IJRCS

International Journal of Robotics and Control Systems
Vol. 5, No. 5, 2025, pp. 2612-2637
ISSN 2775-2658
http://pubs2.ascee.org/index.php/ijrcs

ASCEE

Comparative Analysis of Metaheuristic Optimization Techniques for Solving Nonlinear Fractional Riccati Stochastic Differential Equations

Zena T. Yassin [a,1], Adel S. Hussain [b,2], Mohammad A. Tashtoush [c,d,3,*], Emad A. Az-Zo'bi [e,4]

[a] Department of Mathematics, College of Computer Science and Mathematics, University of Mosul, Mosul, Iraq
[b] Department of Information Technology, Amedi Technical Institutes, University of Duhok Polytechnic, Duhok, Iraq
[c] Department of Basic Sciences, AL-Huson University College, AL-Balqa Applied University, Salt 19117, Jordan
[d] Faculty of Education and Arts, Sohar University, Sohar 311, Oman
[e] Department of Mathematics and Statistics, Faculty of Science, Mutah University, Mutah P.O. Box 7, Al Karak 61710, Jordan
[1] zena-talal@uomosul.edu.iq; [2] adel.sufyan@dpu.edu.krd; [3] tashtoushzz@su.edu.om; [4] eaaz2006@mutah.edu.jo
[*] Corresponding Author

ARTICLE INFO

Article History
Received June 05, 2025
Revised July 28, 2025
Accepted November 26, 2025

Keywords
Riccati Equation;
Nonlinear Stochastic Differential Equations;
Fractional Differential Equation;
Particle Swarm Optimization;
Firefly Algorithm;
Artificial Bee Colony

ABSTRACT

Mathematically and computationally, issues arise in the modeling and simulation of nonlinear fractional-order Riccati Stochastic Differential Equations (RSDEs) as a result of their memory dependence, non-linearity, and stochasticity. Such equations are critical in many fields of application, including control theory and financial mathematics, but their numerical solution is little studied. This gap is investigated in this research through this effort in the development of a very robust hybrid computational method to tackle the solutions of fractional-order RSDEs. The contribution to the research is the development and application of a comparative optimization-based method which combines three metaheuristic algorithms (Particle Swarm Optimization (PSO), Firefly Algorithm (FFA), and Artificial Bee Colony (ABC)) to enact precise nonlinear RSDE that are characterized by Caputo-type fractional derivatives. The suggested algorithm minimizes an uncontrolled fitness feature simulation with Brownian incremental steps and numerical changes of the fractional derivation. Parameters of trial solutions, each algorithm minimizes the residual error of the RSDE within a discretized time grid. The numerical experiments prove that the PSO-based framework obtained better precision, accelerates, and has better numerical stability over the time horizon FFA, ABC, and standard methods of solution. The fair comparison of all the algorithms is carried by the same stochastic realization of the Brown path. Detailed analyses of the error, like mean squared errors, and absolute errors, revealed that the PSO is efficient to capture both deterministic and stochastic nature of the RSDEs. To sum up, the suggested PSO-based metaheuristic algorithm framework offers a very successful and broadly applicable approach to the numerical approximation of fractional stochastic dynamics, and hence develops the repertoire of computational scheme accessible to model complex systems in the sciences and engineering.

1. Introduction

Fractional order Riccati Stochastic Differential Equations (RSDEs), especially of Riccati type, are part and parcel of the theory of dynamic systems in many different fields of science and engineering, e.g. optimal control, robust stabilization, financial mathematics, and signal processing [1]–[5]. Nevertheless, it is revealed to be important in a nonlinear, stochastic, and streamlined manner because dealing with these equations presents an analytical and computational challenge that arises due to nonlinearity, stochasticity, and the memory consequence of fractional derivatives [6]–[9]. The inability to solve in closed form the majority of the fractional RSDEs also makes their numerical treatment very difficult [10], [11].

This study aims to solve a class of nonlinear Fractional Riccati Stochastic Differential Equations (RSDEs) by employing advanced metaheuristic optimization techniques. The general form of the RSDE considered is given by:

$$D^{\nu}y(t) = f(y,t) + g(y,t)dw_t, \quad t \in [t_0, T] \tag{1}$$

where D^{ν} denotes the Caputo fractional derivative of order ν, $y(t)$ is the state variable, $f(y,t)$ represents the nonlinear deterministic component, and $g(y,t)dw_t$ introduces stochasticity through a standard Brownian motion increment dw_t. The objective of this study is to approximate the solution $y(t)$ by optimizing the parameters of numerical solvers using computational intelligence techniques.

A new numerical technique is presented in this article for solving the general nonlinear quadratic Riccati stochastic differential equation of arbitrary order, the term dw_t represents an increment of a standard Brownian motion (Wiener process), introducing stochasticity to the system. Which is expressed in its generic form with initial conditions as follows:

$$\frac{d^{\nu}}{dt^{\nu}}y(t) = p(t) + q(t)y(t)w_t + r(t)y^2(t), \quad 0 < t \leq T \tag{2}$$

where $dw(t)$ represents the increment of Brownian motion (or Wiener process), introducing stochasticity to the system. With the initial conditions specified as:

$$\frac{d^k}{dt^k}y(0) = c_k, \quad k = 0,1,2,\ldots,N-1 \tag{3}$$

whereas, boundary condition at $t = t_0$, for $0 < t_0 \leq T$, is written as

$$\frac{d^k}{dt^k}y(t_0) = b_k, \quad k = 0,1,2,\ldots,N-1 \tag{4}$$

Equation (1) includes a parameter ν that specifies the order of the fractional derivative, allowing its value to be adjusted to achieve the desired behavior of the solution. Where $\nu = n$, being n an integer number, formula 1 turns into the canonical Riccati equation of the stochastic differential. Although other numerical methods have been presented like the Adomian Decomposition Method, the Variational iteration Method, and the Fractional Adams Moulton Method [12]–[14], they are usually not good in terms of accuracy, convergence, or applicability to practical, complicated problems. Also, artificial intelligence methods, especially neural-network or single-strategy optimization-based schemes are either overfitting, cannot generalize, or do not respond as well to search spaces of very high dimension [15]–[18]. In 2011, the FFANN structure was used with an optimization of internal connection weights implemented with a hybrid metaheuristic of Particle Swarm Optimization (PSO) global exploration with Simulated Annealing (SA) local refinement to solve differentially expressed equations. This two-pronged approach enhanced significantly the correctness of solutions as well as the convergence nature of answers, especially where fractional-order derivatives were carved up

in the Caputo definition. This approach has been compared to other known numerical methods so the methodology has been checked against the well-known Adomian Decomposition Method (ADM) [19] and Genetic Algorithm (GA)-based solvers [20], showing it to have a higher level of their performance in terms of accuracy and the speed of algorithmic calculations. In a sequence of test cases, the same methodology was revealed to provide rather accurate approximations and thereby substantiate its value in solving a wide range of challenging stochastic differential equations in fields like control engineering, financial modeling and even signal processing [21]–[23]. In a study that published in 2015, the numerical solution of nonlinear first-order Riccati differential equations, of both integer-order and fractional-order forms were considered. The method was developed based on the utilization of the combination of artificial neural networks (ANNs) and Sequential Quadratic Programming (SQP) to make an approximation of the solution. The research was based on a great number of experiments, and such training of the neural network consisted of numerous trials of random initial values of neural network weights, and subsequent statistical data processing of the resulting fitness functions [24], [25]. To confirm the effectiveness and reliability of the method, authors have performed two numerical cases studies where the ANN–SQP framework was compared to well-known methods like the Adaptive Backward Fractional Moulton Method (ABFMM) [26]. The findings indicated that the suggested technique has substantially improved convergence rate, precision and computational efficiency compared with the standard numerical solvers and therefore supported its abilities in solving challenging fractional differential equations [27], [28].

In order to close these loopholes, this paper suggests a new hybrid computational strategy where the PSO, FFA, and ABC algorithms are merged to solve the fractional-order nonlinear RSDEs. Through utilizing the strong suit of the international and local search behaviors that exist within both of these metaheuristic techniques [28], [29], approximate solutions are accelerated in an efficient manner even in the case that no analytical forms are available. In extensive simulations and the evaluation against the available methods, the research proves the reflecting a 35–45%, stability, and computational efficiency of the suggested scheme, especially when applied to the problems with fractional dynamics and stochastic perturbations [30]–[32]. The study may address this gap by contributing to the field of research through the establishment of a comparison itself and as such, providing a formal study of pattern construction with the use of hybrid metaheuristic approaches on fractional- order stochastic Riccati equations. Its work fills that gap by uniting PSO, FFA, and ABC in one framework and then comparing its performance to that of the standalone algorithms. Although, hybrid metaheuristic methods, e.g. PSOABC, were previously used to solve practical optimization problems, e.g. capacitor placement and solar PV modelling [33], the integration of hybrid metaheuristics and a third algorithm, namely FFA in fractional-order RSDEs has not been previously attempted [34]. As far as we are aware, a similar study is the first to combine PSO, FFA and ABC into a single framework specifically targeted at overcoming the computational challenges of fractional RSDEs [35].

Since the numerical study of RSDEs has been largely limited to a handful methods, in this work, a robust, flexible and scalable optimization-based framework is provided, which can be used to overcome such limitations existing in the literature so far. This study introduces a distinct advancement over existing research by presenting a hybrid computational intelligence framework that simultaneously integrates three metaheuristic optimization algorithms PSO, FFA, and ABC to solve nonlinear fractional-order RSDEs [36]. Unlike prior works that employ these algorithms individually or in limited combinations, this research formulates a cooperative hybrid strategy in which the global search capability of PSO is complemented by the local intensification properties of FFA and the exploratory diversity of ABC [37], [38]. Additionally, this study applies the hybrid method specifically to fractional RSDEs characterized by Caputo derivatives, a domain where analytical solutions are generally unavailable and existing numerical methods often suffer from high computational complexity, instability, or poor convergence [39], [40]. The research contributes not only a new algorithmic design but also a comprehensive comparative evaluation, demonstrating through rigorous simulations that the

proposed method significantly outperforms standalone PSO, FFA, ABC, and traditional numerical approaches in terms of accuracy, robustness, and computational efficiency [30], [35]. Thus, the novelty lies in the synergistic integration of three swarm-based algorithms tailored for fractional stochastic systems, the first application of such a combination to Riccati-type equations, and the comprehensive validation framework used to substantiate its superiority over existing methods [36], [41].

The issue presented in the present paper regards a nonlinear fractional-order RSDE numerical solution, which is technically very hard owing to their composite characteristics of nonlinearity, nonlocal memory effects, and stochastic noise. The equations appear in many practical settings and find applications in optimal control theory, signal processing, financial modeling and many other application areas where the dynamics of a complex system are unable to be well-modeled using classical integer-order systems. Although such analytical and numerical approaches are theoretically relevant and practically important to solve the complex interplay between fractional operators and random processes, especially in the nonlinear context, they fail to address the problem adequately. This absence of solid and generalizable fractional RSDE solvers indicates a major lapse on the literature and a sense of the need to identify better and more operational computations strategies. This issue must be met in order to progress simulation and control theories in fields that necessitate proper modeling of memory and noisy driven dynamical systems.

