Mathematical Analysis
& Convex Optimization

Vol. 4 (2023), No. 2, 127-141
https:\\ maco.lu.ac.ir
DOI: 10.22034/maco.4.2.12

Research Paper

SOLUTION OF THE TIME- AND RIESZ SPACE-FRACTIONAL FOKKER-PLANCK EQUATION BY A STABLE GAUSSIAN RADIAL BASIS FUNCTION METHOD

Z. SHAMOHAMMADI[1], AND B. SEPEHRIAN[1,*]

Abstract. In this article the Caputo time- and Riesz space-fractional Fokker-Planck equation (TSFFPE) is solved by the stable Gaussian radial basis function (RBF) method. By a spatial discretization and using the Riesz fractional derivative of the stable Gaussian radial basis function interpolants computed in [23], the computations of TSFFPE reduced to a system of fractional ODEs. A high order finite difference method is presented for this system of ODEs, and the computations are converted to a system of linear or nonlinear algebraic equations, in each time step. In the nonlinear case, these systems can be easily solved by the Newton iterative method. Numerical illustrations are performed to confirm the accuracy and efficiency of the presented method. Some comparisons are made with the results in other literature .

MSC(2010): 35G16, 65N06.
Keywords: Caputo derivative; Fractional Fokker-Planck equation; Radial basis functions; Riesz derivative.

1. Introduction and Background

The Fokker-Planck equation

$$\frac{\partial u(x,t)}{\partial t} = \left[\frac{\partial}{\partial x}\frac{V'(x)}{m\eta_1} + \kappa_1 \frac{\partial^2}{\partial x^2}\right]u(x,t), \tag{1.1}$$

was introduced to describe the Brownian motion of an article in the presence of an external force field $F(x) = -V'(x)$ [21]. In (1) $u(x,t)$ is the probability density function of finding the particle at position x and at the given time t, m is the mass of particle, κ_1 denotes the markovian diffusion constant, and η_1 denotes the friction coefficient which is a measure for interaction of the particle with its medium. The mean square displacement for markovian diffusion process is as $< r^2(t) >= 2\kappa_1 t$. For highly non-homogeneous mediums, the diffusion process is not markovian and is named anomalous diffusion. In fact for anomalous process $< r^2(t) >= \frac{2\kappa_\alpha}{\Gamma(1+\alpha)}t^\alpha$, $\alpha \neq 1$, in which κ_α denotes the anomalous diffusion coefficient, and α is the anomalous diffusion exponent and ranges from 0 to 1 for subdiffusion and more than 1 for super diffusion. It is observed that the fractional Fokker-Planck equations (FFPEs) are adequate to model the particle transport in anomalous diffusion [14]. In [14, 27] the authors showed that the time-fractional Fokker-Planck equation (TFFPE) depicts traps, the

---

Date: Received: May 4, 2023, Accepted: Novevmer 17, 2023.

[*] Corresponding author.

space-fractional Fokker-Planck equation (SFFPE) describes the Lévy flights, and TSFFPE characterize the competition between traps and Lévy flights.
In this article, we consider TSFFPE with a source term as [26]

$$ {}^{c}D_t^{\alpha}\ u(x,t) = \left[\frac{\partial}{\partial x}\frac{V'(x)}{m\eta_\alpha} + \kappa_\alpha^{\gamma}\frac{\partial^\gamma}{\partial|x|^\gamma}\right]u(x,t) + s\Big(x,t,u(x,t)\Big),\ a \le x \le b, \tag{1.2}$$

with the initial condition

$$ u(x,0) = g(x),\ a \le x \le b, \tag{1.3}$$

and boundary conditions

$$ u(a,t) = h_1(t),\ \ 0 < t \le T, \tag{1.4}$$

$$ u(b,t) = h_2(t),\ 0 < t \le T, \tag{1.5}$$

where $f(x) = \dfrac{V'(x)}{m\eta_\alpha}$ is known as the drift coefficient and κ_α^γ denotes the anomalous diffusion coefficient. ${}^{c}D_t^{\alpha}$ is the Caputo fractional derivative of order $\alpha \in (0,1)$ which is defined by

$$ {}^{c}D_t^{\alpha}\ u(x,t) = \frac{1}{\Gamma(1-\alpha)}\int_0^t (t-\zeta)^{-\alpha}\ \frac{\partial u(x,\zeta)}{\partial \zeta}d\zeta. \tag{1.6}$$

Moreover, $\dfrac{\partial^\gamma}{\partial|x|^\gamma}$ is the Riesz derivative of order $\gamma \in (1,2)$ and is defined as

$$ \frac{\partial^\gamma u(x,t)}{\partial|x|^\gamma} = -c_\gamma\Big[{}_aD_x^\gamma u(x,t) + {}_xD_b^\gamma u(x,t)\Big], \tag{1.7}$$

where $c_\gamma = \dfrac{1}{2\cos(\frac{\pi\gamma}{2})}$. Also,

$$ {}_aD_x^\gamma u(x,t) = \frac{1}{\Gamma(2-\gamma)}\frac{d^2}{dx^2}\int_a^x (x-\zeta)^{1-\gamma}u(\zeta,t)d\zeta,\ x > a, \tag{1.8}$$

and

$$ {}_xD_b^\gamma u(x,t) = \frac{1}{\Gamma(2-\gamma)}\frac{d^2}{dx^2}\int_x^b (\zeta-x)^{1-\gamma}u(\zeta,t)d\zeta,\ x < b, \tag{1.9}$$

are the left- and right-sided Riemann-Liouville fractional derivatives, respectively. Clearly, these fractional derivatives are linear operators.

Lemma 1.1. For the Riemann-Liouville fractional derivatives we have [17]

$$ {}_aD_x^\gamma 1 = \frac{1}{\Gamma(1-\gamma)}(x-a)^{-\gamma}, \tag{1.10}$$

$$ {}_xD_b^\gamma 1 = \frac{1}{\Gamma(1-\gamma)}(b-x)^{-\gamma}. \tag{1.11}$$

Theorem 1.2. The Riemann-Liouville fractional derivatives of the power functions satisfy [17]

$$ {}_aD_x^\gamma\ (x-a)^p = \frac{\Gamma(p+1)}{\Gamma(p+1-\gamma)}(x-a)^{p-\gamma}, \tag{1.12}$$

$$ {}_xD_b^\gamma (b-x)^p = \frac{\Gamma(p+1)}{\Gamma(p+1-\gamma)}(b-x)^{p-\gamma}, \tag{1.13}$$

where $n-1<\gamma<n$, $p>-1$ and $p\in\mathbb{R}$.

There is not any method for finding the exact solution of FFPEs except for special cases [1, 11]. That's why several analytical and numerical methods for solving the FFPEs have been proposed in some literature. In [9], an analytic solution for the linear and nonlinear TSFFPEs was derived by the homotopy perturbation method. In [29], the authors presented an analytic solutions for a type of SFFPE. A numerical solution of TFFPEs was gained by a spectral collocation method in [25]. Xie *et al.* [24] applied the Chebyshev wavelets for solving TSFFPEs with variable coefficients. In [26] a finite difference scheme, and in [7] a finite elements method for solving the Caputo time- and Riesz space-FFPE were introduced. For more study, e.g., see [6, 15, 20, 22, 28].

