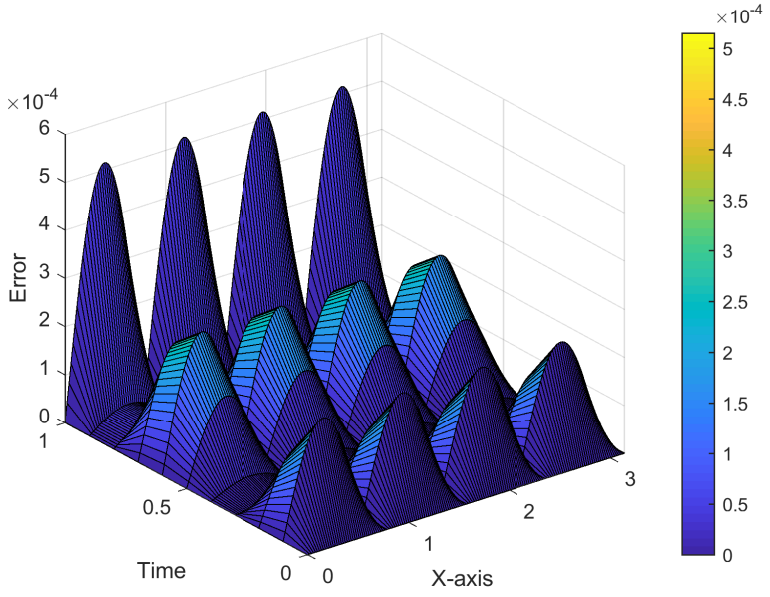

FIGURE 2.7: Surface plot of absolute error in Ex. 2.5.2 for $\alpha = 1.5$, $h = \pi/160$ and $M = 10$ SLP basis.

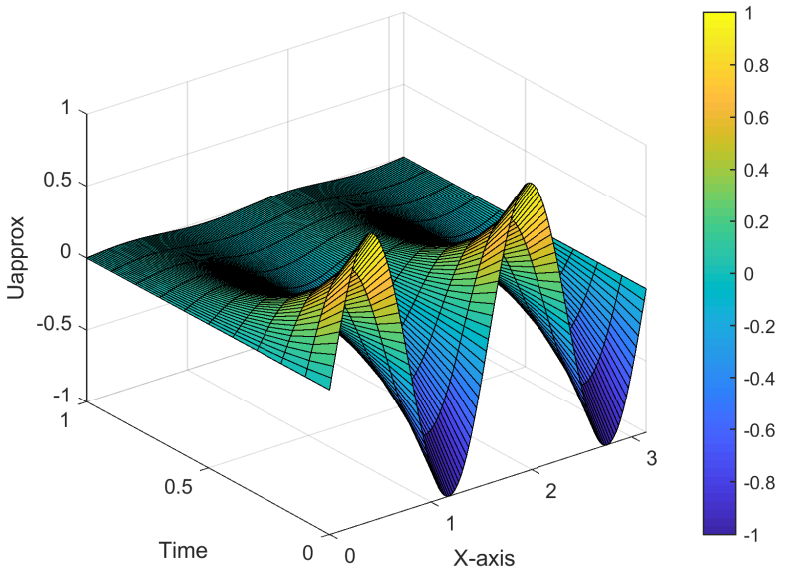

FIGURE 2.8: Surface plot of approximate solution of Ex. 2.5.2 for $\alpha = 2$ (standard diffusion), $h = \pi/160$ and $M = 10$ SLP basis.

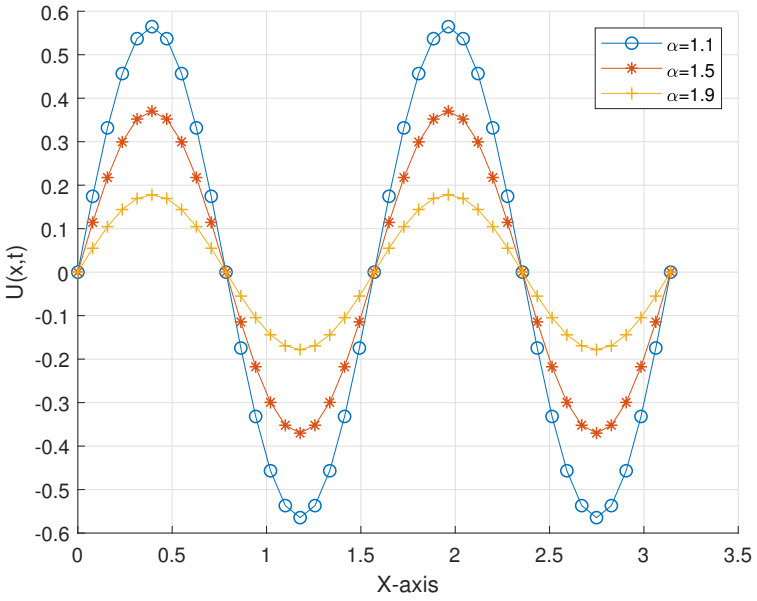

FIGURE 2.9: Approximate solution of Ex. 2.5.2 for $\alpha = 1.1, 1.5, 1.9$ at $T = 0.5$, $h = \pi/40$ and $M = 10$ SLP basis.

h	$M=10,\ \alpha=1.1$		$M=10,\ \alpha=1.5$		$M=10,\ \alpha=1.9$	
	$\|u-U\|_2$	CO	$\|u-U\|_2$	CO	$\|u-U\|_2$	CO
$\pi/5$	1.022e-01	-	2.814e-02	-	2.414e-01	-
$\pi/10$	2.614e-02	1.96	1.290e-02	1.12	5.617e-02	2.10
$\pi/20$	6.573e-03	1.99	3.691e-03	1.81	1.373e-02	2.04
$\pi/40$	1.642e-03	2.00	9.415e-04	1.97	3.402e-03	2.01
$\pi/80$	4.116e-04	2.00	2.362e-04	1.99	8.471e-04	2.00
$\pi/160$	1.032e-04	2.00	5.983e-05	1.98	2.124e-04	2.00

TABLE 2.5: Errors and Convergence Order (CO) by using SLP basis for Ex. 2.5.2 at time $T=0.4$ with $dt=0.1$ and different values of α.