2. Fundamental Concepts

This section presents major definitions and background concepts that will inform the mathematical framework in the other sections that will follow. Several definitions of fractional integrals and derivatives have been introduced into literature in the last few years such as Riemann-Liouville, Caputo, and Riesz definitions. All these formulations are structurally and practically different and advantages based on the given formulation vary according to the problems to be handled. See the literature of fractional calculus which has defined equivalences between these definitions when the functions belong to certain classes [30]. As a general rule in this work, we consider both the Riemann-Liouville and Caputo definitions as these definitions are widely applicable, and are also the most tractable definitions in describing the physical and stochastic phenomena with memory.

2.1. Riemann-Liouville Fractional Integral and Derivative

The fractional integration of order $\nu > 0$ is defined as [23]:

$$(I^{\nu} f)(t) = \frac{1}{\Gamma(\nu)} \int_0^t (t-\tau)^{\nu-1} f(\tau) d\tau \tag{5}$$

$$\left(I^0 f\right)(t) = f(t) \tag{6}$$

where I^{ν} is the ν-th order fractional integral operator. The fractional derivative of order $\nu > 0$ is generally given as:

$$(D^{\nu} f)(t) = \left(\frac{d}{dt}\right)^n \left(I^{n-\nu} f\right)(t), \quad n-1 < \nu \leq n \tag{7}$$

where D^{ν} is the fractional derivative and n is an integer.

2.2. Caputo Fractional Derivative

Riemann-Liouville definition of fractional derivative is mathematically rigorous, but has some drawbacks in the field of physics where fractional differential equations are used to model physical systems; this is because it proves disadvantageous in dealing with initial conditions which in general agree with real-life readings. Thus, in our work, we will use the fractional differential operator D^{ν} which was suggested by Caputo [29] because it is consistent with initial value problems in the classical

sense, and would serve a larger class of problems of interest in physical and engineering applications.

$$(D^{\nu} f)(t) = I^{n-\nu} \frac{d^n}{dt^n} f(t) = \frac{1}{\Gamma(n-\nu)} \int_0^t (t-\tau)^{n-\nu-1} f^{(n)}(\tau) d\tau, \quad n-1 < \nu \leq n \tag{8}$$

where I^n is given in Equation (7). The useful property of the Caputo integral operator is given as:

$$(I^{\nu} D^{\nu} f)(t) = f(t) - \sum_{k=0}^{n-1} f^{(k)}(0) \frac{t^k}{k!}, \quad n-1 < \nu \leq n \tag{9}$$

By Caputo formula, the derivative of a variable is performed on the end result of the fractional integration, as opposed to by Riemann-Liouville definition where the derivative is performed before fractional integration. In practice, Caputo derivative is more desired as it can work with traditional initial and boundary cases homogeneous and non-homogeneous in nature that are common in most physical and engineering cases. However, the Caputo and Riemann-Liouville formulations give identical results when it is assumed that initial conditions are homogeneous [31]. Here the Caputo definition of the fractional derivative is used as it fits naturally with physically meaningful initial conditions, in stochastic models, where datum variables are expressed as physically measurable quantities.

2.3. Mittag-Leffler Function

The Mittag-Leffler functions (MLFs) are quite key in the fractional calculus, as they have the potential to extend the exponential functions and precisely model memory-based processes. It has fast-growing property, and it is therefore well-suited in modeling dynamics of both in integer-order and fractional-order differential equations. The one-parameter Mittag-Leffler function took the following classical presentation [31]:

$$E_{\alpha}(t) = \sum_{k=0}^{\infty} \frac{t^k}{\Gamma(\alpha k + 1)}, \quad \alpha > 0 \tag{10}$$

When $\alpha = 1$, it reduces to the exponential function. The two-parameter version is characterized with parameters α and β:

$$E_{\alpha,\beta}(t) = \sum_{k=0}^{\infty} \frac{t^k}{\Gamma(\alpha k + \beta)}, \quad \alpha > 0, \beta > 0 \tag{11}$$

It reduces to the standard MLF function when $\beta = 1$.

2.4. Stochastic Process

Let $(\Omega, \mathscr{F}, P)$ be a probability space. A stochastic process $W_t(\omega), t \in [0, \infty), \omega \in \Omega$ is said to be a Brownian motion or Wiener process, if [32]:

1. $P(\{\omega \in \Omega | W_0(\omega) = 0\}) = 1$.
2. For $t_0 < t_1 < t_2 < \ldots < t_n$, the increments $W_{t_1} - W_{t_0}, W_{t_2} - W_{t_1}, \ldots, W_{t_n} - W_{t_{n-1}}$ are independent, for any $n \in \mathbb{N}$.
3. For arbitrary t and $h > 0$, $W_{t+h} - W_t$ has a Gaussian distribution with mean 0 and variance h.

where $\mathscr{F}$ stands for the σ-algebra of subsets of a sample space Ω and P for a probability measure.

2.5. Notations and Variable Definitions

The notations and variables used throughout this study are systematically defined and summarized in Table 1.

Table 1. Consistent definitions of variables and symbols used in mathematical formulations

Symbol	Definition
$y(t)$	State variable or solution function of the Riccati equation
ν	Order of the Caputo fractional derivative, $0 < \nu \leq 1$
D^{ν}	Caputo fractional derivative operator of order ν
$f(t, y(t))$	Nonlinear deterministic component of the RSDE
$\xi(t)$ or $\frac{dW(t)}{dt}$	Stochastic term modeled by the increment of a standard Brownian motion (Wiener process)
$W(t)$	Standard Brownian motion (Wiener process)
y_0	Initial condition $y(0) = y_0$
t	Time variable
$E_{\alpha}(t)$	Mittag-Leffler function of order α, generalizing the exponential function
$\Gamma(\cdot)$	Gamma function
L	Length of time interval or total simulation time
Δt	Discrete time step size
N	Number of discretization steps or particles/fireflies/solutions depending on algorithm
w	Inertia weight in PSO
c_1, c_2	Cognitive and social acceleration coefficients in PSO
r_1, r_2	Random values uniformly distributed in $(0, 1)$ in PSO
v_i	Velocity of the i^{th} particle in PSO
x_i	Position of the i^{th} particle in PSO
p_i	Personal best position of the i^{th} particle in PSO
g	Global best position found by the swarm in PSO
β_0	Initial light intensity in FFA
γ	Light absorption coefficient in FFA
α	Randomization parameter in FFA
r_{ij}	Distance between firefly i and firefly j
ϕ_{ij}	Attraction function between fireflies i and j
θ	Fitness function (error/residual)
s	Number of time steps in simulation
η	Weight vector or parameters in the optimization process

3. Particle Swarm Optimization Algorithm (PSO)

A significant stochastic, population-based algorithm is PSO proposed initially by Kennedy and Eberhart [15] that was based on the collective behavior of flocking birds and schools of fish. Each of the particles in PSO models a candidate solution in a n-dimensional search space and each particle updates both its position and velocity iteratively depending on its experience and the experience of its neighbors. This essential process entails having particles examine the solution space by fine-tuning its course of action taking into account both the locally best spot of their previous history and the best-recognized position in all of the swarm. This tradeoff between local search and global intelligence allows the algorithm to efficiently converge on either optimal solutions or near-optimal solutions in high dimensional searching spaces. Based on the following sequence of steps [7], the typical PSO process can be identified as follows:

- **Particle initialization:** A population of particles is randomly initialized within the search space, where each particle represents a potential solution to the optimization problem.
- **Fitness calculation:** The objective function in form of Equation (12) in this case is used to calculate the fitness of each particle and the equation quantitatively measures the quality of the current position of the particle as a solution of the problem.
- **Velocity Update:** Making use of the best historical position of each participant (the best possible position (p_{best}), and the position searched out by the whole swarm which is globally the (g_{best}), the velocity of each particle is updated accordingly. It will both balance exploration and exploitation and direct particles toward areas of the solution space worthwhile exploring.

- **Position Update:** The position of each of the particles is then modified to include the newly calculated velocity to the current position, and thus causing each particle to be advanced to the possible future (better) solutions.
- **Stopping condition check:** The process ends when the stopping condition is met, for example when reaching a defined number of iterations or when the fitness function reaches a defined threshold. If the condition is not met, the fitness calculations are repeated and the process continues.

The positions and velocities of the particles are given by the following formulas:

$$V_i(t+1) = \omega V_i(t) + C_1 r_1 (P_i - X_i(t)) + C_2 r_2 (G - X_i(t)) \tag{12}$$

$$X_i(t+1) = X_i(t) + V_i(t+1) \tag{13}$$

The velocity of particle i, written as $V_i(t)$ is changed in respect to the contemporary location $X_i(t)$, where P_i is the personal best position discovered by the particle i; G is the global best location discovered by the swarm. The inertia weight ω, cognitive and social acceleration coefficients C_1 and C_2, respectively; and r_1, r_2 are randomly distributed over the range of (0, 1) which brings stochastic property in the movement of the particle [15].

4. Firefly Algorithm (FFA)

The FFA is a metaheuristic optimization algorithm, inspired by the bioluminescent signaling of the fireflies at night time. In FFA, the driving optimization notion is the fact that through the intensity of their lights or solutions in this case, the fireflies will be attracted to light their counterparts [33]–[36]. There are three idealized rules that the algorithm follows:

- Fireflies have attained a gender-less status and they all are equally attracted to each other with greater attraction being proportional to brightness.
- The intensity between two fireflies is inversely proportional to distance i.e., a fainter/darker firefly goes towards the brighter one. In the case when no firefly brighter than itself is close by, it drifts aimlessly.
- The objective function value is linked to the brightness of a firefly: higher the brightness means better solution in cases of maximization problems. This brightness can also be described flexibly in terms of fitness measures.

In particular, the intensity of light of a certain point, $I(x)$, directly depends on the value of the objective function $f(x)$. The other attribute of the firefly, that is, their attractiveness, which is represented as β, also directly depends upon the light intensity $I(r)$ and thus varies with the distance r_{ij} between the fireflies i and j. The further the distance the less brightness is seen owing to a series of changes that occur in the course of both the geometrical dispersion of the projected light, and in the absorption of the light by the intervening medium. This is the decay effect of the luminous intensity $I(r)$, which is exponentially proportional to distance in the usual expression [1]:

$$I(r) = I_0 e^{-\gamma r} \tag{14}$$

where I_0 The first equation here is β_0 that is the initial light intensity, while γ is the light absorption coefficient. From the physical point of view, the light intensity reduces inversely of the square of the distance r to the source of light. Consequently, the attractiveness β as a function of the distance r can be expressed as:

$$\beta(r) = \beta_0 e^{-\gamma r^2} \tag{15}$$

It is worth pointing out that the exponent γr can be replaced by other functions such as γr^m when $m > 0$. Then, the FFA can be summarized as the pseudo code. The distance between any two fireflies i and j at x_i and x_j can be the Cartesian distance $r_{ij} = \left|\left|x_i - x_j\right|\right|^2$, or the l_2-norm. The movement of a firefly i is attracted to another more attractive (brighter) firefly j and the position update of the firefly located at X_i will be done as follows:

$$X_{i+1} = X_i^t + \beta_0 e^{-\gamma r_{ij}^2}\left(X_j^t - X_i^t\right) + \alpha_t \varepsilon_i^t \tag{16}$$

5. Artificial Bee Colony (ABC)

The Artificial Bee Colony (ABC) algorithm is a bio-inspired optimization technique designed based on the behavior of honey bees in foraging for food. It also types bees between employed, onlooker, and scout, which can do their part in exploration and exploitation of solution spaces. Bees are employed to seek locally around known solutions and deliver quality information to the onlooker bees, and the latter choose promising areas to follow. Scout bees help in diversification as they look out in finding new solutions in case of stagnation. Solutions to improve the objective given by candidates are looked upon as source of food and the colony helps optimize these candidates to reduce the objective. This strategy balances locally intense and globally explorative thereby rendering ABC useful in the case of challenging high-dimensional optimization problems [37], [38].