There are only a few methods for solving nonlinear space- and time-fractional PDEs. On the other hand, it is common to use the finite difference and finite elements methods in order to discretizing the fractional derivatives. But, the fractional derivatives are non-local differential operators, thus the non-local methods such as the radial basis functions (RBFs) method are more efficient for discretizing them. Moreover, the RBFs methods are usually more accurate than those methods, because the interpolating of smooth data using global, infinitely differentiable RBFs has a spectral accuracy [2, 4, 12, 13]. Also unlike those methods, the RBFs methods are efficient for problems with irregular domain, because no mesh generation is need in RBFs methods [16]. However, only a few RBFs methods have been presented to solve fractional PDEs. For these reasons, we were motivated to propose a RBF method to solve TSFFPE.

The RBFs may be applied to interpolate a function $f(\mathbf{x})$ at the distinct points $\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_M$ as

$$f(\mathbf{x})\approx\sum_{i=1}^{M}\lambda_i\phi_i(\mathbf{x}),\quad \mathbf{x}\in\mathbb{R}^d, \tag{1.14}$$

in which $\phi_i(\mathbf{x})=\phi(\|\mathbf{x}-\mathbf{x}_i\|_2)$ is radial basis function, λ_i's are scalars to be determined in such a way that Eq. (14) is satisfied as equality for $\mathbf{x}_j$'s, and d is the dimension of problem. Thus a linear system of algebraic equations is obtained as

$$A\Lambda=b,$$

in which $\Lambda=(\lambda_1,\lambda_2,...,\lambda_M)^T$ is an unknown vector to be determined,

$$b=[f(\mathbf{x}_1),f(\mathbf{x}_2),...,f(\mathbf{x}_M)]^T$$

is the right-hand side vector, and the RBF interpolation matrix is given by

$$A=[\Phi_{ij}]=[\phi(\|\mathbf{x}_i-\mathbf{x}_j\|_2)]_{1\leq i,j\leq M}.$$

The coefficients matrix, A, has usually a very large condition number i.e. A is very ill-conditioned. This is a major problem in RBF interpolation.

In this study, we use the Gaussian basis functions

$$\phi(r)=e^{-\varepsilon r^2}, \tag{1.15}$$

where, ε is called shape parameter that controls the flatness of the function. As the shape parameter becomes smaller, a better accuracy is obtained. But the smaller shape parameter causes the condition number of the interpolation matrix rapidly increase. To solve this issue,

Fasshauer *et al.* proposed a stable method to compute and evaluate the Gaussian RBF interpolants [10]. Their method was based on Mercer's theorem and the eigenfunction expansion of the Gaussian RBF. They showed that the eigenfunctions of the Gaussian RBF can be written in terms of Hermite polynomials. Laterally, the authors in [18] showed that these eigenfunctions can be rewritten in terms of the shifted Chebyshev polynomials, and this can computationally improve the RBF interpolation.
In this work, we apply the Gaussian RBF interpolant with the Chebyshev polynomials type eigenfunctions, for TSFFPE (1). Since the spatial derivative is of the fractional type, so for our development we apply the left- and right-sided Riemann-Liouville fractional derivatives of the eigenfunctions of the Gaussian RBF computed in [23]. Then, by a spatial discretization, we reduce the problem to a system of fractional ODEs. To solve this system of ODEs, we propose a high order finite difference scheme. If the source term, $s(x,t,u)$, in (1), is a linear (nonlinear) function of u, these ODEs are linear (nonlinear), and our finite difference scheme leads to a linear (nonlinear) system of algebraic equations, in each time step.

Other sections of the paper are as follows: In section 2, Chebyshev polynomials and stable Gaussian RBF interpolation are expressed and, the left- and right-sided Riemann-Liouville derivatives of the eigenfunction expansions based on Chebyshev polynomials are presented. In section 3, we describe our method for solving Eq. (1.2). Several numerical examples are provided in section 4 to certify the efficiency and accuracy of the new method.

2. Preliminaries

2.1. Chebyshev polynomials.

The Chebyshev polynomial of degree n for $n=0$, $n=1$ and $n\geq 2$, is as follows: $T_0(x)=1$, $T_1(x)=x$ and

$$T_{n+1}(x)=2xT_n(x)-T_{n-1}(x),\ n=1,2,\dots$$

where $x\in[-1,1]$. For $n=1,2,\dots$, the function $T_n(x)$ has $n+1$ extrema points as

$$x_j=\cos\frac{\pi(j-1)}{n},\ j=1,2,\dots,n+1.$$

2.2. Stable Gaussian RBF interpolation.

The Gaussian RBF as a positive definite kernel can be expanded in terms of positive eigenvalues λ_n and normalized eigenfunctions ϕ_n as [10, 18]

$$e^{-\varepsilon(x-z)^2}=\sum_{n=1}^{\infty}\lambda_n\phi_n(x)\phi_n(z), \tag{2.1}$$

in which the functions

$$\phi_n(x)=\sqrt{\beta}e^{-\delta^2x^2}\tilde{H}_{n-1}(\sigma\beta x),\quad n=1,2,\cdots \tag{2.2}$$

are the orthogonal functions with respect to the weight function $w(x)=\frac{\sigma}{\sqrt{\pi}}e^{-\sigma^2x^2}$, $\sigma>0$, and $\tilde{H}_n(x)$ are normalized Hermite polynomials. Moreover,

$$\beta=\Big(1+\frac{4\varepsilon^2}{\sigma^2}\Big)^{\frac{1}{4}},\ \delta^2=\frac{\sigma^2}{2}(\beta^2-1),$$

and

$$\lambda_n=\sqrt{\frac{\sigma^2}{\sigma^2+\varepsilon^2+\delta^2}}\big(\frac{\varepsilon^2}{\sigma^2+\varepsilon^2+\delta^2}\big)^{n-1}.$$

The Gaussian RBF interpolant of $f(x)$ at $x_1, x_2, \cdots, x_M$ is as

$$s_f(x) = \sum_{j=1}^{M} c_j e^{-\varepsilon(x-x_j)^2}, \tag{2.3}$$

where c_j's are scalars to be determined in such a way that the interpolation conditions $s_f(x_i) = f(x_i)$, $i = 1, \dots, M$ are satisfied.