2.5.2 Numerical example for Riesz-space fractional advection dispersion equation

Example 2.5.3. Consider the following RFADE

$$\frac{\partial}{\partial t}u(x,t)=k_\alpha\frac{\partial^\alpha}{\partial|x|^\alpha}u(x,t)+k_\beta\frac{\partial^\beta}{\partial|x|^\beta}u(x,t),\quad 1<\alpha\leq 2,\ 0<\beta<1, \tag{2.83}$$

with initial and boundary condition

$$u(x,0)=x^2(\pi-x),\quad 0\leq x\leq\pi, \tag{2.84}$$

$$u(0,t)=u(\pi,t)=0,\quad 0\leq t\leq T. \tag{2.85}$$

From [72], we know the exact solution of the RFADE in Ex. 2.5.3 is given by

$$u(x,t)=\sum_{n=1}^{\infty}\left[\frac{8}{n^3}(-1)^{n+1}-\frac{4}{n^3}\right]\sin(nx)e^{-\left[(n^2)^{\frac{\alpha}{2}}k_\alpha+(n^2)^{\frac{\beta}{2}}k_\beta\right]t}. \tag{2.86}$$

The outcomes of Ex. 2.5.3 are presented as:

- Fig. 2.10 and Fig. 2.11 shows approximate solution, exact solution and absolute errors in Ex. 2.5.3, respectively, for $\alpha = 1.5$, $\beta = 0.5$, $k_\alpha = k_\beta = 0.25$ and $M = 10$ SLP basis functions. It can be observed from these figures that, approximate solutions by scheme (2.53) shows a good similarity with exact solutions at each time level.

- Fig. 2.12 reflects the behavior of fractional order α, for a fix β and Fig. 2.13 reflects the behavior of fractional order β, for a fix α, in the solution profile of Ex. 2.5.3 at various time levels. In this case also, the numerical scheme very well captures the rightward shifting nature of fractional advection dispersion process than standard advection dispersion process ($\alpha = 2,\ \beta = 1$) at each time level. The role of α dominates over β in this process which is more effectively seen in Ex. 2.5.4.

- Fig. 2.14 verifies the numerical stability of our scheme. The behavior of absolute errors at final time $T = 1$ for $\alpha = 1.4$ and $\beta = 0.6$ without noise and with two different noisy inputs, $\delta^1 = 0.01\%\mu^{160}$ and $\delta^2 = \sigma^{160}$, in the initial data is shown in Fig. 2.14. The variation in absolute errors with noisy data is negligible as compared to without noisy data. Therefore, it can be concluded that the scheme (2.53) is numerically stable.

- Tables 2.6–2.9 show the L_2-error and spacial order of convergence of RFADE by the proposed scheme for various values of α and β. All tests have been performed with respect to L_2-norm at time $T = 0.4$ by using $M = 10$. From the results in Tables 2.6–2.9, we can conclude that the spacial order of convergence are of second order for all values of $\alpha \in (1, 2)$ and $\beta \in (0, 1)$.

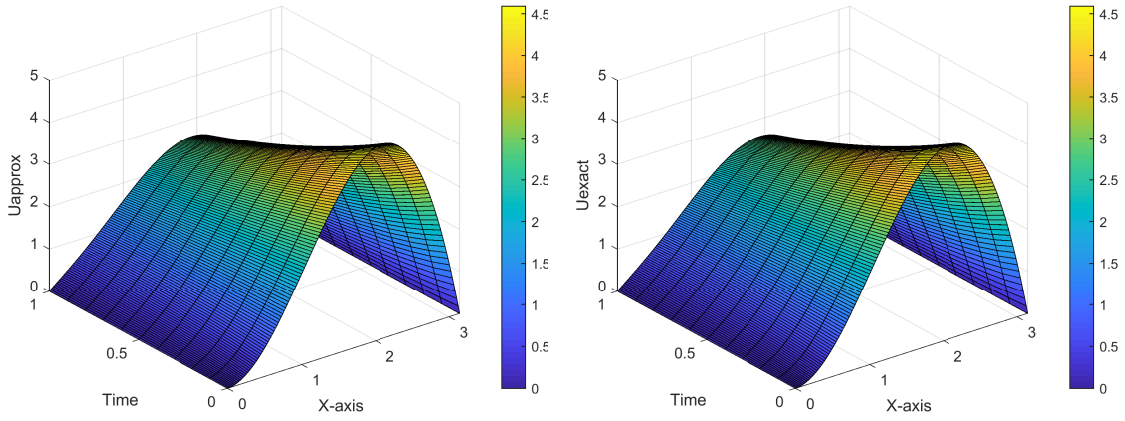

FIGURE 2.10: Surface plot of approximate solution (left) and exact solution (right) of Ex. 2.5.3 for $\alpha = 1.5$, $\beta = 0.5$, $h = \pi/160$ and $M = 10$ SLP basis.

- It can be seen from the Tables 2.6-2.7 that L_2 error obtained by using SCP is almost same as obtained by SLP but the CPU time taken by SLP basis function is very less as compared to SCP basis. Therefore the proposed numerical scheme is more efficient with SLP basis function.

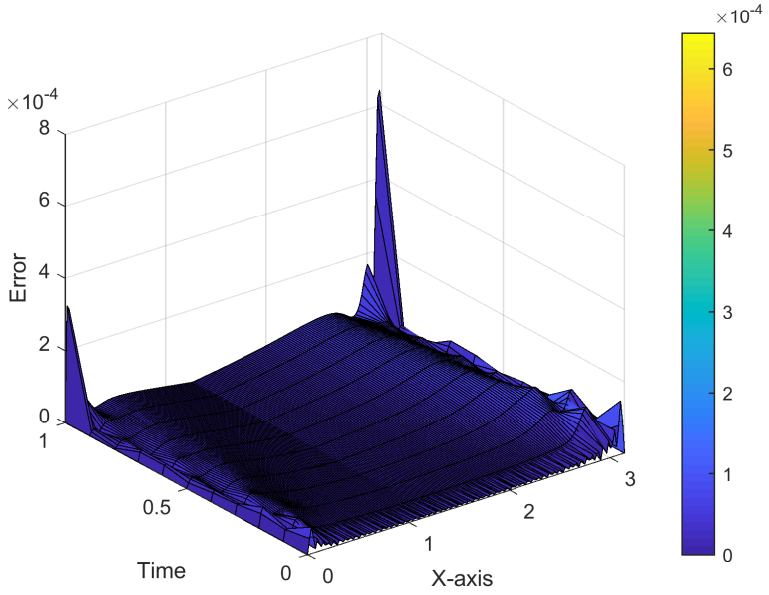

FIGURE 2.11: Surface plot of absolute errors of Ex. 2.5.3 for $\alpha = 1.5$, $\beta = 0.5$, $h = \pi/160$ and $M = 10$ SLP basis.