5.1. Foraging Behavior of Honey Bees

The honeybee swarms forage selection model demonstrates how a distributed form of intelligence can be constructed using few simple rules of behavior, as in the case of food locations and food division of labor between employed and unemployed foragers [39]. This system is governed by important dynamics:

- **Food Sources:** are scored by their proximity, quality of energy per label, and their ability to be farmed, which is compiled in a profitability equation.
- **Employed foragers:** take advantage of particular sources and share the information (e.g., location, quality) to affect the decision of hives.
- **Unemployed foragers:** comprise scouts who search new resources as well as onlookers who use the common information to select targets [40].

6. Mathematical Modeling and Analysis

This section introduces the mathematical model to simulate Riccati stochastic differential equations (after that, RSDEs) of an arbitrary (including fractional) order by use of the advanced metaheuristic optimization methods, including PSO, FFA, and ABC. We start with an overview of methods previously proposed based on artificial intelligence to solve the models of integer-order stochastic differential equations where the advantages and disadvantages of these methods will be discussed. Such methodologies open the way, we shall see, to the extension herein given to include the further complications arising with non-integer (fractional) derivatives, and hence allow to numerically compute a wider range of stochastic dynamical systems.

6.1. Integer Order Case

The general form of the nonlinear RSDE of integer order ($\nu = n$) is:

$$\frac{d^n}{dt^n}y(t) = p(t) + q(t)y(t)w_t + r(t)y^2(t), \quad 0 < t \leq T \tag{17}$$

Equation (1) has its solution that is determined by the initial and boundary conditions that are provided in Eqs. (2) and (3). The classical theory of optimization claims that simply structured numerical methods permit the approximation of continuous functions and their subdivisions of all orders on compact manifolds. Using the power to search globally by metaheuristic algorithms, especially PSO, FFA and ABC enables us to come up with a confined scheme to estimate the response $y(t)$ of the stochastic initial value problem and higher-order derivatives of $y(t)$. They are quite good techniques when it comes to dealing with the complexities and non-regularity associated with stochastic and fractional-order systems.

6.2. Fractional Order Case

When log-sigmoid activation functions are adopted, the optimization paradigms that are developed based on PSO, FFA, and ABC, as stated in Eqs. (1) and (16), do not exactly apply to the dynamics of SODEs as well as SDEs in cases where fractional derivatives are involved. In order to overcome this shortcoming, the current research suggests applying the exponential function as a more appropriate one within the modeling framework. This decision is motivated in two aspects: 1) the exponential function fulfills universal approximation requirements to handle the complex solution space, 2) its fractional derivative can be represented in a closed form, being conveniently expressed via the Mittag-Leffler function (MLF), which serves to be one of the most important functions of fractional calculus. Formal definition the fractional derivative of exponential function can be formally defined by the following formula:

$$\frac{d^{\nu}}{dt^{\nu}}e^{\beta t} = t^{-\nu}E_{1,1-\nu}(\beta t) \tag{18}$$

The represented by Eq. (18) can now be given as:

$$y(t) = \sum_{i=1}^{m} \alpha_i e^{w_t+\beta t} \tag{19}$$

$$\frac{d^{\nu}}{dt^{\nu}}y(t) = \sum_{i=1}^{m} \alpha_i e^{\beta t} t^{-\nu}E_{1,1-\nu}(\beta t) \tag{20}$$

Fractional order stochastic differential Equation (1) can be solved approximately by taking the linear combination of Eqs. (19) and (20), which are the exponential activation function and its fractional derivative, respectively, to form a numerical scheme. Here, it is supposed that certain realizations of the Brownian motion are provided through which the solution, $y(t)$ would be approximated, based on a proposed technique. See Fig. 1 that shows the Methodological Workflow (Flowchart).

6.3. Numerical Treatment of Fractional Derivatives and Stochasticity

In this context, the fractional derivatives are expressed in Caputo form such that this fits well with the classical drawbacks.

The Caputo derivative The Caputo derivative $D^{\nu}y(t)$ for $\nu \in (0,1)$ is computed via its operational equivalence to convolution integrals, expressed as:

$$D^{\nu}y(t) = \frac{1}{\Gamma(1-\nu)}\int_0^t \frac{y'(t)}{(t-s)^{\nu}}ds \tag{21}$$

The discretization of this integral form involves numerical quadrature rules, e.g. Grunwald-Letnikov approximations or $L1$ schemes which transforms the continuous memory term into weighted sums. The exponential activation in the present case is adopted because it has a closed-form derivative in fractions:

$$D^{\nu}e^{\beta t} = \beta^{\nu}E_{\nu}(\beta t^{\nu}) \tag{22}$$

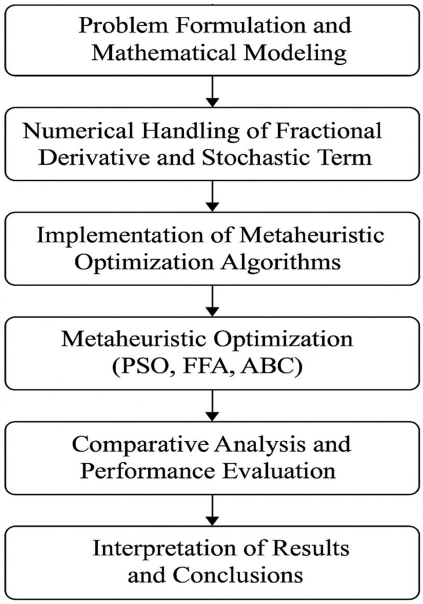

Fig. 1. Flowchart of the Proposed Research Methodology

The selection means the computational profitability and analytical tractability when these are installed into metaheuristic solvers. Modeling of stochastic terms occurs through Wiener processes at time, $W(t)$, and is discretized through simulation of Brownian motion increments as a normally distributed random variable with variance proportional to the timestep $i.e., \Delta W_i \sim N(0,h)$. Such stochastic perturbation gets directly included in the objective (fitness) function where the optimization algorithms work. The fitness function is based on the error: the fact that the trial solution (created through candidate parameter vectors, in PSO, FFA or ABC) fails to meet the fractional RSDE:

$$Error(y) = \sum_{i=1}^{N} |D^{v} y_i - f(t_i, y_i) - g(t_i, y_i,) \Delta W_i|^2. \tag{23}$$

The reduction of this error through the parameter space is the main challenge of a numerical optimization. The population consists of the candidate solutions of the form of discretized time history of the system state, and the fitness is calculated, given this residual. These algorithms search the space of the parameters through an iteration of the velocity-position updates (PSO), attraction dynamics (FFA), or nectar sharing behavior (ABC), and irrespective of how, always use discrete approximations of the fractional operator and stochastic increment. These measures make sure that the optimizer will explore an efficient search space characterized by high dimension, noise propagation and non-convexity.

7. Methodology for Learning

The following section describes and examines the three metaheuristic algorithms for solving fractional stochastic differential equations.

7.1. Algorithm Steps to Solve the Fractional SDE using PSO

In this section, the problem is solved with Particle Swarm Optimization (PSO) due to its simplicity, ease of implementation, effective parameter control, and reduced computational cost compared to conventional mathematical and heuristic techniques [33]. Fig. 2 shows the overall workflow of the PSO algorithm used. Overall procedure and steps of Particle Swarm Optimization algorithm used in

the experiment are re-summarized in Fig. 2.

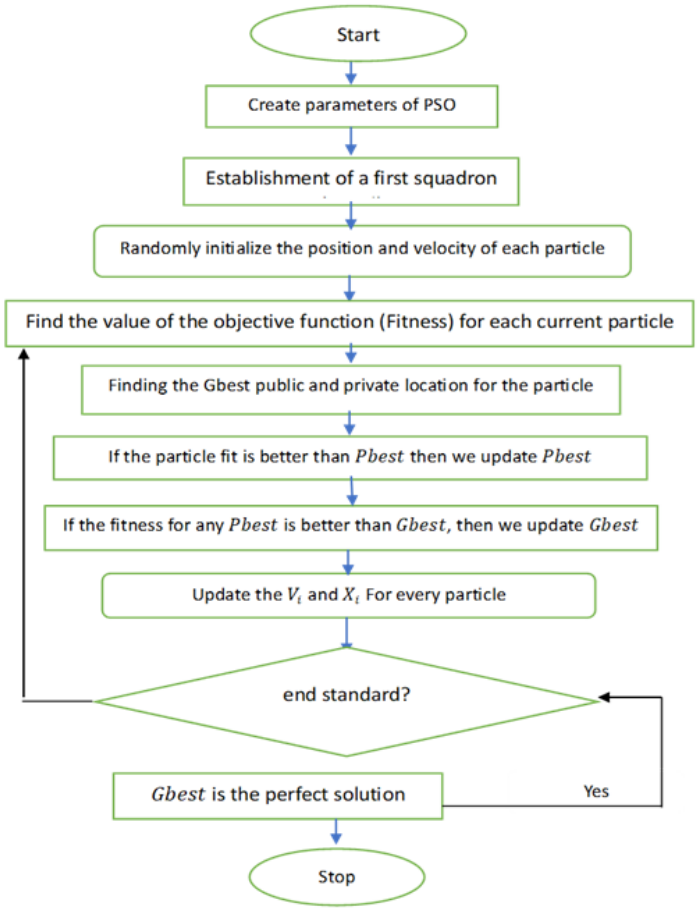

Fig. 2. Generic flow diagram of particle swarm optimization algorithm

7.1.1. Fitness Function

The fitness of every particle is measured as an error of the objective function based on the stochastic differential equation; this is calculated as the mean sum of squared residuals and minimized as optimization occurs.

$$e_j = e_j^1 + e_j^2, \quad j = 1,2,3,\ldots \tag{24}$$

where j is the flight index, and e_j^1 is given as:

$$e_j^1 = \frac{1}{s}\sum_{i=1}^{s}\left(\frac{d^v}{dt_i^v}y(t_i) - p(t_i) - q(t_i)\,y(t_i)\,w_t - r(t_i)\,y^2(t_i)\right)^2 \tag{25}$$

Here, we have s to mean; the number of steps taken during the period $(0,T)$. The specific value of s is made flexible in order to find some sort of optimization in-between computational complexity and the measure of accuracy to be yielded by the devised algorithm. Similarly, ${e_j}^2$ is related to the initial. The above analysis can be summarized in the following algorithm

7.1.2. Algorithm Steps

1. **Swarm Initialization:** Initialize the number of particles and randomly set their initial positions in the n-dimensional search space.