In practice by choosing M terms of the series in (2.1), we approximate $e^{-\varepsilon(x-z)^2}$, and consequently we can rewrite (2.3) as

$$s_f(x) = \sum_{j=1}^{M} c_j \sum_{n=1}^{M} \lambda_n \phi_n(x) \phi_n(x_j). \tag{2.4}$$

In [5, 18] the authors showed that $s_f(x)$ can be written independent of coefficients c_j and eigenvalues λ_n as

$$s_f(x) = V_\phi^T(x)\ \Phi_{\mathbf{X}}^{-T}\ \mathbf{f}, \tag{2.5}$$

where $V_\phi^T(x) = [\phi_1(x), ..., \phi_M(x)]$, $\mathbf{f} = [f(x_1), ..., f(x_M)]^T$, and

$$\Phi_{\mathbf{X}} = \begin{bmatrix} \phi_1(x_1) & \dots & \phi_1(x_M) \\ \vdots & & \vdots \\ \phi_M(x_1) & \dots & \phi_M(x_M) \end{bmatrix}. \tag{2.6}$$

The Hermite polynomials can grow dramatically and it can lead to instability in our computations. For this reason the authors of [19] rebuilt the eigenfunctions as

$$\phi_n(x) = \sqrt{\beta} e^{-\delta^2 x^2} \hat{T}_{n-1}(x),\ n = 1, \dots, M, \tag{2.7}$$

in which $\hat{T}_{n-1}$'s are the shifted Chebyshev polynomials on the interval $[0, 1]$ that are given by

$$\hat{T}_n(x) = \begin{cases} \dfrac{1}{\sqrt{M}}, & n = 0, \\ \sqrt{\dfrac{2}{M}}\ T_n(2x-1), & n \geq 1. \end{cases}$$

The shifted Chebyshev polynomial of degree n can be presented by the analytical form [8]

$$\hat{T}_n(x) = \begin{cases} \dfrac{1}{\sqrt{M}}, & n = 0, \\ \sqrt{\dfrac{2}{M}}\ n \displaystyle\sum_{i=0}^{n} (-1)^{n-i} \frac{2^{2i}(n+i-1)!}{(2i)!(n-i)!} x^i, & n \geq 1. \end{cases} \tag{2.8}$$

In our development we apply the Gaussian RBF interpolant (2.5) with the eigenfunctions (2.7).

The left- and right-sided Riemann-Liouville fractional derivative of $\phi_n(x)$, $n = 1, 2, \cdots$ are as follows [23]

$${}_aD_x^\gamma \phi_1(x) = \sqrt{\frac{\beta}{M}} \sum_{k=0}^{\infty} \frac{(-\delta^2)^k}{k!} \sum_{j=0}^{2k} \binom{2k}{j} a^{2k-j}\ \frac{\Gamma(j+1)}{\Gamma(j+1-\gamma)} (x-a)^{j-\gamma}, \tag{2.9}$$

$$ {}_xD_b^{\gamma}\phi_1(x) = \sqrt{\frac{\beta}{M}}\sum_{k=0}^{\infty}\frac{(-\delta^2)^k}{k!}\sum_{j=0}^{2k}\binom{2k}{j}b^{2k-j}(-1)^j\,\frac{\Gamma(j+1)}{\Gamma(j+1-\gamma)}(b-x)^{j-\gamma}, \tag{2.10} $$

and for $n \geq 2$,

$$ \begin{aligned} {}_aD_x^{\gamma}\phi_n(x) = \sqrt{\frac{2\beta}{M}}\sum_{k=0}^{\infty}\frac{(-\delta^2)^k}{k!}\;(n-1)\sum_{i=0}^{n-1}(-1)^{n-i-1}\frac{2^{2i}(n+i-2)!}{(2i)!(n-i-1)!} \\ \sum_{j=0}^{i+2k}\binom{i+2k}{j}a^{i+2k-j}\,\frac{\Gamma(j+1)}{\Gamma(j+1-\gamma)}(x-a)^{j-\gamma}, \end{aligned} \tag{2.11} $$

and

$$ \begin{aligned} {}_xD_b^{\gamma}\phi_n(x) = \sqrt{\frac{2\beta}{M}}\sum_{k=0}^{\infty}\frac{(-\delta^2)^k}{k!}\;(n-1)\sum_{i=0}^{n-1}(-1)^{n-i-1}\frac{2^{2i}(n+i-2)!}{(2i)!(n-i-1)!} \\ \sum_{j=0}^{i+2k}\binom{i+2k}{j}b^{i+2k-j}(-1)^j\frac{\Gamma(j+1)}{\Gamma(j+1-\gamma)}(b-x)^{j-\gamma}. \end{aligned} \tag{2.12} $$

3. Solution of TSFFPE

Here, we present a method to solve Eq. (1.2). For this purpose, we substitute (1.7) in (1.2) and rewrite it as follows:

$$ {}^cD_t^{\alpha}\ u(x,t) = f'(x)u(x,t) + f(x)\frac{\partial u(x,t)}{\partial x} - \kappa_\alpha^\gamma c_\gamma\Big[{}_aD_x^{\gamma}u(x,t) + {}_xD_b^{\gamma}u(x,t)\Big] + s\Big(x,t,u(x,t)\Big), \tag{3.1} $$

in which, $f'(x)$ is the first order derivative of $f(x)$.
Now, we consider the distinct points $x_1, x_2, \ldots, x_M$ where $x_1 = a$ and $x_M = b$ are boundary points, and we discretize Eq. (3.1) in the interior points $x_2, \ldots, x_{M-1}$ as

$$ \begin{aligned} {}^cD_t^{\alpha}\ u_i(t) = f'(x_i)u_i(t) + f(x_i)\frac{\partial u(x,t)}{\partial x}|_{x=x_i} - \kappa_\alpha^\gamma c_\gamma\Big[{}_aD_x^{\gamma}u(x,t) \\ + {}_xD_b^{\gamma}u(x,t)\Big]_{x=x_i} + s\Big(x_i,t,u_i(t)\Big), \end{aligned} \tag{3.2} $$

in which $u_i(t) = u(x_i,t)$.
Using Eq. (2.5), we write

$$ u(x,t) \approx V_\phi^T(x)\ \Phi_X^{-T}\ U(t), \tag{3.3} $$

where $U(t) = \Big[u_1(t), \ldots, u_M(t)\Big]^T$.
Replacing Eq. (3.3) in (3.2), gives

$$ \begin{aligned} {}^cD_t^{\alpha}u_i(t) = f'(x_i)u_i(t) + f(x_i)V_\phi'^T(x_i)\ \Phi_X^{-T}\ U(t) - \kappa_\alpha^\gamma c_\gamma\Big[{}_aD_x^{\gamma}V_\phi^T(x_i) + {}_xD_b^{\gamma}V_\phi^T(x_i)\Big]\Phi_X^{-T}U(t) \\ + s\Big(x_i,t,u_i(t)\Big),\ \ i = 2, \ldots, M-1 \end{aligned} \tag{3.4} $$

in which the vectors

$${}_aD_x^{\gamma}V_{\phi}^T(x_i) = \Big[{}_aD_x^{\gamma}\phi_1(x_i), ..., {}_aD_x^{\gamma}\phi_M(x_i)\Big],$$

and

$${}_xD_b^{\gamma}V_{\phi}^T(x_i) = \Big[{}_xD_b^{\gamma}\phi_1(x_i), ..., {}_xD_b^{\gamma}\phi_M(x_i)\Big],$$