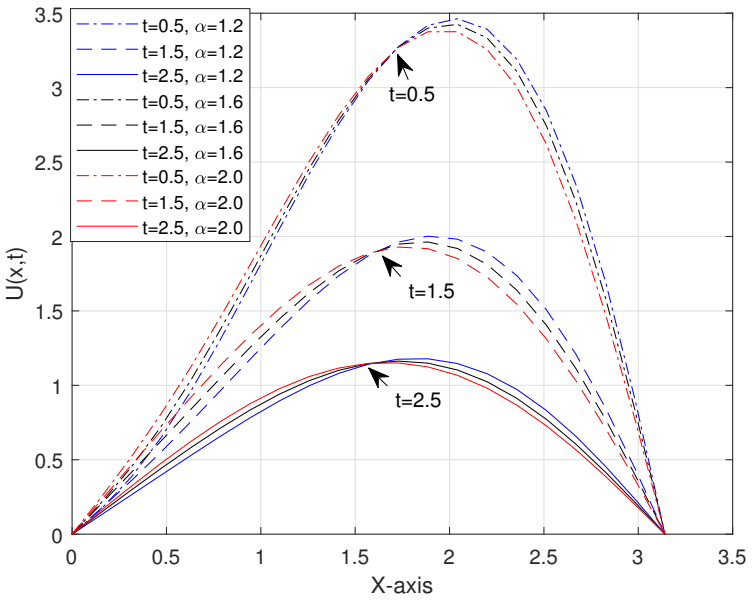

FIGURE 2.12: Approximate solution of Ex. 2.5.3 at various α and at various time level, when $\beta = 0.4$, $h = \pi/20$ and $M = 10$ SLP basis.

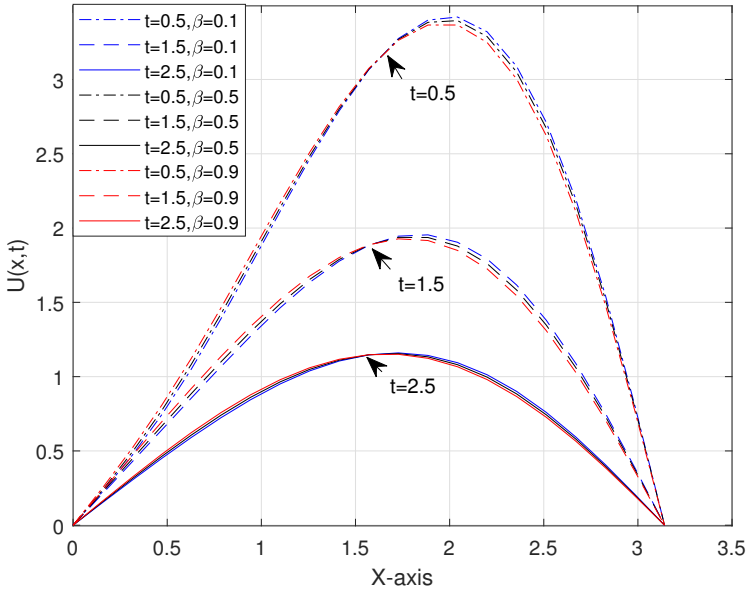

FIGURE 2.13: Approximate solution of Ex. 2.5.3 at various β and at various time level when $\alpha = 1.8$, $h = \pi/20$ and $M = 10$ SLP basis.

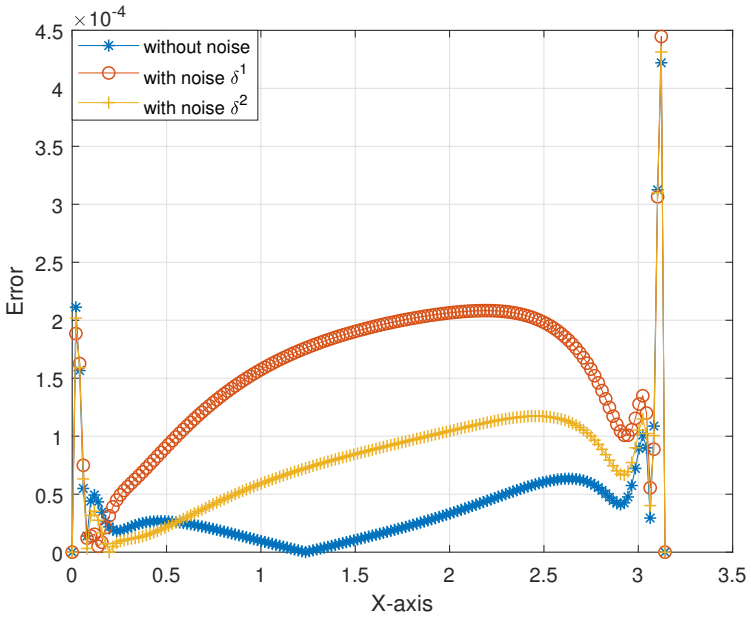

FIGURE 2.14: Absolute errors of Ex. 2.5.3 for $\alpha = 1.4$, $\beta = 0.6$, $h = \pi/160$, at $T = 1.0$, with different noisy input δ^i in the initial data, when $dt = 0.1$ and $M = 10$ SLP basis.