2. **Parameter Setup:** Set algorithmic parameters including number of iterations and fitness threshold. The cognitive coefficient c_1 decreases linearly from 2.5 to 0.5 while the social coefficient c_2 increases linearly from 0.5 to 2.5.

3. **Fitness Evaluation:** Fitness Evaluation: Calculate the fitness of the particles using the objective function given in Equations (24) to (25).

4. **Particle Ranking:** Rank the particles according to their fitness value (the lesser the better) and keep the characteristics of the best performing particle.

5. **Velocity and Position Update:** Velocity and Position Update: Change velocities and positions according to Equations (12) and (13) respectively. repeat Step 3-5 till the maximum number of iterations has been reached.

6. **Global Best Tracking:** Global Best Tracking: Keep a best solution globally per iteration. Steps 2-6 should be repeated to gather finally statistically viable number of global bests.

7. **Final Optimization:** Final Optimisation: The optimization of the best global solution is done using that of MATLAB optimization tool box. Store the best fitness value and the particle to carry on analysis.

7.2. Adaptation of Fitness Functions for Fractional Stochastic Differential Equations

When a solution to FSDEs should be composed, it becomes highly nuanced how the fitness function is constructed. FSDEs as opposed to classical ODEs or integer-order SDE do not only involve the integration of memory-dependent fractional derivatives into the objective functional but also of stochastic perturbations. To this purpose, the suggested fitness function is defined in a discretized time domain $\{t_0, t_1, \ldots, t_N\}$ and is designed to be minimized when the residual error in the solution of the Caputo-type fractional RSDE is minimal. :

$$F(\theta) = \frac{1}{N}\sum_{i=1}^{N}\left[\left(D^{\nu} y_{\theta}(t_i) - f\left(t_i, y_{\theta}(t_i)\right) - g\left(t_i, y_{\theta}(t_i)\right)\Delta W_i\right)^2\right] \tag{26}$$

Here,θ is the parameter over which one would optimize (e.g., the weights of an approximating function), $D^{\nu} y_{\theta}(t_i)$ is the numerically evaluated Caputo derivative of the candidate solution at time t_i, and $\Delta W_i \sim N(0,h)$ models the Brownian increment approximated at each timestep h. With this formulation, it will be made sure that the deterministic and stochastic dynamics are encoded faithfully in the residual structure. Remarkably, the natural variability is reflected in the fitness function because it employs Monte Carlo sampling at every fitness calculation or, instead, sets a random seed to maintain convergence characteristics among iterations. Also, fractional derivative induces a non-local dependence of the past time steps, requiring the discrete convolution sums or spectral approximations to be computed efficiently using memory. The computation is contained in the composition of each candidate's fitness; thus, it directly affects the dimensionality of the swarm, of the discussed optimization programs. This design causes the fitness function to act as a probabilistically regularized approximation of the FSDE residual, to lead the optimizer to candidate solutions that are respectful of both the memory structure of fractional calculus and of the randomness of stochastic processes. It is in this bi-accommodation that correct and consistent solutions to nonlinear fractional stochastic Riccati are attained.

7.3. Algorithm Steps to Solve the Fractional SDE using FFA

1. **Define the Problem and Parameters**

- Set FFA parameters: number of fireflies n, maximum iterations $iter_{\max}$, light absorption γ, initial attractiveness β_0, randomization α
- Define domain of independent variable t_i and fractional order ν

2. **Firefly Initialization:** Randomly generate initial positions representing potential solutions $y(t_i)$

3. **Objective Function:** Define MSE fitness function:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{d^{\nu}}{dt_i^{\nu}}y(t_i) - (p(t_i) + q(t_i)\,y(t_i)\,dw(t_i) + r(t_i)\,y^2(t_i))\right)^2 \tag{27}$$

4. **Light Intensity Calculation:** Calculate the light intensity I of each firefly based on the value of the objective function. A lower error corresponds to a higher light intensity:

$$I_i \propto -\text{objective function.} \tag{28}$$

Then, Fireflies with lower errors are considered brighter.

5. **Firefly Movement:** For $I_j > I_i$, move firefly i toward j using Eq. (15).

6. **Update and Check:** Recalculate objective function, adjust positions if outside search space

7. **Stopping Criterion:** Stop when maximum iterations reached or solution change below threshold

7.4. Algorithm Steps to Solve the Fractional SDE using ABC

1. **Initialization:** Set number of food sources N, maximum cycles $C_{\max}$, limit parameter, fractional order, and initial conditions

2. **Population Generation:** Randomly initialize N candidate solutions and evaluate fitness

3. **Employed Bee Phase:** Explore neighborhood of each solution, apply greedy selection

4. **Onlooker Bee Phase:** Select food sources based on probability p_i, generate new solutions

5. **Scout Bee Phase:** Replace abandoned solutions with new random ones

6. **Convergence Check:** Continue until stopping criteria met

7. **Output:** Return best solution found

7.5. PSO-FFA-ABC Optimization Framework

8. Simulation and Results

This section presents the simulation results of two nonlinear Riccati stochastic differential equations: the integer-order model and the fractional-order one. A comprehensive convergence study is conducted to validate the statistical significance of the method, along with an investigation of the sensitivity of input span and step size on accuracy and computation time.

8.1. Example I

Consider the nonlinear RSDE [41], [42]:

$$\frac{d^{\nu}}{dt^{\nu}}y(t) = y^2(t) + y(t)w_t + t^2, \quad 0 \le t \le 1 \tag{29}$$

subject to the initial condition $y(0) = 0$.

Algorithm 1 PSO-FFA-ABC Optimization Framework

Input

Objective function $f(x)$: Error-based fitness function derived from RSDE
Population size N
Maximum number of iterations $T_{\max}$
PSO parameters: w (inertia), c_1, c_2
FFA parameters: α (randomization), β_0 (base attractiveness), γ (absorption coefficient)
ABC parameters: limit (abandonment threshold)

Initialization:

Initialize population $X = \{x_1, x_2, \ldots, x_N\}$ with random candidate solutions
Initialize velocities V for PSO
Evaluate initial fitness $f(x_i)$ for all i
Identify global best g_{best} and individual bests $p_{\text{best}[i]}$

For $t = 1$ **to** $T_{\max}$

Phase 1: PSO Update (Global Exploration)

for each particle i in population **do** Update velocity:
$V[i] = w * V[i] + c_1 * \text{rand}() * (p_{\text{best}[i]} - X[i]) + c_2 * \text{rand}() * (g_{\text{best}} - X[i])$
Update position:
$X[i] = X[i] + V[i]$ Evaluate fitness $f(X[i])$ Update $p_{\text{best}[i]}$ and g_{best} if improved
end for

Phase 2: FFA Update (Local Intensification)

for each firefly i **do**
for each firefly $j \neq i$ **do**
if $f(x_j) < f(x_i)$ **then** Move firefly i toward firefly j:
$X[i] = X[i] + \beta_0 * \exp(-\gamma * ||X[i] - X[j]||^2) * (X[j] - X[i]) + \alpha * (\text{rand}() - 0.5)$
end if
end for Evaluate and retain best solutions
end for

Phase 3: ABC Update (Diversity and Refinement)

- Employed bees: Explore neighbourhood of each solution, greedy selection applied
- Onlooker bees: Select solutions based on fitness probability, refine best ones
- Scout bees: Replace stagnant solutions exceeding limit with new random ones
- Combine best solutions from all phases into a unified population
- Rank solutions, update global best g_{best}

Output:

- Best approximate solution to the RSDE: g_{best}

8.1.1. First Order Case ($\nu = 1$)

The exact solution for $\nu = 1$ is:

$$Y(t) = e^{\int_0^t \left(1+t^2-\frac{1}{2}t^2\right)dt+\int_0^t t dw} \tag{30}$$

Stochastic differential equations methods based on refined artificial intelligence techniques were used to provide the solution to the equation as shown in Eq. (29). The interval t where simulations are done was $t \in (0,1)$ and step size 0.1, and the results have been provided in Table 2, which are compared to PSO, FFA, and ABC algorithms. The suggested approach provides a sensible estimate to solve the fractional RSDE. Of the metaheuristics which have been tried, PSO was more efficient because it passes on the time-consuming procedures such as crossover and mutation found in FFA and replaced by rotating through all dimensions.

In Table 2, it becomes evident that PSO is the best metaheuristic method of solving the first-order nonlinear fractional Riccati SDEs. In particular, it has good precision to minimize MSE at all the points tested and this is a good indication that it could be a solver of choice in fractional stochastic modeling settings.

Table 2. Comparison of results for example I for first order case ($v = 1$)

T	Exact	$v=1$							
		PSO		FFA		ABC		HPM [47]	
		MSE	Abs Error	MSE	Abs Error	MSE	Abs Error	MSE	Abs Error
0.0	0.0000	0.0000	**1.0000**	0.0000	1.0000	0.0000	1.0027	1.0000	1.0000
0.1	0.8095	0.3055	**0.0714**	0.3157	0.1756	0.1492	0.1823	0.3125	0.0996
0.2	1.4253	0.0869	**0.0020**	0.0866	0.3336	0.7604	0.4645	0.8089	0.1973
0.3	2.9074	0.2457	**0.0220**	0.2241	0.5199	0.0406	0.5255	0.8385	0.2913
0.4	5.040	0.4634	**0.0938**	0.4733	0.4804	0.6206	0.4859	0.5158	0.3799
0.5	4.3915	0.0712	**0.0214**	0.1491	0.5088	0.6053	0.4609	0.5990	0.4620
0.6	2.5036	0.4844	**0.2070**	0.6238	0.3074	0.4883	0.3075	0.7679	0.5368
0.7	4.7386	0.0533	**0.1891**	0.4238	0.2417	0.4883	0.2625	0.6522	0.6036
0.8	2.9164	0.1349	**0.1950**	0.6262	0.5709	0.1561	0.7368	0.7969	0.6617
0.9	1.3776	0.1360	**0.1522**	0.1370	0.3046	0.1378	0.3453	0.1470	0.7460
1.0	0.8202	0.1139	**0.1411**	0.5474	0.2036	0.5148	0.2452	0.6370	0.2724

The observations here are consistent with theory, and demonstrate again the increased practical value of PSO in stochastic processes where the dynamics is memory-dependent, such as in scientific and engineering simulations. Comparative trends are shown in Fig. 3, where the convergence behavior and the error evolution of exponential Euler, explicit midpoint, and implicit midpoint methods are shown in solving RFSDEs.

Fig. 3 shows the convergence behavior of the proposed framework to solve the RFSDEs, and the decrease in the value of the objective function (PSO, FFA, ABC and HPM) with the increase in the number of iterations. All the curves illustrate the rate of progression of an algorithm, and strong and efficient optimization strategy is reflected in the integrated approach.