are obtained by (2.9), (2.10), (2.11) and (2.12).
Eq. (3.4) gives a system of fractional ODEs in unknowns $u_2(t), \dots, u_{M-1}(t)$. We obtain the structure of this system as follows: We decompose $U(t)$ to two vectors $U_{\mathcal{B}}(t)$ and $U_{\mathcal{I}}(t)$ that are the boundrary and interior entries, respectively. We set

$$\tilde{\Psi}_X^T = \begin{bmatrix} f(x_2){V'}_{\phi}^T(x_2) \\ \vdots \\ f(x_{M-1}){V'}_{\phi}^T(x_{M-1}) \end{bmatrix}, \; {}_a\tilde{\Psi}_X^T = \begin{bmatrix} {}_aD_x^{\gamma}V_{\phi}^T(x_2) \\ \vdots \\ {}_aD_x^{\gamma}V_{\phi}^T(x_{M-1}) \end{bmatrix}, \; {}_b\tilde{\Psi}_X^T = \begin{bmatrix} {}_xD_b^{\gamma}V_{\phi}^T(x_2) \\ \vdots \\ {}_xD_b^{\gamma}V_{\phi}^T(x_{M-1}) \end{bmatrix},$$

$\dot{A} = [\dot{a}_{ij}] = \tilde{\Psi}_X^T\ \Phi_X^{-T}$, $\bar{\bar{A}} = [\bar{\bar{a}}_{ij}] = {}_a\tilde{\Psi}_X^T\ \Phi_X^{-T}$ and $\bar{A} = [\bar{a}_{ij}] = {}_b\tilde{\Psi}_X^T\ \Phi_X^{-T}$. So, we have

$$\tilde{\Psi}_X^T\ \Phi_X^{-T}U(t) = \underbrace{\begin{bmatrix} \dot{a}_{1,1} & \dot{a}_{1,M} \\ \dot{a}_{2,1} & \dot{a}_{2,M} \\ \vdots & \vdots \\ \dot{a}_{M-2,1} & \dot{a}_{M-2,M} \end{bmatrix}}_{=\dot{A}_{\mathcal{B}}} \underbrace{\begin{bmatrix} u_1(t) \\ u_M(t) \end{bmatrix}}_{=U_{\mathcal{B}}(t)}$$

$$+ \underbrace{\begin{bmatrix} \dot{a}_{1,2} & \dots & \dot{a}_{1,M-1} \\ \dot{a}_{2,2} & \dots & \dot{a}_{2,M-1} \\ \vdots & \dots & \vdots \\ \dot{a}_{M-2,2} & \dots & \dot{a}_{M-2,M-1} \end{bmatrix}}_{=\dot{A}_{\mathcal{I}}} \underbrace{\begin{bmatrix} u_2(t) \\ \vdots \\ u_{M-1}(t) \end{bmatrix}}_{=U_{\mathcal{I}}(t)},$$

where $U_{\mathcal{B}}(t)$ is given by boundary conditions (4) and (5), and $U_{\mathcal{I}}(t)$ is unknown.
Similarly, ${}_a\tilde{\Psi}_X^T\ \Phi_X^{-T}U(t) = \bar{\bar{A}}_{\mathcal{I}}U_{\mathcal{I}}(t) + \bar{\bar{A}}_{\mathcal{B}}U_{\mathcal{B}}(t)$ and ${}_b\tilde{\Psi}_X^T\ \Phi_X^{-T}U(t) = \bar{A}_{\mathcal{I}}U_{\mathcal{I}}(t) + \bar{A}_{\mathcal{B}}U_{\mathcal{B}}(t)$ are gained.

Replacing the obtained relations in (3.4), we get

(3.5)
$${}^cD_t^{\alpha}U_{\mathcal{I}}(t) + \Big[\kappa_{\alpha}^{\gamma}c_{\gamma}(\bar{\bar{A}}_{\mathcal{I}} + \bar{A}_{\mathcal{I}}) - \dot{A}_{\mathcal{I}} - F'\Big]U_{\mathcal{I}}(t) + \Big[\kappa_{\alpha}^{\gamma}c_{\gamma}(\bar{\bar{A}}_{\mathcal{B}} + \bar{A}_{\mathcal{B}}) - \dot{A}_{\mathcal{B}}\Big]U_{\mathcal{B}}(t) - S(t, U_{\mathcal{I}}(t)) = 0,$$

in which F' is a diagonal matrix as

$$F' = diag\Big(f'(x_2), \dots, f'(x_{M-1})\Big),$$

and

$$S(t, U_{\mathcal{I}}(t)) = \Big[s(x_2, t, u_2(t)), \dots, s(x_{M-1}, t, u_{M-1}(t))\Big]^T.$$

To solve the fractional system (3.5), we discretize it in the time direction as

$$\begin{aligned} &{}^cD_t^\alpha U_{\mathcal{I}}(t^n) + \Big[\kappa_\alpha^\gamma c_\gamma(\bar{\bar{A}}_{\mathcal{I}} + \bar{A}_{\mathcal{I}}) - \dot{A}_{\mathcal{I}} - F'\Big] U_{\mathcal{I}}(t^n) + \\ &\Big[\kappa_\alpha^\gamma c_\gamma(\bar{\bar{A}}_{\mathcal{B}} + \bar{A}_{\mathcal{B}}) - \dot{A}_{\mathcal{B}}\Big] U_{\mathcal{B}}(t^n) - S\Big(t^n, U_{\mathcal{I}}(t^n)\Big) = 0, \end{aligned} \tag{3.6}$$

where $t^n = n\tau$ for $n = 0, 1, \ldots, N$, and τ is time step size. Now, we approximate ${}^cD_t^\alpha U_{\mathcal{I}}(t^n)$ for $n = 1$, $n = 2$ and $n \geq 3$ by the method presented in [3] as

$${}^cD_t^\alpha U_{\mathcal{I}}(t^1) = \mu a_0\Big(U_{\mathcal{I}}(t^1) - U_{\mathcal{I}}(t^0)\Big) + O(\tau^{2-\alpha}), \tag{3.7}$$

$${}^cD_t^\alpha U_{\mathcal{I}}(t^2) = \mu\Big[(b_0 - a_1)U_{\mathcal{I}}(t^0) + (a_1 - a_0 - 2b_0)U_{\mathcal{I}}(t^1) + (a_0 + b_0)U_{\mathcal{I}}(t^2)\Big] + O(\tau^{3-\alpha}), \tag{3.8}$$