h	$\alpha = 1.1\ \beta = 0.1$			$\alpha = 1.1\ \beta = 0.5$			$\alpha = 1.1\ \beta = 0.9$		
	$\|u - U\|_2$	CO	CPU time	$\|u - U\|_2$	CO	CPU time	$\|u - U\|_2$	CO	CPU time
$\pi/5$	1.598e-02	-	5.02	2.068e-02	-	5.29	2.69e-02	-	5.26
$\pi/10$	6.615e-03	1.27	7.86	7.923e-02	1.38	7.98	1.00e-02	1.43	8.09
$\pi/20$	2.179e-03	1.60	15.52	2.445e-03	1.70	14.83	2.92e-02	1.78	15.44
$\pi/40$	5.934e-04	1.88	61.93	6.491e-04	1.91	62.09	7.54e-03	1.95	59.85
$\pi/80$	1.508e-04	1.98	292.95	1.642e-04	1.98	322.13	1.90e-04	1.99	345.36
$\pi/160$	3.782e-05	1.99	1931.83	4.115e-05	2.00	2000.74	4.75e-04	2.00	2253.46

TABLE 2.6: Errors, Convergence Order (CO) and CPU time (in seconds) by using $M = 10$ SLP basis function for Ex. 2.5.3 for $\alpha = 1.1$, $\beta = 0.1, 0.5, 0.9$ at time $T = 0.4$.

h	$\alpha = 1.1\ \beta = 0.1$			$\alpha = 1.1\ \beta = 0.5$			$\alpha = 1.1\ \beta = 0.9$		
	$\|u - U\|_2$	CO	CPU time	$\|u - U\|_2$	CO	CPU time	$\|u - U\|_2$	CO	CPU time
$\pi/5$	1.583e-02	-	27.15	2.068e-02	-	24.93	2.69e-02	-	24.52
$\pi/10$	6.614e-03	1.27	35.35	7.922e-02	1.38	30.78	1.00e-02	1.44	29.99
$\pi/20$	2.217e-03	1.60	80.96	2.446e-03	1.69	70.81	2.92e-03	1.78	74.53
$\pi/40$	5.931e-04	1.88	329.29	6.492e-04	1.91	309.27	7.54e-03	1.95	327.23
$\pi/80$	1.505e-04	1.99	6290.23	1.640e-04	1.89	6352.23	1.90e-04	1.99	6436.89

TABLE 2.7: Errors, Convergence Order (CO) and CPU time (in seconds) by using $M = 10$ SCP basis function for Ex. 2.5.3 for $\alpha = 1.1$, $\beta = 0.1, 0.5, 0.9$ at time $T = 0.4$.

h	$\alpha = 1.5,\ \beta = 0.1$		$\alpha = 1.5,\ \beta = 0.5$		$\alpha = 1.5,\ \beta = 0.9$	
	$\|u - U\|_2$	CO	$\|u - U\|_2$	CO	$\|u - U\|_2$	CO
$\pi/5$	2.630e-02	-	3.091e-02	-	3.738e-02	-
$\pi/10$	1.182e-02	1.15	1.261e-02	1.29	1.397e-02	1.42
$\pi/20$	3.359e-03	1.82	3.502e-03	1.85	3.767e-03	1.89
$\pi/40$	8.560e-03	1.97	8.890e-04	1.98	9.539e-04	1.98
$\pi/80$	2.150e-04	1.99	2.234e-04	1.99	2.394e-04	1.99
$\pi/160$	5.456e-05	1.98	5.673e-05	1.98	6.091e-05	1.97

TABLE 2.8: Errors and Convergence Order (CO) by using $M = 10$ SLP basis function for Ex. 2.5.3 for $\alpha = 1.5$, $\beta = 0.1, 0.5, 0.9$ at time $T = 0.4$.

h	$\alpha = 1.9,\ \beta = 0.1$		$\alpha = 1.9,\ \beta = 0.5$		$\alpha = 1.9,\ \beta = 0.9$	
	$\|u - U\|_2$	CO	$\|u - U\|_2$	CO	$\|u - U\|_2$	CO
$\pi/5$	4.551e-02	-	4.900e-02	-	5.440e-02	-
$\pi/10$	1.676e-02	1.44	1.721e-02	1.51	1.795e-02	1.60
$\pi/20$	4.367e-03	1.94	4.462e-03	1.95	4.635e-03	1.95
$\pi/40$	1.102e-03	1.99	1.124e-03	1.99	1.168e-03	1.99
$\pi/80$	2.813e-04	1.97	2.871e-04	1.97	2.983e-04	1.97
$\pi/160$	9.002e-05	1.64	9.185e-05	1.64	9.618e-05	1.63

TABLE 2.9: Errors and Convergence Order (CO) by using $M = 10$ SLP basis function for Ex. 2.5.3 for $\alpha = 1.9$, $\beta = 0.1, 0.5, 0.9$ at time $T = 0.4$.

To further demonstrate the role of fractional order α and β, we consider another example

Example 2.5.4. Consider the following RFADE

$$\frac{\partial}{\partial t}u(x,t) = k_\alpha \frac{\partial^\alpha}{\partial |x|^\alpha}u(x,t) + k_\beta \frac{\partial^\beta}{\partial |x|^\beta}u(x,t), \quad 1 < \alpha \leq 2, \quad 0 < \beta < 1, \tag{2.87}$$

with initial and boundary condition

$$u(x,0) = \sin(4x), \qquad 0 \leq x \leq \pi, \tag{2.88}$$

$$u(0,t) = u(\pi,t) = 0, \quad 0 \leq t \leq T. \tag{2.89}$$

From [72], we know the exact solution of the RFADE in Ex. 2.5.4 is given by

$$u(x,t) = \sum_{n=1}^{\infty} b_n \sin(nx) e^{-\left[(n^2)^{\frac{\alpha}{2}} k_\alpha + (n^2)^{\frac{\beta}{2}} k_\beta\right] t}, \tag{2.90}$$

where

$$b_n = \frac{2}{\pi}\int_0^{\pi} f(\xi)\sin(n\xi)d\xi. \tag{2.91}$$

The outcomes of Ex. 2.5.4 are presented as:

- Fig. 2.15 shows approximate solution of Ex. 2.5.4 for $\alpha = 1.5$, $\beta = 0.5$, $k_\alpha = k_\beta = 0.25$ and $M = 10$. The absolute errors shown in Fig. 2.16 verify that approximate solution by scheme (2.53) are in good agreement with exact solution at each time level.

- Fig. 2.17 demonstrate the case of standard advection dispersion equation ($\alpha = 2$, $\beta = 1$) of Ex. 2.5.4. From Figs. 2.15 and 2.17, we can say that the phenomena of fractional advection-dispersion process is slower than the standard advection dispersion process.