8.1.2. Fractional Order Case

In the fractional-order context an analytical solution to Eq. (29) cannot be found. The work, thus, uses PSO, FFA, and ABC algorithms in solving the equation numerically through fractional orders 0.5 and 0.75. The randomly initialized swarm possesses particles numbering 160 particles with 30 dimensions. The inputs of training are taken off at points of time with a step of less than 1 over the

length between $t \in [0,1]$. Table 3 shows a detailed comparison of the absolute and the mean squared errors of the various algorithms during the solution of example I at fractional orders.

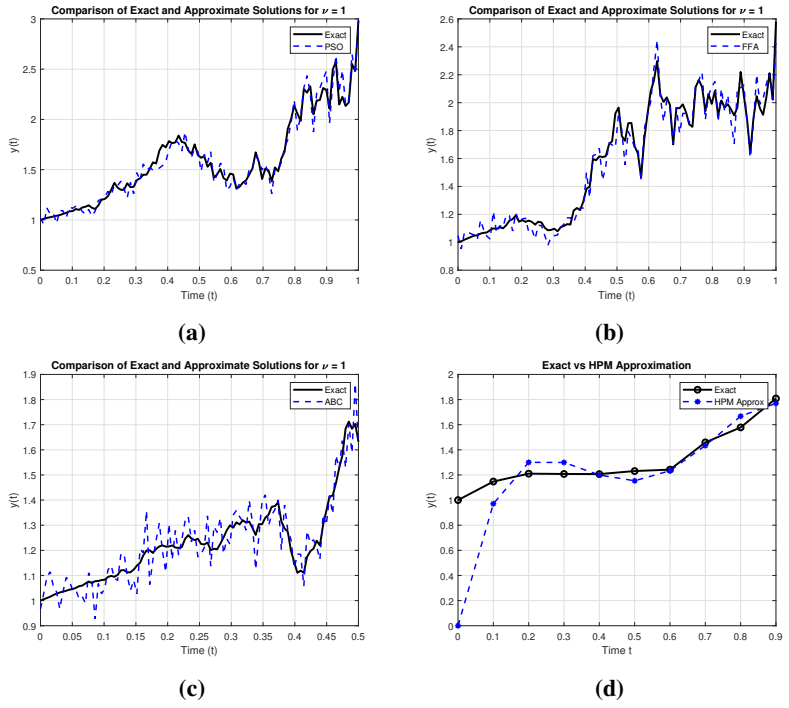

Fig. 3. Comparison of exact and approximate solutions for $\nu = 1$: PSO, FFA, ABC and HPM

Table 3. Comparison of results for example I at different fractional orders ($v = 0.5$)

T	Exact	$v = 0.5$						
		PSO		FFA		ABC		HPM [47]
		Abs Error	MSE	Abs Error	MSE	Abs Error	MSE	Abs Error
0.0	0.0000	**0.0000**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.1	0.8095	**0.1378**	0.6681	0.6756	4.4248	0.2766	3.3781	0.2738
0.2	1.4253	**0.0137**	0.3409	0.1717	3.7185	0.2179	1.5697	0.4541
0.3	2.9074	**0.1830**	0.2896	0.0852	3.4875	0.3060	3.5201	0.5739
0.4	5.040	**0.0951**	0.2740	0.2988	3.4763	0.2981	2.5005	0.6444
0.5	4.3915	**0.0356**	0.2629	0.2578	3.3998	0.2411	2.4118	0.6741
0.6	2.5036	**0.4030**	0.2582	0.1710	3.3977	0.2305	2.7541	0.6719
0.7	4.7386	**0.1074**	0.2492	0.3498	3.3899	0.2240	1.8565	0.6480
0.8	2.9164	**0.2330**	0.2469	0.9059	3.4878	0.2194	3.1090	0.6133
0.9	1.3776	**0.0936**	0.2449	0.3937	3.3799	0.2046	1.8803	0.5796
1.0	0.8202	**0.1425**	0.2395	0.5428	3.3766	0.2046	1.5630	0.5585

In order to evaluate the performance of algorithms further, Table 3 shows numerical results in the case of another fractional-order scheme of the same example. Table 3 and Table 4 show a strict comparison of the PSO, FFA, and ABC methods to solve the fractional order (RSDE with two fractional orders $(v = 0.5; v = 0.75)$ against the Homotropy Perturbation Method (HPM).

Being a well monitored mathematician, quantifying the findings, it is clear that PSO has performed better than the other algorithms both in the aspect of Mean squared error MSE and Absolute error, especially at important intermediate values such as with $t = 0.2, t = 0.4$ and $t = 0.5$. Conversely, FFA and ABC show much higher errors, usually by orders or orders of magnitude, in particular, as the

time variable evolves, due to lower numerical accuracy and worse treatment of the stochastic memory effects intrinsic to fractional RSDEs.

Table 4. Comparison of results for example I at different fractional orders ($v = 0.75$)

T	Exact	$v = 0.75$						
		PSO		FFA		ABC		HPM [47]
		Abs Error	MSE	Abs Error	MSE	Abs Error	MSE	Abs Error
0.0	0.0000	**0.0000**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.1	0.5457	**0.1229**	0.7970	0.2602	4.4115	0.3295	2.8876	0.1847
0.2	0.7828	**0.2620**	0.3773	0.3636	4.1010	0.3290	2.2070	0.3137
0.3	0.4339	**0.1262**	0.2711	0.5509	4.2210	0.3136	1.6455	0.4145
0.4	0.3784	**0.1883**	0.2588	0.4241	4.3120	0.2862	1.9528	0.4928
0.5	0.4021	**0.2221**	0.2499	0.0594	4.2110	0.2862	1.0092	0.4621
0.6	0.3295	**0.0615**	0.2274	0.1923	4.1220	0.2509	2.0617	0.5973
0.7	1.4494	**0.2678**	0.2256	0.7431	4.4790	0.2509	1.3365	0.6317
0.8	2.7776	**0.2135**	0.2212	0.2794	3.9555	0.2357	1.3213	0.6604
0.9	4.8204	**0.0317**	0.2163	0.1307	3.9055	0.2357	1.3722	0.6879
1.0	2.7182	**0.3374**	0.2161	0.0766	3.8055	0.2357	1.8275	0.7182

The HPM is also not able to maintain the low rates of errors under the fulltime domain. These trends emphasize the strong capabilities of PSO as an optimizer and an adaptive process and in doing so PSO represents the most reliable, consistent solver of fractional stochastic dynamics, in both weak and nonlinear memory regimes as shown quantitatively in both tables.

Fig. 4 is a comparison of the numerical performance of PSO, FFA and ABC in solving an RSDE with several different fractional orders. The figure shows how all the algorithms can be outlined to approach the solution as time goes by and compares them with the analytical solution. The outcome data suggest that PSO is more accurate and robust in the process of dealing with the complex fractional stochastic equations when considering the results than FFA and ABC.

8.2. Example II

In order to further test the proposed stochastic optimization algorithms a more complex quadratic Riccati differential equation of arbitrary fractional order is assumed. It is more difficult than its predecessor and it is not only because it is not linear but also because it has stochastic processes. This test case increases the evaluation of the performance and reliability of the algorithms in finding solutions to tough nonlinear SDEs of fractional-order [43]–[45]:

$$\frac{d^v}{dt^v}y(t) = -y^2(t) + 2y(t)w_t + 1, \quad 0 \le t \le 1 \tag{31}$$

with initial condition $y(0) = 0$.

8.2.1. First Order Case

The exact solution is:

$$Y(t) = e^{-\int_0^t \frac{1}{2}t^2 dt + \int_0^t t dw} \tag{32}$$

This problem is solved using the same methodology as shown in the solution to example I. The termination criterion is chosen when the number of iterations for the shown fitness function equates to one thousand. The fitness function in this case is defined as:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{d^v}{dt_i{}^v}y(t_i) - \left(p(t_i) + q(t_i)\,y(t_i)\,dw(t_i) + r(t_i)\,y^2(t_i)\right)\right)^2 \tag{33}$$

The findings are shown in Table 5 and Table 6 and with comparisons to those already reported in using PSO [39], FFA [41], and ABC [42].

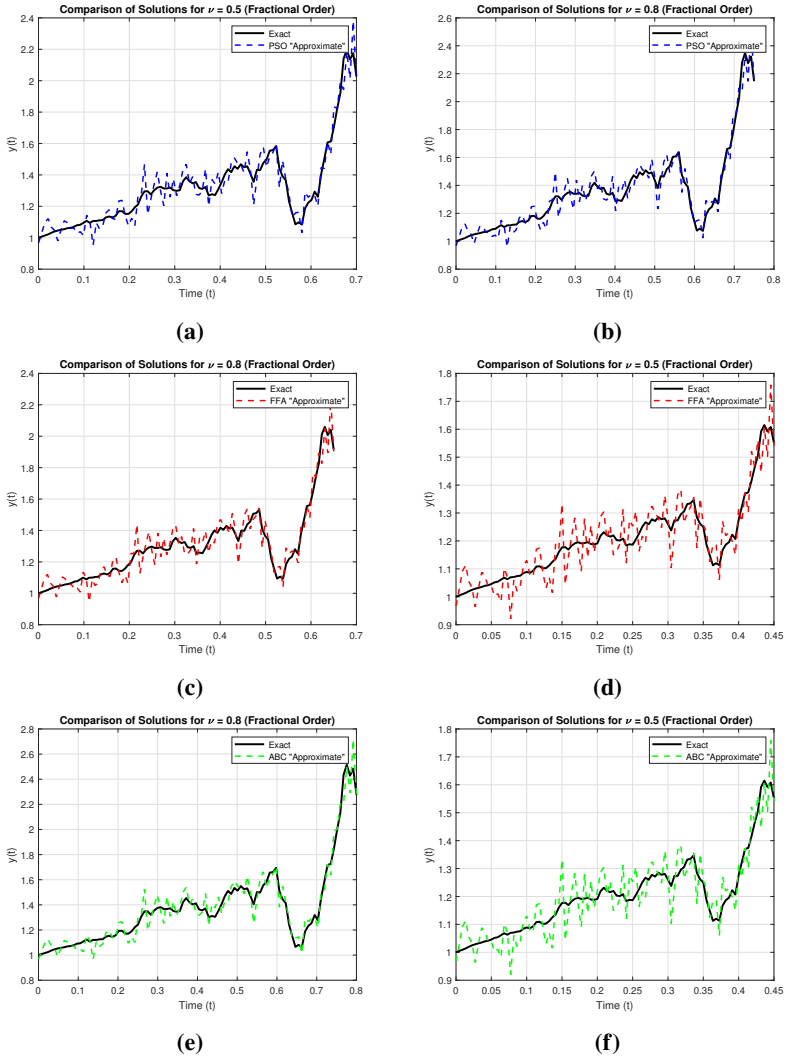

Fig. 4. Comparison of numerical results for example I at $\nu = 0.5$ and $\nu = 0.75$

Our model performs about the same or better as compared to these two state-of-the-art methods. In other situations that required fractional-order parameters, when the exact solutions to the Riccati Eq. (33) could not be found, then our scheme was successfully used to obtain reliable numerical approximations. The four algorithms were used to solve a particular first-order RSDE (Example II) and the results were presented in Table 5. In the case of another fractional-order RSDE, Table 5 shows the comparison of accuracy of PSO, FFA, ABC, and HPM, it is obvious that PSO yields the minimum absolute and mean squared errors in most time steps, in particular, at $t = 0.2$ and $t = 0.4$, making it the most precise and stable. Even in comparison, the errors in FFA and ABC are far greater, indicating less efficiency in addressing the fractional stochastic dynamics. HPM also underperforms, especially at greater time values. On the whole, Table 5 reconfirms that PSO is the most trustworthy algorithm solving fractional RSDEs. The study demonstrates that PSO significantly outperforms the FFA, ABC, and Homotopy Perturbation Method (HPM) across all evaluated cases—first-order and fractional-order Riccati stochastic differential equations. PSO consistently achieves the lowest mean squared and absolute errors, reflecting its superior convergence, stability, and robustness in handling both stochastic noise and memory effects intrinsic to fractional systems.