$$\begin{aligned} &{}^cD_t^\alpha U_{\mathcal{I}}(t^n) = \mu\Big[(b_{n-2} - a_{n-1})U_{\mathcal{I}}(t^0) + (a_{n-1} - a_{n-2} - 2b_{n-2})U_{\mathcal{I}}(t^1) + (a_{n-2} + b_{n-2})U_{\mathcal{I}}(t^2) \\ &+ \sum_{k=3}^{n-1}\Big(w_{1,n-k}U_{\mathcal{I}}(t^k) + w_{2,n-k}U_{\mathcal{I}}(t^{k-1}) + w_{3,n-k}U_{\mathcal{I}}(t^{k-2}) + w_{4,n-k}U_{\mathcal{I}}(t^{k-3})\Big) \\ &+ w_{1,0}U_{\mathcal{I}}(t^n) + w_{2,0}U_{\mathcal{I}}(t^{n-1}) + w_{3,0}U_{\mathcal{I}}(t^{n-2}) + w_{4,0}U_{\mathcal{I}}(t^{n-3})\Big] + O(\tau^{4-\alpha}), \;\; n \geq 3 \end{aligned} \tag{3.9}$$

in which

$$\begin{aligned} \mu &= \frac{\tau^{-\alpha}}{\Gamma(2-\alpha)}, \\ a_j &= (j+1)^{1-\alpha} - j^{1-\alpha}, \\ b_j &= \frac{(j+1)^{2-\alpha} - j^{2-\alpha}}{2-\alpha} - \frac{(j+1)^{1-\alpha} + j^{1-\alpha}}{2}, \\ w_{1,j} =& \frac{1}{6}\Big[2(j+1)^{1-\alpha} - 11j^{1-\alpha}\Big] + \frac{1}{2-\alpha}\Big[(j+1)^{2-\alpha} - 2j^{2-\alpha}\Big] \\ &+ \frac{1}{(2-\alpha)(3-\alpha)}\Big[(j+1)^{3-\alpha} - j^{3-\alpha}\Big], \\ w_{2,j} =& \frac{1}{2}\Big[(j+1)^{1-\alpha} + 6j^{1-\alpha}\Big] - \frac{1}{2-\alpha}\Big[2(j+1)^{2-\alpha} - 5j^{2-\alpha}\Big] \\ &- \frac{3}{(2-\alpha)(3-\alpha)}\Big[(j+1)^{3-\alpha} - j^{3-\alpha}\Big], \\ w_{3,j} =& -\frac{1}{2}\Big[2(j+1)^{1-\alpha} + 3j^{1-\alpha}\Big] + \frac{1}{2-\alpha}\Big[(j+1)^{2-\alpha} - 4j^{2-\alpha}\Big] \\ &+ \frac{3}{(2-\alpha)(3-\alpha)}\Big[(j+1)^{3-\alpha} - j^{3-\alpha}\Big], \end{aligned}$$

and

$$
\begin{aligned}
w_{4,j} =& \frac{1}{6}\Big[(j+1)^{1-\alpha}+2j^{1-\alpha}\Big]+\frac{1}{2-\alpha}j^{2-\alpha}-\frac{1}{(2-\alpha)(3-\alpha)}\times\\
&\Big[(j+1)^{3-\alpha}-j^{3-\alpha}\Big].
\end{aligned}
$$

By replacing (3.7)-(3.9) one by one in (3.6), the following finite differences equations are achieved, respectively

$$
{}_1A_{\mathcal{I}}U_{\mathcal{I}}(t^1)-A_{\mathcal{B}}U_{\mathcal{B}}(t^1)-S\Big(t^1,U_{\mathcal{I}}(t^1)\Big)-\mu U_{\mathcal{I}}(t^0)=0, \tag{3.10}
$$

$$
{}_2A_{\mathcal{I}}U_{\mathcal{I}}(t^2)-A_{\mathcal{B}}U_{\mathcal{B}}(t^2)-S\Big(t^2,U_{\mathcal{I}}(t^2)\Big)+\mu\Big[(b_0-a_1)U_{\mathcal{I}}(t^0)+(a_1-a_0-2b_0)U_{\mathcal{I}}(t^1)\Big]=0, \tag{3.11}
$$

and

$$
\begin{aligned}
&{}_3A_{\mathcal{I}}U_{\mathcal{I}}(t^n)-A_{\mathcal{B}}U_{\mathcal{B}}(t^n)-S\Big(t^n,U_{\mathcal{I}}(t^n)\Big)+\mu\Bigg[(b_{n-2}-a_{n-1})U_{\mathcal{I}}(t^0)+(a_{n-1}-a_{n-2}-2b_{n-2})U_{\mathcal{I}}(t^1)\\
&+(a_{n-2}+b_{n-2})U_{\mathcal{I}}(t^2)+\sum_{k=3}^{n-1}\Big(w_{1,n-k}U_{\mathcal{I}}(t^k)+w_{2,n-k}U_{\mathcal{I}}(t^{k-1})+w_{3,n-k}U_{\mathcal{I}}(t^{k-2})+w_{4,n-k}U_{\mathcal{I}}(t^{k-3})\Big)\\
&+w_{2,0}U_{\mathcal{I}}(t^{n-1})+w_{3,0}U_{\mathcal{I}}(t^{n-2})+w_{4,0}U_{\mathcal{I}}(t^{n-3})\Bigg]=0,
\end{aligned} \tag{3.12}
$$

in which

$$
\begin{aligned}
{}_1A_{\mathcal{I}}&=\big(\mu I-F'\big)-\dot{A}_{\mathcal{I}}+\kappa_\alpha^\gamma c_\gamma(\bar{\bar{A}}_{\mathcal{I}}+\bar{A}_{\mathcal{I}}),\\
{}_2A_{\mathcal{I}}&=\Big(\mu(a_0+b_0)I-F'\Big)-\dot{A}_{\mathcal{I}}+\kappa_\alpha^\gamma c_\gamma(\bar{\bar{A}}_{\mathcal{I}}+\bar{A}_{\mathcal{I}}),\\
{}_3A_{\mathcal{I}}&=\big(\mu w_{1,0}I-F'\big)-\dot{A}_{\mathcal{I}}+\kappa_\alpha^\gamma c_\gamma(\bar{\bar{A}}_{\mathcal{I}}+\bar{A}_{\mathcal{I}}),
\end{aligned}
$$

and

$$
A_{\mathcal{B}}=\dot{A}_{\mathcal{B}}-\kappa_\alpha^\gamma c_\gamma(\bar{\bar{A}}_{\mathcal{B}}+\bar{A}_{\mathcal{B}}).
$$

Clearly, Eqs. (43), (44) and (45) are linear (nonlinear) systems, when in Eq. (1), $s(x,t,u)$ is a linear (nonlinear) function of u. For linear case, the obtained linear systems have identical coefficients matrices for $n\geq 2$. So, the decomposition methods seem to be suitable for solving them. But, we found that the "LU" decomposition methods were inefficient for these. So, we solved them using "QR" decomposition method as a stable method. In nonlinear case, the resulted nonlinear systems are solved by utilizing the Newton iterative method as follows:

In nth time step for $n=1,2,\ldots,N$, the unknown vector $U_{\mathcal{I}}(t^n)$ is given by

$$
U_{\mathcal{I}}^{k+1}(t^n)=U_{\mathcal{I}}^{k}(t^n)-\Big(J_n\big(U_{\mathcal{I}}^{k}(t^n)\big)\Big)^{-1}\tilde{F}_n\big(U_{\mathcal{I}}^{k}(t^n)\big),\ k=0,1,2,..., \tag{3.13}
$$

with a suitable initial guess $U_{\mathcal{I}}^{0}(t^n)$. In Eq. (3.13), $\tilde{F}_n$ is a function of $U_{\mathcal{I}}(t^n)$ which is denoted by the left sides of the Eqs. (3.10), (3.11) and (3.12), and $J_n(X)$ is the Jacobian matrix of