- Fig. 2.18 reflects the behavior of fractional order α, for a fix β and Fig. 2.19 reflects the behavior of fractional order β, for a fix α, at time $T = 0.5$. It can be seen that by increasing the value of fractional order α and β, the amplitude of sine wave decreases. Therefore, we can say that the advection-dispersion process is also proportional to the Riesz-space fractional derivative of order α and β. The dominant nature of fractional order α over β is clearly visible in this example.

- Tables 2.10–2.12 shows the L_2-error and second order of convergence of the proposed scheme with respect to L_2-norm for all values of $\alpha \in (1, 2]$ and $\beta \in (0, 1)$.

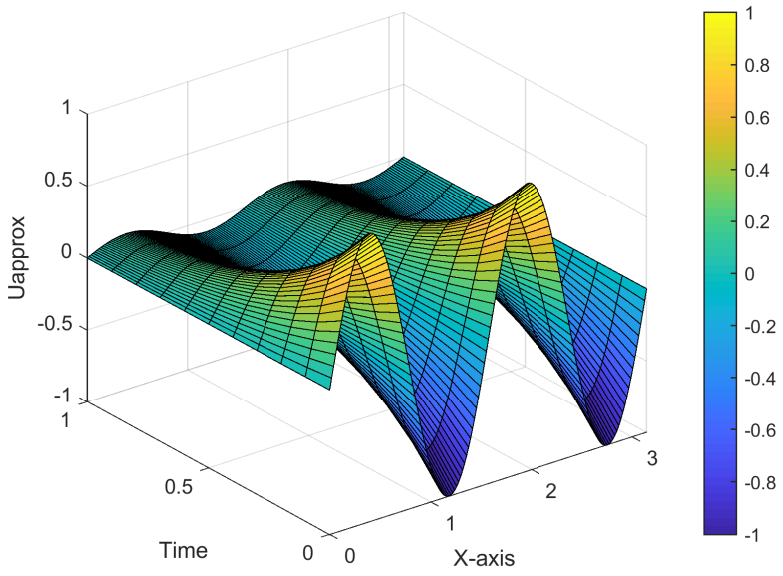

FIGURE 2.15: Surface plot of approximate solution of Ex. 2.5.4 for $\alpha = 1.5$, $\beta = 0.5$, $h = \pi/160$ and $M = 10$ SLP basis.

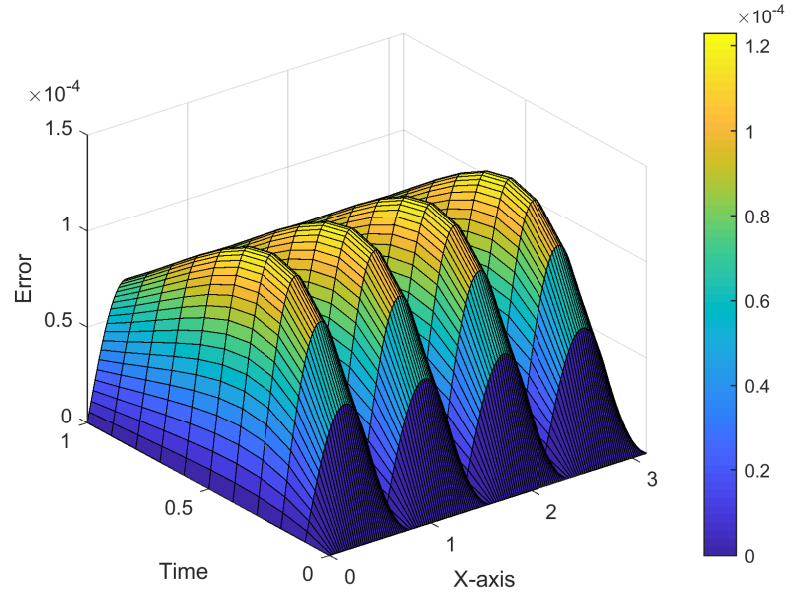

FIGURE 2.16: Surface plot of absolute error of Ex. 2.5.4 for $\alpha = 1.5$, $\beta = 0.5$, $h = \pi/160$ and $M = 10$ SLP basis.

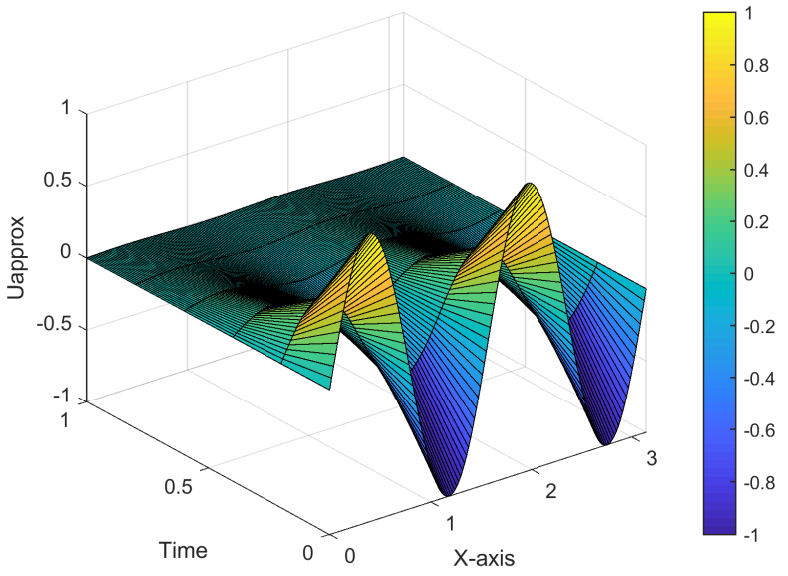

FIGURE 2.17: Surface plot of approximate solution of Ex. 2.5.4 for $\alpha = 2$, $\beta = 1$, i.e. standard advection-dispersion equation, $h = \pi/160$ and $M = 10$ SLP basis.