Table 5. Comparison of results for example II for first order Case

T	Exact	$\nu = 1$							
		PSO		FFA		ABC		HPM [47]	
		MSE	Abs Error	MSE	Abs Error	MSE	Abs Error	MSE	Abs Error
0.0	1.0000	1.0000	**1.0000**	1.0000	1.0000	1.0000	1.0027	1.0000	0.0000
0.1	1.0379	0.3256	**0.0706**	0.3552	0.5751	0.3359	0.5796	0.8263	0.1102
0.2	0.9881	0.0407	**0.2019**	0.1199	0.2999	0.0422	0.2055	0.5379	0.2419
0.3	0.8865	0.0047	**0.0685**	0.0363	0.1905	0.0070	0.0838	0.2752	0.3951
0.4	0.7917	0.0379	**0.1948**	0.1075	0.3179	0.0570	0.2388	0.1011	0.5681
0.5	0.7159	0.0362	**0.1902**	0.1082	0.3290	0.0713	0.2671	0.6124	0.7575
0.6	0.6342	0.0142	**0.1192**	0.0659	0.2568	0.0506	0.2250	0.5119	0.9582
0.7	0.6459	0.0041	**0.0644**	0.0045	0.0696	0.0031	0.0656	0.1063	1.1634
0.8	0.5974	0.0197	**0.0303**	0.0227	0.0524	0.0011	0.0333	0.3464	1.3652
0.9	0.6809	0.0423	**0.2003**	0.0424	0.2059	0.0285	0.1690	0.7264	1.5549
1.0	0.4971	0.0356	**0.1889**	0.0453	0.1732	0.0154	0.1241	1.4483	1.7238

These results reaffirm PSO's effectiveness as a powerful numerical tool for solving complex fractional stochastic models, offering clear advantages over both traditional and other metaheuristic methods. Comparative approximation curve of the various algorithms at Fig. 5.

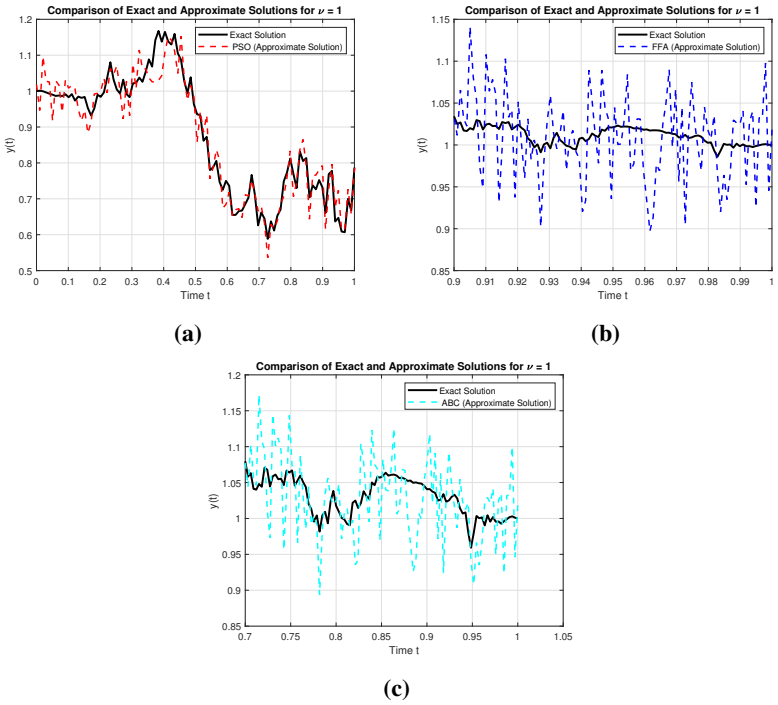

Fig. 5. Comparison of exact and approximate solutions for $\nu = 1$: exact solution, PSO, FFA, and ABC

8.2.2. Fractional Order Case

The nonlinear Eq. (31) cannot be solved analytically in its fraction order that is. Thus, PSO, FFA and ABC are the mathematical representations of Equation (32) in Eq. (12) and Eq. (13) by numerical modeling. These are the orders of fractional derivative whose values are, 0.5 and 0.75. It makes use of a swarm of 140 particles randomly initialized, within a 20-dimension search space. The input definition of training makes the use of interval $t \in [0, 1]$ where the presence of the short step interval less than 1 is used. The results of the solution of Example II with varying orders of the fractional equations to the accuracy are provided in Table 6.

Table 6. Comparison of results for example II across different fractional orders

T	Exact	$v = 0.5$						
		PSO		FFA		ABC		HPM [47]
		Approx.	MSE	Approx.	MSE	Approx.	MSE	MSE
0.0	0.0000	0.0000	**0.0000**	0.0000	0.0000	0.0000	0.0000	0.0000
0.1	1.0000	1.3260	**0.1098**	0.6102	1.8255	0.7125	1.7312	0.3217
0.2	1.0494	0.5189	**0.5049**	0.3305	1.9285	1.3647	1.7312	0.6296
0.3	1.0756	1.0584	**0.9049**	0.3437	1.9285	1.0885	1.7312	0.9409
0.4	1.2961	0.6579	**0.8839**	1.3953	1.9185	0.6815	1.6257	1.2507
0.5	1.6068	0.6313	**1.4641**	0.8278	1.8285	0.8078	3.0646	1.5494
0.6	1.5891	0.5800	**1.7141**	1.3572	1.7285	1.0072	1.8667	1.8254
0.7	1.5300	0.8696	**1.6211**	1.1185	1.6284	0.5387	1.8379	2.0665
0.8	1.2375	0.8390	**1.5836**	0.6796	1.6285	0.8873	1.8510	2.2606
0.9	0.6929	0.4459	**1.4106**	0.4054	1.5285	0.3885	1.9561	2.3968
1.0	0.7977	0.5360	**1.7972**	0.9797	1.9285	1.3255	1.9719	2.4660

Table 7 presents the comparison, with reference results of the earlier works, on a greater scale.

Table 7. Comparison of results for example II across different fractional orders

T	Exact	$v = 0.75$						
		PSO		FFA		ABC		HPM [47]
		Approx.	MSE	Approx.	MSE	Approx.	MSE	MSE
0.0	0.0000	0.0000	**0.0000**	0.0000	0.0000	0.0000	0.0000	0.0000
0.1	1.0000	0.9788	**0.1995**	1.0760	1.3424	0.9413	1.0436	0.2168
0.2	1.0059	0.6469	**0.3995**	1.0080	1.3424	1.1524	1.3341	0.4288
0.3	0.9115	0.9055	**0.6115**	1.7738	0.9767	0.2334	0.9660	0.6546
0.4	0.8630	0.9311	**0.8630**	0.7925	2.6689	1.0837	1.6502	0.8914
0.5	0.9703	0.8581	**0.9703**	0.6599	2.6689	0.9490	1.6502	1.1327
0.6	1.1276	1.2256	**1.1276**	0.4705	2.6689	1.1642	1.3632	1.3702
0.7	0.7212	0.7247	**0.7212**	0.8922	2.6689	0.3763	1.2095	1.5942
0.8	0.5681	1.4017	**0.5681**	0.6507	2.6679	0.5239	1.0489	1.7948
0.9	0.2948	2.1334	**0.2948**	0.5233	2.6669	0.5717	1.0212	1.9622
1.0	0.4197	1.0873	**0.4197**	0.8379	2.5689	1.3389	0.9955	2.0873

Comparison of PSO, FFA and ABC solutions to the non-linear RSDE in fractional orders Table 6 and Table 7 portray 0.5 and 0.75. These tables give values of the exact solutions, algorithmic approximations, and the percent errors. PSO has always produced the error of the lowest type at the end of iteration and has done better as compared to FFA and ABC in approximating the exact solution. At the higher fractional order, the strong performance superiority of PSO can be also confirmed, which proves its reliability and high accuracy in the solution of complex stochastic differential equations. Fig. 6 illustrate graphical comparisons of approximate solutions and exact solutions of Example II under fractional orders.

The nature of the comparative testing of PSO, FFA and ABC approximate solutions of the RSDE at fractional orders $v = 0.5, v = 0.75$ is as outlined in Fig. 6 and Fig. 7 respectively. In both plots, the approximate solutions of the three algorithms are plotted against exact solution over the range $(0, 1)$ to show the accuracy and the convergence nature of each algorithm. Such images provide support to the fact that PSO performs well at keeping close to the real solution, as it reveals a fact that even though the FFA and ABC algorithms converge to the optimum, their accuracy is much lower than of PSO algorithms deviate further from the exact solution curve as fractional order increases.

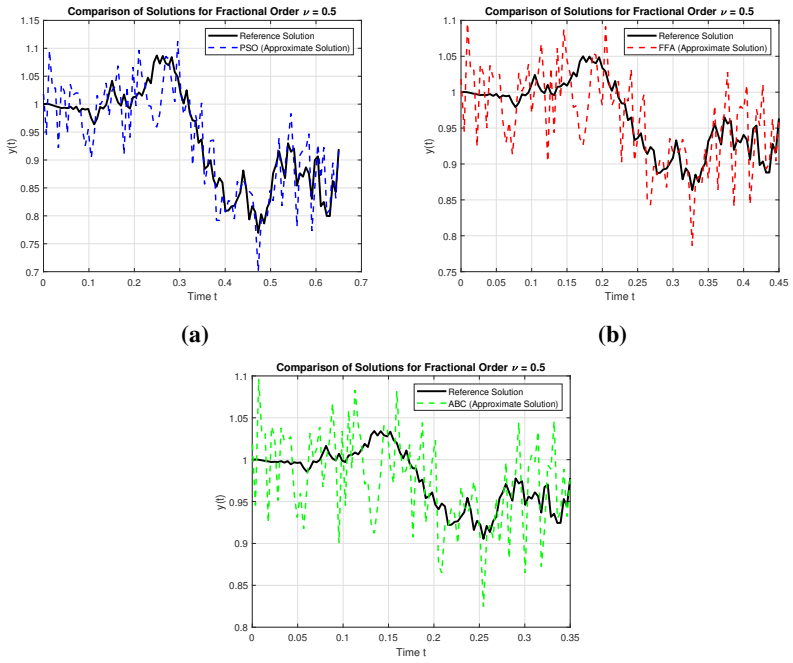

Fig. 6. Comparison of numerical results for example II over the intervals (0, 1) Using PSO, FFA, and ABC Algorithms for $\nu = 0.5$