$\tilde{F}_n(X)$. Also, $U_{\mathcal{B}}(t^n)$ is calculated by the boundary conditions (1.4) and (1.5). Then, the vector $U(t^n) = \Big[u_1(t^n), u_2(t^n), \dots, u_M(t^n)\Big]^T$ is formed by adding $U_{\mathcal{B}}(t^n) = \Big[u_1(t^n), u_M(t^n)\Big]^T$ to $U_{\mathcal{I}}(t^n) = \Big[u_2(t^n), \dots, u_{M-1}(t^n)\Big]^T$. Finally, by substituting $U(t^n)$ in Eq. (3.3), the unknown function $U(x, t^n)$ is achieved.

4. **Numerical illustrations**

Here, we illustrate three numerical examples by the mentioned method. We put a positive integer ”$\mathcal{N}$” instead of ”∞” in equations (2.9)- (2.12). In all examples, $\mathcal{N} = 15$ is considered. Also, we utilize the extrema points of $\hat{T}_{M-1}(x)$ as discretization points that are given as

$$x_j = -\frac{1}{2}\cos\frac{\pi(j-1)}{M-1} + \frac{1}{2},\ j = 1, 2, \dots, M.$$

The numerical experiments is implemented in **Maple 16** and **SageMath 8.8** software on a PC with an Intel(R) Core(TM) i5-4210U CPU, a 64-bit Windows 8.1 operating system, and 6 GB internal memory. The stop condition of the Newton method is considered as

$$\frac{\|X_{k+1} - X_k\|_\infty}{\|X_{k+1}\|_\infty} < 10^{-5}.$$

The errors are computed at $t = 1$ by the formulas

$$E_\infty = \Big\|u_{exact}(x,1) - u_{approx}(x,1)\Big\|_\infty = \max_{1\le i\le \mathcal{M}}\Big|u_{exact}(x_i,1) - u_{approx}(x_i,1)\Big|,$$

$$E_2 = \left[\sum_{i=1}^{\mathcal{M}}\Big(u_{exact}(x_i,1) - u_{approx}(x_i,1)\Big)^2\right]^{\frac{1}{2}},$$

$$RMSE = \left[\frac{1}{\mathcal{M}}\sum_{i=1}^{\mathcal{M}}\Big(u_{exact}(x_i,1) - u_{approx}(x_i,1)\Big)^2\right]^{\frac{1}{2}}.$$

We calculate the errors with $\mathcal{M} = 101$ uniform points.
Moreover, the following formula is used to compute the experimental convergence order ($C-Order$) of the new method

$$C - Order = log_2\left(\frac{\ell(\Delta x, 2\tau)}{\ell(\Delta x, \tau)}\right),$$

where ℓ can be E_∞, E_2 and $RMSE$ errors.

4.1. **Illustration 1.**

Consider the nonlinear TSFFPE:

$$\frac{\partial^\alpha u(x,t)}{\partial t^\alpha} = \left[-\frac{\partial}{\partial x} + 2\frac{\partial^\gamma}{\partial|x|^\gamma}\right]u(x,t) + s(x,t,u),\ 0 \le x \le 1,$$

where

$$s(x,t,u) = u^3(x,t) + \frac{\Gamma(6)}{\Gamma(6-\alpha)}t^{5-\alpha}x^2(1-x)^2 + 2(1+t^5)(x - 3x^2 + 2x^3)$$

$$+ \frac{2(1+t^5)}{\cos(\frac{\pi\gamma}{2})}\left[\frac{x^{2-\gamma}}{\Gamma(3-\gamma)} - \frac{6x^{3-\gamma}}{\Gamma(4-\gamma)} + \frac{12x^{4-\gamma}}{\Gamma(5-\gamma)} + \frac{(1-x)^{2-\gamma}}{\Gamma(3-\gamma)} - \frac{6\ (1-x)^{3-\gamma}}{\Gamma(4-\gamma)} + \frac{12\ (1-x)^{4-\gamma}}{\Gamma(5-\gamma)}\right],$$

with the initial and boundary conditions,

$$u(x,0) = x^2(1-x)^2,\ 0 \le x \le 1,$$
$$u(0,t) = 0,\ t \ge 0,$$
$$u(1,t) = 0,\ t \ge 0.$$

The exact solution is $u(x,t) = (1+t^5)x^2(1-x)^2$.
In Table 1, the errors obtained by our technique for $\tau = 0.0125$, $\gamma = 1.4$, $M = 13$, $\varepsilon = 0.4$, $\sigma = 3$, $t = 1$ and various values of α are listed. In this case, $\kappa_\infty(\phi_X^{-T}) = 19.094$ and so our interpolation is well-condition. In Table 2, we show the E_∞, E_2 and $RMSE$ errors of our numerical findings for various values of γ. In this case, the condition number is 22.099. The E_∞ errors and $C-Order$s obtained by the mentioned method with some time steps are listed in Table 3. This table demonstrates the $C-Order$ for our method is approximately $4-\alpha$. Here the condition number of matrix ϕ_X^{-T} is 15.199. Table 4 depicts the errors and $\kappa_\infty(\phi_{\mathrm{X}}^{-\mathrm{T}})$ for different values of M. As we expected the greater values of M lead to the smaller errors.

Table 1. The errors for different values of α in Example 1.

α	E_∞	E_2	$RMSE$
0.1	4.75198×10^{-10}	3.37112×10^{-9}	3.354389×10^{-10}
0.3	3.33917×10^{-9}	2.36611×10^{-8}	2.35437×10^{-9}
0.5	1.41023×10^{-8}	9.99002×10^{-8}	9.94044×10^{-9}
0.7	5.26611×10^{-8}	3.73007×10^{-7}	3.71156×10^{-8}
0.9	1.85930×10^{-7}	1.31697×10^{-6}	1.31043×10^{-7}

Table 2. The comparison of E_∞, E_2 and $RMSE$ errors using presented method with $\tau = 0.01$, $\alpha = 0.45$, $M = 15$, $\varepsilon = 0.4$, $\sigma = 3$ at $t = 1$ in Example 1.

γ	E_∞	E_2	$RMSE$
1.1	6.13963×10^{-9}	4.38157×10^{-8}	4.35983×10^{-9}
1.3	5.07513×10^{-9}	3.61010×10^{-8}	3.59218×10^{-9}
1.5	4.05968×10^{-9}	2.87585×10^{-8}	2.86158×10^{-9}
1.7	3.15647×10^{-9}	2.22254×10^{-8}	2.21151×10^{-9}
1.9	2.39382×10^{-9}	1.67260×10^{-8}	1.66430×10^{-9}

Table 3. The comparison of $C-Order$s and E_∞ errors using presented method with $\alpha = 0.35$, $\gamma = 1.6$, $M = 11$, $\varepsilon = 0.1$, $\sigma = 3$ at $t = 1$ in Example 1.