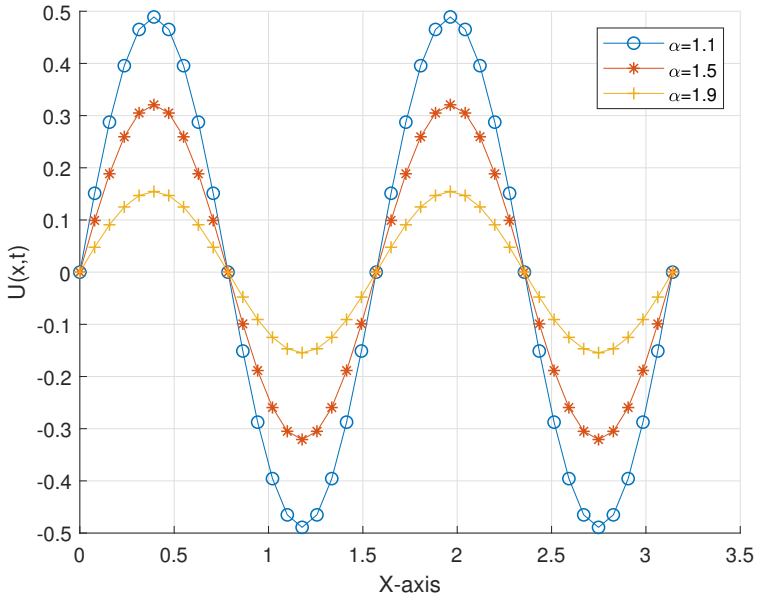

FIGURE 2.18: Approximate solution of Ex. 2.5.4 for $\alpha = 1.1, 1.5, 1.9$, $\beta = 0.1$, $h = \pi/40$ and $M = 10$ SLP basis.

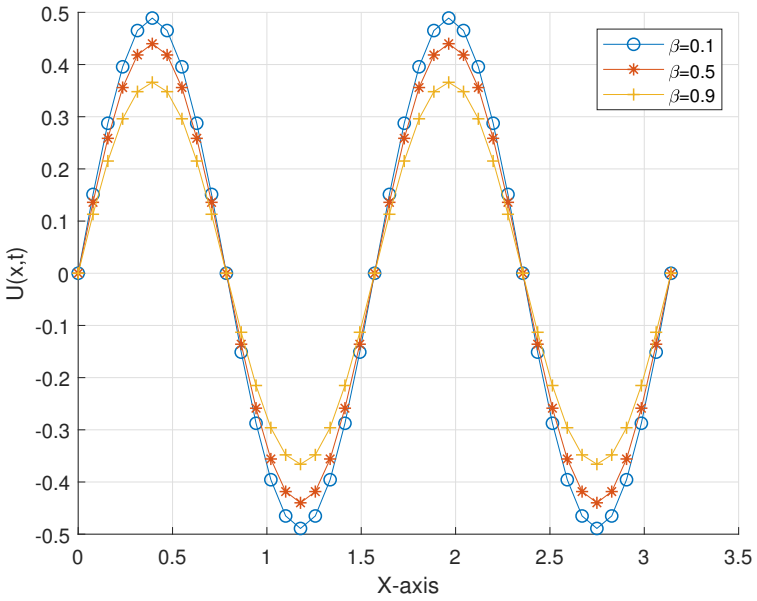

FIGURE 2.19: Approximate solution of Ex. 2.5.4 for $\alpha = 1.1$, $\beta = 0.1, 0.5, 0.9$, $h = \pi/40$ and $M = 10$ SLP basis.

h	$\alpha = 1.1,\ \beta = 0.1$		$\alpha = 1.1,\ \beta = 0.5$		$\alpha = 1.1,\ \beta = 0.9$	
	$\|u - U\|_2$	CO	$\|u - U\|_2$	CO	$\|u - U\|_2$	CO
$\pi/5$	9.352e-02	-	1.029e-01	-	1.227e-01	-
$\pi/10$	2.387e-02	1.97	2.581e-02	1.99	3.029e-02	2.02
$\pi/20$	5.992e-03	1.99	6.455e-03	2.00	7.539e-03	2.01
$\pi/40$	1.500e-04	2.00	1.614e-03	2.00	1.882e-03	2.00
$\pi/80$	3.750e-04	2.00	4.034e-04	2.00	4.705e-04	2.00
$\pi/160$	9.376e-05	2.00	1.009e-05	2.00	1.176e-05	2.00

TABLE 2.10: Errors and Convergence Order (CO) by using $M = 10$ SLP basis function for Ex 2.5.4 for $\alpha = 1.1$, $\beta = 0.1, 0.5, 0.9$ at time $T = 0.4$.

h	$\alpha = 1.5,\ \beta = 0.1$		$\alpha = 1.5,\ \beta = 0.5$		$\alpha = 1.5,\ \beta = 0.9$	
	$\|u - U\|_2$	CO	$\|u - U\|_2$	CO	$\|u - U\|_2$	CO
$\pi/5$	1.599e-01	-	1.610e-01	-	1.669e-01	-
$\pi/10$	4.014e-02	1.99	3.980e-02	2.02	4.025e-02	2.05
$\pi/20$	1.001e-03	2.00	9.880e-03	2.01	9.940e-03	2.02
$\pi/40$	2.502e-03	2.00	2.470e-03	2.00	2.477e-03	2.00
$\pi/80$	6.235e-04	2.00	6.160e-04	2.00	6.187e-04	2.00
$\pi/160$	1.563e-05	2.00	1.540e-05	2.00	1.546e-04	2.00

TABLE 2.11: Errors and Convergence Order (CO) by using $M = 10$ SLP basis function for Ex. 2.5.4 for $\alpha = 1.5$, $\beta = 0.1, 0.5$ & 0.9 at time $T = 0.4$.

h	$\alpha = 1.9,\ \beta = 0.1$		$\alpha = 1.9,\ \beta = 0.5$		$\alpha = 1.9,\ \beta = 0.9$	
	$\|u - U\|_2$	CO	$\|u - U\|_2$	CO	$\|u - U\|_2$	CO
$\pi/5$	2.160e-01	-	2.088e-02	-	2.011e-02	-
$\pi/10$	5.028e-02	2.10	4.793e-02	2.12	4.493e-02	2.16
$\pi/20$	1.225e-03	2.04	1.163e-03	2.04	1.084e-02	2.05
$\pi/40$	3.040e-03	2.01	2.886e-03	2.01	2.683e-03	1.94
$\pi/80$	7.587e-04	2.00	7.200e-04	2.00	6.692e-04	2.00
$\pi/160$	1.896e-04	2.00	1.700e-04	2.00	1.672e-04	2.00

TABLE 2.12: Errors and Convergence Order (CO) by using $M = 10$ SLP basis function for Ex. 2.5.4 for $\alpha = 1.9$, $\beta = 0.1, 0.5, 0.9$ at time $T = 0.4$.