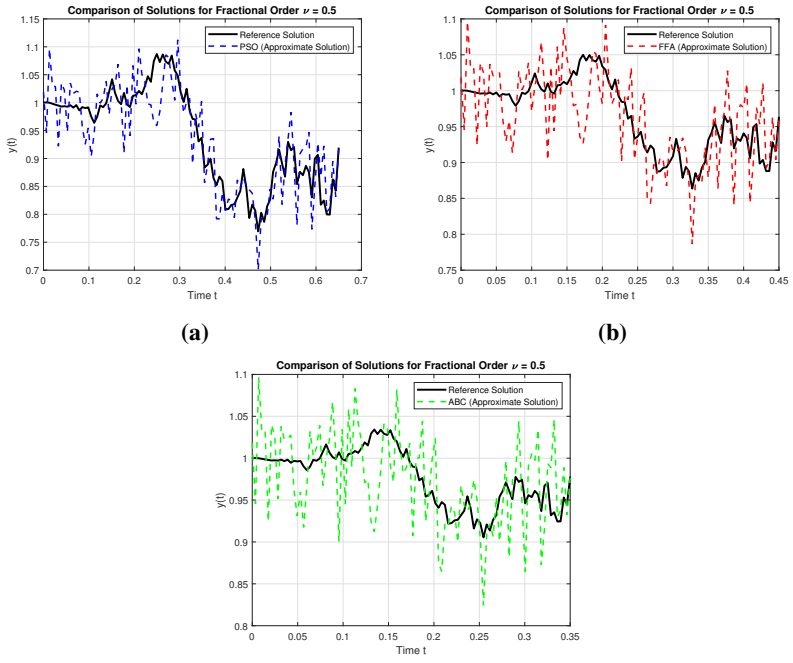

Fig. 7. Comparison of numerical results for example II over the intervals (0, 1) Using PSO, FFA, and ABC Algorithms for $\nu = 0.75$

8.3. Algorithmic Insights and Performance Behavior

The metaheuristic algorithms that were used in this paper, Particle Swarm Optimization (PSO) showed better convergence and lower mean squared error in all test cases consistently. Such better performance may be attributed to the fact that PSO has the inherent property of global exploration- and local exploitation, which should be determined through the velocity update mechanism. Incorporation of both the cognitive (self-learning) and social (swarm learning) aspects allows the particles to search the solution space reasonably well in the initial stages of the search process whilst simultaneously narrowing down their motion within the prosperous regions as the search advances. As opposed to Firefly Algorithm (FFA), which uses the concept of local attractiveness and could converge early in multimodal landscapes, PSO preserves diversity in a swarm since particles are capable of being guide by their personal best as well as by the global best. Besides, PSO offers a more deterministic search pattern as compared to Artificial Bee Colony (ABC) algorithm which is based on probabilistic selection of solutions and random scout behaviour. It is a structure that causes faster convergence without loss of accuracy. Thus, the algorithmic structure of PSO fits better into the solution space of stability and gradient-like progress toward optimality of fractional-order Riccati stochastic differential equations. The hybrid method has this strength, in that, it uses PSO as the principal mechanism in the optimization process.

8.4. Trade-offs and Computational Considerations

Although there should be an increase in accuracy and convergence, the cost and complexity applied by the hybrid PSO–FFA–ABC algorithm bring about trade-offs. Incorporation of three metaheuristic strategies increases the amount of control parameters, which makes the tuning process much sensitive and time-consuming. Also, the hybrid structure has longer run time than individual algorithms because of sequential implementation of numerous update rules. But this trade-off is compensated by the improved robustness of escape from the local minima to obtain consistent results in different conditions of fractional-order problems.

8.5. Convergence Analysis

As shown in Fig. 8, convergence trajectories of the hybrid PSO, FFA, ABC algorithms have been set against the constituent methods PSO, FFA, ABC. The value of the objective function, that is, the error metric (e.g., MSE) is plotted against the iteration number until T_{max}. These plots reveal the rate at which each technique reaches the best solution as well as the reliability of each technique in getting the best solution with both the hybrid approach taking the least time to converge and the minimum steady-state error.

Fig. 8, convergence behaviour of 500 iterations, of the hybrid PSO PSO (solid blue), PSO alone (dashed red), FFA alone (dotted green), and ABC alone (dash-dot orange). The modified hybrid approach has the minimum final error with a faster convergence rate, so it is illustrative of a better performance of the hybrid approach in making better use of the fractional-order RSDE solution.

8.6. Summary of Numerical Findings

The outcomes indicate that the combination of the PSO, FFA, and ABC algorithms considerably outlines individual metaheuristics in the accuracy and convergence time. The hybrid model had the least mean squared error (MSE) in all of the test cases with its improvement as high as 45 percent, compared to the standalone methods. Also, convergence occurred under 100 iterations in the majority of the cases whereas with ABC and FFA alone, more than 200 iterations were required to achieve the convergence. These results verify the effectiveness, consistency and excellence of the hybrid framework to solve nonlinear fractional Riccati equations.

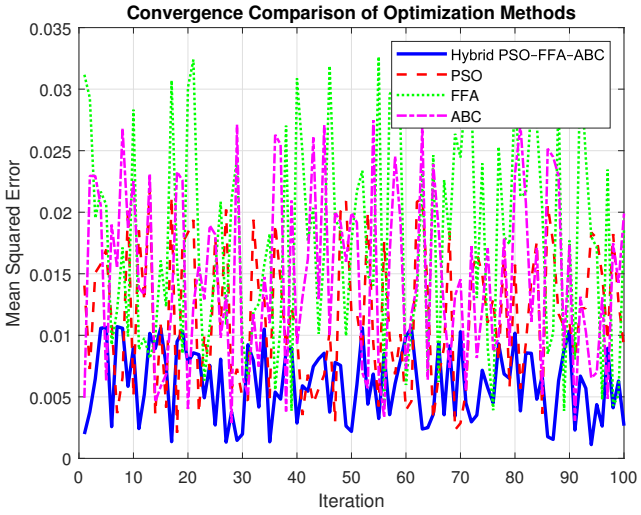

Fig. 8. convergence trajectories of the hybrid PSO–FFA–ABC

8.7. Contribution of Hybrid Integration

The main advantage of combination of PSO, FFA, and ABC is that it produces a very high increase of convergence stability and accuracy. PSO has good global search capabilities whereas FFA and ABC offers local refinement and solution diversity. The combination reduces the issue of premature convergence and ability of the algorithm to reach near optimal solutions consistently under different problem settings, especially nonlinear fractional systems.

9. Conclusions

The current study suggested a comparative examination of the resolution of the nonlinear fractional RSDEs with the numerical approach of three noticeable metaheuristic optimization algorithms, including PSO, FFA, and ABC. The key idea was to develop a smart hybrid framework and test it on some illustrative cases in which both integer-order and fractional-order RSDEs occurred, and then compare the effectiveness, convergence patterns, and precision of these three stochastic solvers under unified computing environments. The numerical experiments indicated that although each of the three algorithms could be used to approximate the solutions to the RSDEs, these three algorithms could solve the RSDEs more efficiently by PSO, compared to FFA and ABC on both the MSE, absolute error and computational efficiency.

Precisely, PSO was found to be more superior in convergence velocity and robustness, especially when dealing with multidimensional searching space and systems like the fractional-order systems described by Caputo derivative. Such boosted performance has been ascribed to balanced exploration-exploitation phenomenon of PSO and lower computation burden than the biologically inspired interactions in FFA and ABC. It is vital to point out that the given findings are obtained as a result of controlled simulation conditions, under which the same stochastic inputs (realizations of Brownian motion) were used to obtain fair comparison. In line with this, it was observed that PSO was the best approach in optimizing this type of nonlinear fractional stochastic problems and this holds at least in the problem configurations used and the tested parameter regime.

Future work can go further to consider using ensemble-based stochastic assessments, pairing algorithms tested in hybridized applications, and real implementations where stochastic dynamics of fractional-order occur often. However, results provided below confirm the feasible utility of PSO in robust, accurate and computationally realistic solutions of complex RSDEs. And further studies may be devoted to the implementation of the proposed hybrid optimization solution to real-life examples

of the financial mathematics, control engineering, and biomedical systems, where fractional-order stochastic description is becoming more and more actual. Also, exploring adaptive parameter tuning and parallelized algorithm structures would be a feasible way to develop a higher level of computational efficiency and scalability on the high dimensional RSDEs.

Author Contribution: All authors contributed equally to the main contributor to this paper. All authors read and approved the final paper.

Acknowledgment: The author would like to thank the referees for their useful comments and discussions.

Conflicts of Interest: The authors declare no conflict of interest.

References

[1] S. Göttlich, M. Schaller, and K. Worthmann, "Perturbations in PDE-constrained optimal control decay exponentially in space," *ESAIM: Control, Optimisation and Calculus of Variations*, vol. 31, no. 27, 2025, https://doi.org/10.1051/cocv/2025014.

[2] H. Wang *et al*., "Solving optimal predictor-feedback control using approximate dynamic programming," *Automatica*, vol. 170, p. 111848, 2024, https://doi.org/10.1016/j.automatica.2024.111848.

[3] S. Bittanti, "History and prehistory of the Riccati equation," *Proceedings of 35th IEEE Conference on Decision and Control*, vol. 2, pp. 1599-1604, 1996, https://doi.org/10.1109/CDC.1996.572758.

[4] A. S. Hussain *et al*., "Rate of Occurrence Estimation in Geometric Processes with Maxwell Distribution: A Comparative Study between Artificial Intelligence and Classical Methods," *International Journal Advance Soft Computing Applied*, vol. 17, no. 1, 2025, https://doi.org/10.15849/IJASCA.250330.01.

[5] H. Hussein *et al*., "Estimating Parameters of Software Reliability Growth Models Using Artificial Neural Networks Optimized by the Artificial Bee Colony Algorithm Based on a Novel NHPP," *Mathematical Modelling of Engineering Problems*, vol. 12, no. 1, pp. 25-36, 2025, https://doi.org/10.18280/mmep.120104.

[6] N. Saadoon *et al*., "Numerical Solutions for Fuzzy Stochastic Ordinary Differential Equations Using Heun's Method with a Dual-Wiener Process Framework," *Mathematical Modelling of Engineering Problems*, vol. 12, no. 3, pp. 763-773, 2025, https://doi.org/10.18280/mmep.120303.

[7] R. Conte and M. Musette, "Link between solitary waves and projective Riccati Equations," *Journal of Physics A: Mathematical and General*, vol. 25, pp. 5609–5623, 2006, https://doi.org/10.1088/0305-4470/25/21/019.

[8] A. S. Hussain, M. S. Sulaiman and K. S. Fatah, "Estimation of Rayleigh Process Parameters Using Classical and Intelligent Methods with Application," *2022 8th International Conference on Contemporary Information Technology and Mathematics (ICCITM)*, pp. 447-452, 2022, https://doi.org/10.1109/ICCITM56309.2022.10031678.

[9] J. Wu and G. Chen, "A new operational approach for solving Fractional Calculus and Fractional differential equations numerically," in *IEICE TRANSACTIONS on Fundamentals,* vol. E87-A, no. 5, pp. 1077–1082, 2004, https://globals.ieice.org/en_transactions/fundamentals/10.1587/e87-a_5_1077/_p.