τ	E_∞	$C-order$
0.04	2.45902×10^{-7}	$-$
0.02	2.07575×10^{-8}	3.57
0.01	1.71719×10^{-9}	3.60
0.005	1.40380×10^{-10}	3.61

TABLE 4. The comparison of E_∞, E_2 and $RMSE$ errors and condition number of matrix ϕ_X^{-T} using presented method with $\alpha = 0.5$, $\gamma = 1.7$, $\varepsilon = 0.9$, $\sigma = 3$, $\tau = 0.02$ at $t = 1$ in Example 1.

M	E_∞	E_2	$RMSE$	$\kappa_\infty(\phi_X^{-T})$
6	2.77098×10^{-3}	1.62115×10^{-2}	1.61310×10^{-3}	11.102
8	8.61846×10^{-5}	4.55483×10^{-4}	4.53223×10^{-5}	14.993
10	1.57389×10^{-6}	8.55279×10^{-6}	8.51034×10^{-7}	18.884
12	6.40795×10^{-8}	3.60980×10^{-7}	3.59188×10^{-8}	22.775

4.2. **Illustration 2.** Consider

$$ {}^{c}D_t^{\alpha} u(x,t) = \left[\frac{\partial}{\partial x}\Big(\frac{1}{x}\Big) + \frac{\partial^\gamma}{\partial |x|^\gamma}\right] u(x,t) + s(x,t,u),\ \ 0 < x \leq 1, $$

where

$$ \begin{aligned} s(x,t,u) = {} & r t^4 x^2 \Big(1 - \frac{t^4 x^2}{k}\Big) + \frac{24}{\Gamma(5-\alpha)} x^2 t^{4-\alpha} - t^4 \\ & + c_\gamma t^4 \Big[\frac{2\ x^{2-\gamma}}{\Gamma(3-\gamma)} + \frac{(1-x)^{-\gamma}}{\Gamma(1-\gamma)} - \frac{2\ (1-x)^{1-\gamma}}{\Gamma(2-\gamma)} + \frac{2\ (1-x)^{2-\gamma}}{\Gamma(3-\gamma)}\Big], \end{aligned} $$

and

$$ \begin{aligned} & u(x,0) = 0,\ \ 0 \leq x \leq 1, \\ & u(0,t) = 0,\ \ t \geq 0, \\ & u(1,t) = t^4,\ \ t \geq 0. \end{aligned} $$

The exact solution of this problem is $u(x,t) = t^4 x^2$.
We solved the problem for $r = 0.2$ and $k = 1$. In Table 5, the E_∞ errors and $C - Order$s for $\alpha = 0.3$, $\gamma = 1.8$, $M = 12$, $\varepsilon = 0.1$, $\sigma = 3$, and $t = 1$ are listed. In this case, $\kappa_\infty(\phi_X^{-T}) = 16.619$, and as the table demonstrates, $C - Order$s are approximately $4 - \alpha$. In Table 6, we show the errors and $\kappa_\infty(\phi_X^{-T})$ for different values of M.

TABLE 5. The comparison of E_∞ errors and $C - order$s for different values of τ using presented method in Example 2.

τ	E_∞	$C - Order$
0.1	9.22702×10^{-6}	–
0.05	7.36608×10^{-7}	3.65
0.025	5.83948×10^{-8}	3.66
0.0125	4.60154×10^{-9}	3.67
0.0625	3.60865×10^{-10}	3.67
0.003125	2.83421×10^{-11}	3.67

TABLE 6. The comparison of the errors and $\kappa_\infty(\phi_X^{-T})$ for various values of M using presented method with $\alpha = 0.4$, $\gamma = 1.5$, $\varepsilon = 0.9$, $\sigma = 3$, $\tau = 0.0125$ at $t = 1$ in Example 2.

M	E_∞	E_2	$RMSE$	$\kappa_\infty(\phi_X^{-T})$
6	1.35139×10^{-3}	7.73837×10^{-3}	7.69996×10^{-4}	11.1024
9	2.89211×10^{-6}	1.53131×10^{-5}	1.52371×10^{-6}	16.9385
12	1.68157×10^{-8}	9.90741×10^{-8}	9.85824×10^{-9}	22.7746
15	1.38782×10^{-8}	9.80144×10^{-8}	9.75280×10^{-9}	28.6108

4.3. **Illustration 3.** Consider [26]

$$\frac{\partial^{0.8}u(x,t)}{\partial t^{0.8}} = \Big[-\frac{\partial}{\partial x} + 25\frac{\partial^{1.9}}{\partial |x|^{1.9}} \Big] u(x,t) + f(x,t), \ 0 \le x \le 1,$$

with the source term

$$f(x,t) = 1.8\Gamma(1.8)tx^2(1-x)^2 + \frac{25\Big(25+t^{1.8}\Big)}{2\cos\left(\frac{1.9\pi}{2}\right)}\Big[g(x) + g(1-x)\Big]$$
$$+ 2\Big(25+t^{1.8}\Big)x(1-x)(1-2x),$$
$$g(x) = \frac{4!}{\Gamma(3.1)}x^{2.1} - 2\frac{3!}{\Gamma(2.1)}x^{1.1} + \frac{2}{\Gamma(1.1)}x^{0.1},$$

and the conditions,

$$u(x,0) = 25x^2(1-x)^2, \ 0 \le x \le 1,$$
$$u(0,t) = u(1,t) = 0, \ 0 \le t \le 1.$$

The exact solution is $u(x,t) = \Big(25+t^{1.8}\Big)x^2(1-x)^2$.

In [26], this problem has been investigated by a finite difference method with spatial step $h = \frac{1}{M}$ and temporal step $\tau = \frac{1}{N}$, and some numerical approximations have been obtained in the mesh points $(x_i, t_j) = (ih, j\tau)$, $i = 1, \cdots, M$, $j = 1, \cdots, N$. We solved the problem by the new method with $\varepsilon = 0.3$, $\sigma = 3$ and the same values of M and N. In Table 7, our results at $t = 1$ are compared with those presented in [26]. Moreover, the condition number of matrix ϕ_X^{-T} is depicted in Table 7.

TABLE 7. The comparison between E_∞ errors of presented method and the method in [26], and values of $\kappa_\infty(\phi_X^{-T})$ in Example 3.