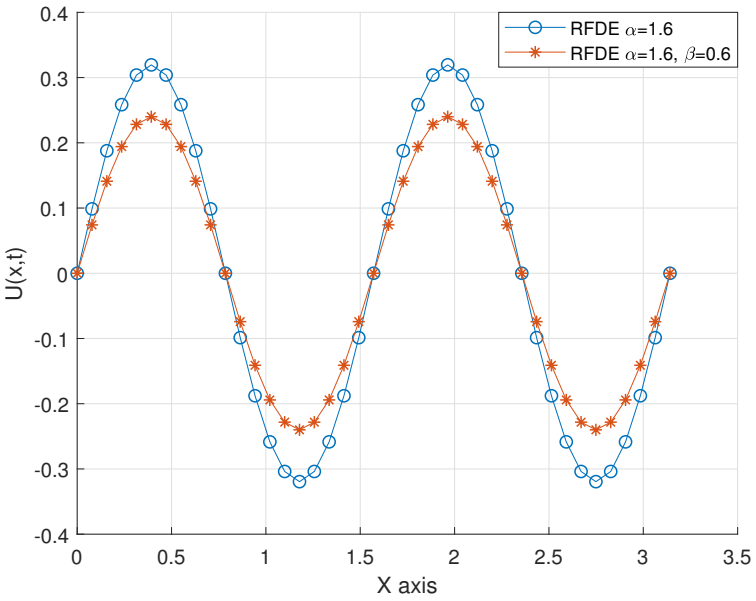

FIGURE 2.20: Comparison of RFDE Ex. 2.5.2 and RFADE Ex. 2.5.4 at $T = 0.5$, $h = \pi/40$ and $M = 10$ SLP basis.

Fig. 2.20 shows the role of fractional advection term in the RFADE. It shows that the diffusion process becomes faster after the addition of advection term in RFDE.

2.6 Appendix

2.6.1 Derivation of equation (2.39)

From equation (2.37)

$$[\mathrm{C}_i^T] = -k_\alpha \mathrm{A}\mathrm{C}_i^T \mathrm{K} - k_\alpha \mathrm{A}\mathrm{F}_i^T,$$

the above equation can be rewritten as

$$\begin{bmatrix} C_1^T \\ \vdots \\ C_{N-1}^T \end{bmatrix} = -k_\alpha \begin{bmatrix} a_{1,1} & \cdots & a_{N-1,1} \\ \vdots & \ddots & \vdots \\ a_{N-1,1} & \cdots & a_{1,1} \end{bmatrix} \begin{bmatrix} C_1^T K \\ \vdots \\ C_{N-1}^T K \end{bmatrix} - k_\alpha \begin{bmatrix} a_{1,1} & \cdots & a_{N-1,1} \\ \vdots & \ddots & \vdots \\ a_{N-1,1} & \cdots & a_{1,1} \end{bmatrix} \begin{bmatrix} F_1^T \\ \vdots \\ F_{N-1}^T \end{bmatrix}, \tag{2.92}$$

Now, putting the value of $[\mathrm{F}_i^T]$, $[\mathrm{C}_i^T]$ from (2.25) and $[\mathrm{C}_i^T \mathrm{K}]$ from (2.38) in above expression, we get

$$\begin{bmatrix} c_{10} & \cdots & c_{1M} \\ \vdots & \ddots & \vdots \\ c_{N-1,0} & \cdots & c_{N-1,M} \end{bmatrix} =$$

$$- k_\alpha \begin{bmatrix} a_{1,1} & \cdots & a_{N-1,1} \\ \vdots & \ddots & \vdots \\ a_{N-1,1} & \cdots & a_{1,1} \end{bmatrix} \begin{bmatrix} \sum_{j=0}^{M} c_{1j} k_{j0} & \cdots & \sum_{j=0}^{M} c_{1j} k_{jM} \\ \vdots & \ddots & \vdots \\ \sum_{j=0}^{M} c_{N-1j} k_{j0} & \cdots & \sum_{j=0}^{M} c_{N-1j} k_{jM} \end{bmatrix}$$

$$- k_\alpha \begin{bmatrix} a_{1,1} & \cdots & a_{N-1,1} \\ \vdots & \ddots & \vdots \\ a_{N-1,1} & \cdots & a_{1,1} \end{bmatrix} \begin{bmatrix} f_{10} & \cdots & f_{1M} \\ \vdots & \ddots & \vdots \\ f_{N-1,0} & \cdots & f_{N-1,M} \end{bmatrix},$$

after matrix multiplication, we get,

$$\begin{bmatrix} c_{10} & \cdots & c_{1M} \\ \vdots & \ddots & \vdots \\ c_{N-1,0} & \cdots & c_{N-1,M} \end{bmatrix} =$$

$$- k_\alpha \begin{bmatrix} \sum_{l=1}^{N-1} a_{1l} \left(\sum_{m=0}^{M} c_{lm} k_{m0} \right) & \cdots & \sum_{l=1}^{N-1} a_{1N-1} \left(\sum_{m=0}^{M} c_{lm} k_{mN-1} \right) \\ \vdots & \ddots & \vdots \\ \sum_{l=1}^{N-1} a_{N-1,l} \left(\sum_{m=0}^{M} c_{lm} k_{m0} \right) & \cdots & \sum_{l=1}^{N-1} a_{N-1,N-1} \left(\sum_{m=0}^{M} c_{lm} k_{mN-1} \right) \end{bmatrix}$$

$$- k_\alpha \begin{bmatrix} \sum_{l=1}^{N-1} a_{1l} f_{l0} & \cdots & \sum_{l=1}^{N-1} a_{1N-1} f_{l0} \\ \vdots & \ddots & \vdots \\ \sum_{l=1}^{N-1} a_{N-1,l} f_{l0} & \cdots & \sum_{l=1}^{N-1} a_{N-1,N-1} f_{l0} \end{bmatrix}. \tag{2.93}$$

comparing the coefficients of c_{ij} for $i = 1, 2, ..., N-1$ and $j = 0, 1, ..., M$ from (2.93), we get (2.39), i.e.

$$c_{ij} = -k_\alpha \sum_{l=1}^{N-1} a_{il} \left(\sum_{m=0}^{M} c_{lm} k_{mj} \right) - k_\alpha \sum_{l=1}^{N-1} a_{il} \left(f_{lj} \right), \quad i = 1, 2, ..., N-1 \quad j = 0, 1, ..., M,$$

where $a_{ij} = a_{ji}$ with $a_{ij} = a_{N-i+1,N-j+1}$ because A is symmetric as well as centro-symmetric matrix.