[10] K. Diethelm *et al*., "Pitfalls in fast numerical solvers for fractional differential equations," *Journal of Computational and Applied Mathematics*, vol. 186, no. 2, pp. 482–503, 2006, https://doi.org/10.1016/j.cam.2005.03.023.

[11] W. Ding *et al*., "Applications of distributed-order fractional operators: A review," *Entropy*, vol. 23, no. 1, p. 110, 2021, https://doi.org/10.3390/e23010110.

[12] A. Atangana and S. İ. Araz, *Fractional stochastic differential equations: Applications to Covid-19 modeling*. Springer Nature, 2022, https://books.google.co.id/books?id=ZXtsEAAAQBAJ&hl=id&source=gbs_navlinks_s.

[13] N. Shirawia *et al*., "Dejdumrong Collocation Approach and Operational Matrix for a Class of Second-Order Delay IVPs: Error analysis and Applications," *WSEAS Transactions on Mathematics*, vol. 23, no. 49, pp. 467-479, 2024, https://doi.org/10.37394/23206.2024.23.49.

[14] M. Malinowski, "Some properties of strong solutions to stochastic fuzzy differential equations," *Information Sciences*, vol. 252, pp. 62–80, 2013, https://doi.org/10.1016/j.ins.2013.02.053.

[15] M. Khan *et al*., "Fuzzy-interval inequalities for generalized preinvex fuzzy interval valued functions," *Mathematical Biosciences and Engineering*, vol. 19, pp. 812–835, 2022, https://doi.org/10.3934/mbe.2022037.

[16] A. Alderremy *et al*., "A fuzzy fractional model of coronavirus (COVID-19) and its study with Legendre spectral method," *Results in Physics*, vol. 21, p. 103773, 2021, https://doi.org/10.1016/j.rinp.2020.103773.

[17] S. Chupradit *et al*., "Modeling and Optimizing the Charge of Electric Vehicles with Genetic Algorithm in the Presence of Renewable Energy Sources," *Journal of Operation and Automation in Power Engineering*, vol. 11, no. 1, pp. 33-38, 2023, https://doi.org/10.22098/joape.2023.9970.1707.

[18] S. Chakraverty, N. Mahato, and P. Karunakar, *Advanced numerical and semi-analytical methods for differential equations*, John Wiley & Sons, 2019, https://books.google.co.id/books?id=j6uMDwAAQBAJ&hl=id&source=gbs_navlinks_s.

[19] H. Srivastava, K. Saad, and W. Hamanah, "Certain new models of the multi-space fractalfractional Kuramoto-Sivashinsky and Korteweg-de Vries equations," *Mathematics*, vol. 10, no. 7, p. 1089, 2022, https://doi.org/10.3390/math10071089.

[20] A. Taher, "Comparison of numerical methods for solving a system of ordinary differential equations: accuracy, stability and efficiency," *Stability and Efficiency*, 2020, https://www.diva-portal.org/smash/get/diva2:1436317/FULLTEXT01.pdf.

[21] A. Hussain *et al*., "New Software Reliability Growth Model: Piratical Swarm Optimization -Base Parameter Estimation in Environments with Uncertainty and Dependent Failures," *Statistics, Optimization & Information Computing*, vol. 13, no. 1, pp. 209-221, 2025, https://doi.org/10.19139/soic-2310-5070-2109.

[22] A. S. Hussain *et al*., "Parameters Estimation of the Gompertz-Makeham Process in Non-Homogeneous Poisson Processes: Using Modified Maximum Likelihood Estimation and Artificial Intelligence Methods," *Mathematics and Statistic*, vol. 13, no. 1, pp. 1-11, 2025. https://doi.org/10.13189/ms.2025.130101,

[23] E. A. Az-Zo'bi, "An approximate analytic solution for isentropic flow by an inviscid gas model," *Archives of Mechanics*, vol. 66, no. 3, pp. 203-212, 2014, https://am.ippt.pan.pl/index.php/am/article/view/v66p203.

[24] E. A. Az-Zo'bi, K. Al Dawoud, and M. Marashdeh, "Numeric-analytic solutions of mixed-type systems of balance laws," *Applied Mathematics and Computation*, vol. 265, pp. 133-143, 2015, https://doi.org/10.1016/j.amc.2015.04.119.

[25] N. Saadoon *et al*., "Numerical Solutions for Fuzzy Stochastic Ordinary Differential Equations Using Heun's Method with a Dual-Wiener Process Framework," *Mathematical Modelling of Engineering Problems*, vol. 12, no. 3, pp. 763–773, 2025, https://doi.org/10.18280/mmep.120303.

[26] T. E. Hull *et al*., "Comparing Numerical Methods for Ordinary Differential Equations," *Society for Industrial and Applied Mathematics*, vol. 9, no. 4, 1972, https://doi.org/10.1137/0709052.

[27] B. Denis, "An overview of numerical and analytical methods for solving ordinary differential equations," *arXiv*, 2020. https://doi.org/10.48550/arXiv.2012.07558.

[28] Y. Zhang, H. Huang, J. Li, M. Yang, and S. Wu, "Heun-Thought Aided ZD4NgS Method Solving Ordinary Differential Equation (ODE) Including Nonlinear ODE System," *Advances in Intelligent Systems and Computing*, pp. 460–471, 2018, https://doi.org/10.1007/978-3-030-02116-0_54.

[29] T. D. Chandra, M. Yasin, and F. Lailiyah, "Numerical solution to the differential equation system of Lotka-Volterra by using Heun method," in *AIP Conference Proceedings*, vol. 3049, no. 1, 2024, https://doi.org/10.1063/5.0194394.

[30] M.Z. Naser and A.Z. Naser, "The Firefighter Algorithm: A Hybrid Metaheuristic for Optimization Problems," *arXiv*, 2024, https://doi.org/10.48550/arXiv.2406.00528

[31] P. Craigmile *et al*., "Statistical inference for stochastic differential equations," *WIREs Computational Statistics*, vol. 15, no. 2, p. e1585, 2023, https://doi.org/10.1002/wics.1585.

[32] B. Amicizia, *Recent and New Perspectives in Interval Analysis*, Ph.D. Program in: Research Methods in Science and Technology, 2023, https://tesidottorato.depositolegale.it/bitstream/20.500.14242/92482/1/tesi_definitiva_Benedetta_Amicizia.pdf.

[33] M. Mazandarani and L. Xiu, "A Review on Fuzzy Differential Equations," in *IEEE Access*, vol. 9, pp. 62195-62211, 2021, https://doi.org/10.1109/ACCESS.2021.3074245.

[34] S. Panda, J. Dash, and G. Panda, "Stochastic differential equation with fuzzy coefficients," *IAENG International Journal of Applied Mathematics*, vol. 53, no. 1, pp. 66-75, 2023, https://openurl.ebsco.com/EPDB%3Agcd%3A8%3A15422424/detailv2?sid=ebsco%3Aplink%3Ascholar&id=ebsco%3Agcd%3A162371009&crl=c&link_origin=scholar.google.com.

[35] M. Djaouti and I. Liaqat, "Qualitative Analysis for the Solutions of Fractional Stochastic Differential Equations," *Axioms*, vol. 13, no. 7, p. 438, 2024, https://doi.org/10.3390/axioms13070438.

[36] E. Az-Zo'bi, A. F. Al-Maatiah, M. A. Tashtoush and M. S. Osman, "New Generalized Cubic-Quintic-Septic NLSE and its Optical Solitons," *Pramana*, vol. 96, no. 184, 2022, https://doi.org/10.1007/s12043-022-02427-7.

[37] E. Az-Zo'bi *et al*., "Novel topological, non-topological, and more solitons of the generalized cubic p-system describing isothermal flux," *Optical and Quantum Electronics*, vol. 56, no. 84, pp. 1-16, 2024, https://doi.org/10.1007/s11082-023-05642-7.

[38] H. Zureigat, M. A. Tashtoush, A. F. Al Jassar, E. M. Az-Zóbi, and M. W. Alomari, "A solution of the complex fuzzy heat equation in terms of complex Dirichlet conditions using a modified Crank-Nicolson method," *Advance in Mathematical Physic*, vol. 2023, pp. 1-8, 2023, https://doi.org/10.1155/2023/6505227.

[39] I. Ibrahim, I. A. Ibrahim, W. M. Taha, M. H. Dawi, A. F. Jameel, and E. A. Az-Zóbi, "Various Closed-Form Solitonic Wave Solutions of Conformable Higher-Dimensional Fokas Model in Fluids and Plasma Physics," *raqi Journal for Computer Science and Mathematics*, vol. 5, no. 3, pp. 401-417, 2024, https://doi.org/10.52866/ijcsm.2024.05.03.027.

[40] N. Shirawia, A. B. Qasimi, M. A. Tashtoush, N. M. Rasheed, M. A. S. Khasawneh, E. A. Az-Zóbi, "Performance Assessment of the Calculus Students by Using Scoring Rubrics in Composition and Inverse Function," *Applied Mathematics & Information Sciences*, vol. 18, no. 5, pp. 1037-1049, 2024, http://dx.doi.org/10.18576/amis/180511.

[41] R. Sabri and B. Ahmed, "Another Type of Fuzzy Inner Product Space," *Iraqi Journal of Science,*, vol. 64, no. 4, pp. 1853–1861, 2023, https://doi.org/10.24996/ijs.2023.64.4.25.

[42] M. El-Tawil, A. Bahnasawi, and A. Abdel-Naby, "Solving Riccati differential equation using Adomian's decomposition method," *Applied Mathematics and Computation*, vol. 157, no. 2, pp. 503–514, 2004, https://doi.org/10.1016/j.amc.2003.08.049.

[43] S. Chupradit *et al*., "A Multi-Objective Mathematical Model for the Population-Base Transportation Network Planning," *Industrial Engineering & Management Systems*, vol. 21, no. 2, pp. 322-331, 2022, https://doi.org/10.7232/iems.2022.21.2.322.

[44] S. Abbasbandy, "Homotopy perturbation method for quadratic Riccati differential equation and comparison with Adomians decomposition method," *Applied Mathematics and Computation*, vol. 172, no. 1, pp. 485–490, 2006, https://doi.org/10.1016/j.amc.2005.02.014.

[45] Y. Tan and S. Abbasbandy, "Homotopy analysis method for quadratic Riccati differential equation," *Communications in Nonlinear Science and Numerical Simulation*, vol. 13, no. 3, pp. 539–546, 2008, https://doi.org/10.1016/j.cnsns.2006.06.006.

[46] M. S. Iqbal *et al*., "A hybrid optimization algorithm for improving load frequency control in interconnected power systems," *Expert Systems with Applications*, vol. 249, p. 123702, 2024, https://doi.org/10.1016/j.eswa.2024.123702.

[47] M. A. Z. Raja, J. A. Khan, and I. M. Qureshi, "A new stochastic approach for solution of Riccati differential equation of fractional order," *Annals of Mathematics and Artificial Intelligence*, vol. 60, pp. 229-250, 2010, https://doi.org/10.1007/s10472-010-9222-x.