	Our method		Method in [26]
$N = M$	E_∞	$\kappa_\infty(\phi_X^{-T})$	E_∞
10	4.0530×10^{-8}	14.202	4.8148×10^{-2}
20	3.2107×10^{-9}	28.832	1.0111×10^{-2}
40	4.3520×10^{-10}	58.093	2.0587×10^{-3}
80	8.4314×10^{-11}	116.61	7.3019×10^{-4}

5. Conclusion

We developed the stable Gaussian RBF method to solve TSFFPE. In this regard, we applied the Riesz fractional derivative of the eigenfunction Gaussian interpolants, and by a spatial discretization we converted the problem to a system of fractional ODEs. To solve this system, we presented a high order finite difference scheme. We included several numerical examples to confirm the validity, convergence and stability of the method. Numerical experiments show that the condition number of ϕ_X^{-T} is small, while the accuracy of solutions are acceptable. In fact, while using the capabilities of the RBFs method, we do not encounter the main drawback of this method, which is the same ill-conditioning. Our method gives a closed form approximate solution in each time step, and it can be extended for many types of time- and space-fractional PDEs.

References

[1] D. A. Benson, S. W. Wheatcraft and M. M. Meerschaert, Application of a fractional advection-dispersion equation, Water resources research, 36 (6), 1403-1412, (2000).

[2] M.D. Buhman, Spectral convergence of multiquadric interpolation, Proc. Edinburg Math. Soc., 36, 319-333, (1993).

[3] J.X. Cao, C.P. Li and Y.Q. Chen, High-order approximation to Caputo derivatives and Caputo-type advection-diffusion equations (II). Fractional Calculus and Applied Analysis, 18(3) 735-761, (2015).

[4] R.E. Carlson and T.A. Foley, The parameter r^2 in multiquadric interpolation, Comput. Math. Appl., 21, 29-42, (1991).

[5] R. Cavoretto, G.E. Fasshauer and M. McCourt, An introduction to the Hilbert-Schmidt SVD using iterated Brownian bridge kernels, Numer. Algorithms, 68(2), 393-422, (2015).

[6] K. Deng and W. Deng, Finite difference/predictor–corrector approximations for the space and time fractional Fokker–Planck equation, Applied Mathematics Letters, 25, 1815-1821, (2012).

[7] W.H. Deng, Finite element method for the space and time fractional Fokker-Planck equation, Siam J. Numer. Anal., 47 (1), 204-226, (2008).

[8] E.H. Dohaa, A.H. Bhrawy and S.S. Ezz-Eldien, A Chebyshev spectral method based on operational matrix for initial and boundary value problems of fractional order, Computers and Mathematics with Applications, 62, 2364-2373, (2011).

[9] R. Sh. Dubey, B.S.T. Alkahtani and A. Atangana, Analytical Solution of Space-Time Fractional Fokker-Planck Equation by Homotopy Perturbation Sumudu Transform Method, Mathematical Problems in Engineering, (2015), doi.org/10.1155/2015/780929, 7 pages.

[10] G.E. Fasshauer and M.J. MacCourt, Stable evaluation of Gaussian radial basis function interpolants, SIAM J. Sci. Comput., 34(2), A737-A762, (2012).

[11] E. K. Lenzy, L. C. Malacarne, R. S. Mendes and I. T. Pedron, Anomalous diffusion, nonlinear fractional Fokker-Planck equation and solutions, Physica A, 319, 245-252, (2003).

[12] W. K. Liu and W. M. Han, Reproducing kernel element method. Part I: Theoritical information, Comput. Methods Appl. Mech. Engrg., 193, 933-951, (2004).

[13] J.M. Melenk and I. Babuska, The partition of unity method: basic theory and applications, Comput. Methods Appl. Mech. Engrg., 139, 289-314, (1996).

[14] R. Metzler and J. Klafter, Fractional Fokker-Planck equation: dispersive transport in an external forcefield, journal of molecular liquids, 86 (1), 219-228, (2000).

[15] L. Pinto and E. Sousa, Numerical solution of a time-space fractional Fokker Planck equation with variable force field and diffusion, Commun Nonlinear Sci Numer Simulat, 50, 211-228, (2017).

[16] C. Piret and E. Hanert, A radial basis functions method for fractional diffusion equations. J. Comput. Phys., 238, 71–81, (2013).

[17] I. Podlubny, Fractional differential equations, Academic Press, San Diego, (1999).

[18] J. Rashidinia, G.E. Fasshauer and M. Khasi, A stable method for the evaluation of Gaussian radial basis function solutions of interpolation and collocation problems, Comput. Math. Appl., 72 (1), 178-193, (2016).

[19] J. Rashidinia, M. Khasi and G.E. Fasshauer, A stable Gaussian radial basis function method for solving nonlinear unsteady convection–diffusion–reaction equations, Computers and Mathematics with Applications 75, 1831-1850, (2018).

[20] S.S. Ray and A.K. Gupta, A two-dimensional Haar wavelet approach for the numerical simulations of time and space fractional Fokker-Planck equations in modelling of anomalous diffusion systems, Mathematical Chemistry, 52 (8), 2277-2293, (2014).

[21] H. Risken, The Fokker-Planck Equation: Method of Solution and Applications, Springer, Berlin, (1989).

[22] A. Saravanan and N. Magesh, An efficient computational technique for solving the Fokker-Planck equation with space and time fractional derivatives, Journal of King Saud University-Science, 28, 160-166, (2016).

[23] B. Sepehrian and Z. Shamohammadi, A method of lines for solving the nonlinear time and space-fractional Schrödinger equation via stable Gaussian radial basis function interpolation, Accepted.

[24] J. Xie, Zh. Yao, H. Gui, F. Zhao, and D. Li, A two-dimensional Chebyshev wavelets approach for solving the Fokker-Planck equations of time and space fractional derivatives type with variable coefficients, Appl. Math. Comput., 332, 197-208, (2018).

[25] Y. Yang, Y. Huang and Y. Zhou, Numerical solutions for solving time fractional Fokker-Planck equations based on spectral collocation methods, Computational and Applied Mathematics, 339, 389-404, (2017).

[26] Q. Yang, F. Liu, and I. Turner, Stability and convergence of an effective numerical method for the time-space fractional Fokker-Planck equation with a nonlinear source term, International Journal of Differential Equations, (2010), doi.org/10.1155/2010/464321, 22 pages.

[27] G. M. Zaslavsky, Chaos, fractional kinetics, and anomalous transport, Physics Reports, 371 (6), 461-580, (2002).

[28] P. Zhuang, F. Liu, V. Anh and I. Turner, Numerical treatment for the fractional Fokker-Planck equation, ANZIAM, 48, 759-774, (2007).

[29] M. L. Zuparic, A. C. Kalloniatis, Analytic solution to space-fractional Fokker-Planck equations for tempered-stable Lévy distributions with spatially linear, time-dependent drift, J. Phys. A: Math. Theor., Volume 51, Issue 3, article id. 035101 (2018), doi: 10.1088/1751-8121/aa9ad8.

(Z. Shamohammadi) DEPARTMENT OF MATHEMATICS, FACULTY OF SCIENCE, ARAK UNIVERSITY, ARAK 38156-8-8349, IRAN

Email address: `bsp408@yahoo.com`

(B. Sepehrian) DEPARTMENT OF MATHEMATICS, FACULTY OF SCIENCE, ARAK UNIVERSITY, ARAK 38156-8-8349, IRAN

Email address: `b-sepehrian@araku.ac.ir`