2.6.2 Derivation of equation (2.52)

From equation (2.51)

$$[\mathrm{C}_i^T] = -k_\alpha \mathrm{A}[\mathrm{C}_i^T \mathrm{K}] - k_\alpha \mathrm{A}[\mathrm{F}_i^T] - k_\beta \mathrm{B}[\mathrm{F}_i^T] - k_\beta \mathrm{B}[\mathrm{C}_i^T \mathrm{K}]. \tag{2.94}$$

Now, putting the value of $[\mathrm{F}_i^T]$, $[\mathrm{C}_i^T]$ from (2.25) and $[\mathrm{C}_i^T \mathrm{K}]$ from (2.38) in above expression and performing the same process as mentioned in 2.6.1 we get,

$$\begin{bmatrix} c_{10} & \cdots & c_{1M} \\ \vdots & \ddots & \vdots \\ c_{N-1,0} & \cdots & c_{N-1,M} \end{bmatrix} =$$

$$- k_\alpha \begin{bmatrix} \sum_{l=1}^{N-1} a_{1l} \left(\sum_{m=0}^{M} c_{lm} k_{m0} \right) & \cdots & \sum_{l=1}^{N-1} a_{1N-1} \left(\sum_{m=0}^{M} c_{lm} k_{mN-1} \right) \\ \vdots & \ddots & \vdots \\ \sum_{l=1}^{N-1} a_{N-1,l} \left(\sum_{m=0}^{M} c_{lm} k_{m0} \right) & \cdots & \sum_{l=1}^{N-1} a_{N-1,N-1} \left(\sum_{m=0}^{M} c_{lm} k_{mN-1} \right) \end{bmatrix}$$

$$- k_\alpha \begin{bmatrix} \sum_{l=1}^{N-1} a_{1l} f_{l0} & \cdots & \sum_{l=1}^{N-1} a_{1N-1} f_{l0} \\ \vdots & \ddots & \vdots \\ \sum_{l=1}^{N-1} a_{N-1,l} f_{l0} & \cdots & \sum_{l=1}^{N-1} a_{N-1,N-1} f_{l0} \end{bmatrix}$$

$$- k_\beta \begin{bmatrix} \sum_{l=1}^{N-1} b_{1l} \left(\sum_{m=0}^{M} c_{lm} k_{m0} \right) & \cdots & \sum_{l=1}^{N-1} b_{1N-1} \left(\sum_{m=0}^{M} c_{lm} k_{mN-1} \right) \\ \vdots & \ddots & \vdots \\ \sum_{l=1}^{N-1} b_{N-1,l} \left(\sum_{m=0}^{M} c_{lm} k_{m0} \right) & \cdots & \sum_{l=1}^{N-1} b_{N-1,N-1} \left(\sum_{m=0}^{M} c_{lm} k_{mN-1} \right) \end{bmatrix}$$

$$- k_\beta \begin{bmatrix} \sum_{l=1}^{N-1} b_{1l} f_{l0} & \cdots & \sum_{l=1}^{N-1} b_{1N-1} f_{l0} \\ \vdots & \ddots & \vdots \\ \sum_{l=1}^{N-1} b_{N-1,l} f_{l0} & \cdots & \sum_{l=1}^{N-1} b_{N-1,N-1} f_{l0} \end{bmatrix}. \tag{2.95}$$

comparing the coefficients of c_{ij} for $i = 1, 2, ..., N-1$ and $j = 0, 1, ..., M$ from (2.95), we get (2.52), i.e.

$$c_{ij} = -k_\alpha \sum_{l=1}^{N-1} a_{il} \left(\sum_{m=0}^{M} c_{lm} k_{mj} \right) - k_\alpha \sum_{l=1}^{N-1} a_{il} f_{lj} - k_\beta \sum_{l=1}^{N-1} b_{il} \left(\sum_{m=0}^{M} c_{lm} k_{mj} \right) - k_\beta \sum_{l=1}^{N-1} b_{il} f_{lj},$$

where $a_{ij} = a_{ji}$ with $a_{ij} = a_{N-i+1,N-j+1}$ for A and $b_{ij} = b_{ji}$ with $b_{ij} = b_{N-i+1,N-j+1}$ for B because A and B are symmetric as well as centro-symmetric matrix.

2.7 Conclusion

In summary, we have proposed two efficient numerical scheme for RFDE and RFADE, by applying a finite difference scheme based on MTM in spatial direction and a meshfree OMM based on SLP/SCP basis function in time direction. It is found from the study that the numerical solution at any time level does not depend on the numerical solution at previous time levels. The optimal error bound for the numerical approximation is investigated. Increasing the number of elements in basis functions improves the accuracy of the solution as α increases. Further, and it is found that the spatial convergence order of the proposed schemes are second order. Noise applied to the initial condition verifies the numerical stability of the schemes. A detailed numerical study of RFDEs and RFADEs, confirms the effectiveness and accuracy of the proposed schemes. It can be observed that both the basis function (SLP/SCP) gives almost the same accuracy but the CPU time taken by SLP is far less than SCP basis function. The proposed schemes very well capture the shifting nature of RFDE and RFADE and signify the role of fractional order α and β. It also justifies that the fractional-order processes are slower than the standard order processes. Hence, it is concluded that the numerical schemes with SLP basis are simple, fast, easy to implement and yield high accurate results. Application of these schemes can further be extended to the higher dimension for solving space fractional PDEs, which is one of our goals and a topic for future study.

****