OPEN ACCESS
Fractals, Vol. 31, No. 4 (2023) 2340043 (13 pages)

DOI: 10.1142/S0218348X23400431





# A NEW FORM OF L1-PREDICTOR–CORRECTOR SCHEME TO SOLVE MULTIPLE DELAY-TYPE FRACTIONAL ORDER SYSTEMS WITH THE EXAMPLE OF A NEURAL NETWORK MODEL

PUSHPENDRA KUMAR,[*,§] VEDAT SUAT ERTURK,[†]
MARINA MURILLO-ARCILA[‡] and V. GOVINDARAJ[*]
[*]*Department of Mathematics*
*National Institute of Technology Puducherry*
*Karaikal 609609, India*
[†]*Department of Mathematics, Faculty of Arts and Sciences*
*Ondokuz Mayis University*
*Atakum 55200, Samsun, Turkey*
[‡]*Instituto Universitario de Matematica Pura y Aplicada*
*Universitat Politècnica de València*
*46022 Valencia, Spain*
[§]*kumarsaraswatpk@gmail.com*

Received February 25, 2022
Accepted November 6, 2022
Published April 15, 2023

[§]Corresponding author.

## Abstract

In this paper, we derive a new version of L1-Predictor–Corrector (L1-PC) method by using some previously given methods (L1-PC for single delay, PC for non-delay, and decomposition algorithm) to solve multiple delay-type fractional differential equations. The Caputo fractional derivative with singular type kernel is used to establish the results. Some important remarks related to the delay term estimation and error analysis are mentioned. In order to check the accuracy and correctness of our method, we solve a neural network system with two delay parameters. A number of graphs are given to justify the role of delays as well as the accuracy of the algorithm. The given method is fully novel and reliable to solve multiple delay type fractional order systems in Caputo sense.

*Keywords*: Neural Networks; Delay-Type Mathematical Model; Caputo Fractional Derivative; L1-Predictor–Corrector Method; Graphical Simulations.



## 1. INTRODUCTION

Because of its relevance, non-classical calculus, an important area of applied mathematical science, is thriving in a variety of engineering and scientific domains.[1–3] Fractional derivatives have been successfully applied in the study of different scientific problems like human and animal epidemics,[4–7] plant epidemics,[8] ecological systems,[9,10] and psychological cases,[11] etc. Delay differential equations (DDEs) are equations in which the function's derivative at a particular time relies on the solution at a prior time. The knowledge of delay in dynamical models improves its dynamics and provides a large understanding of real-life difficulties. DDEs have been utilized in traffic simulations, control systems, lasers, neurobiology, and chemical kinetics, among other applications. Delay systems have been used to simulate various infections and epidemics,[12] immune response,[13] and medication treatment phases and therapies of infectious epidemics.[14,15] Chemostat dynamical studies,[16] tumour development,[17] circadian rhythms,[18] respiratory systems,[19] and neural networks[20] have been also linked to delays.

For solving different types of fractional order systems, we need a number of numerical methods. A big discussion on the finite difference schemes for solving ordinary fractional differential equations was proposed in Ref. 21. Recently, different types of numerical schemes have been proposed by the researchers. In Ref. 22, authors have proposed a generalized form of Predictor–Corrector (PC) method to simulate initial value problems (IVPs) for fractional order systems. A new and efficient method in the sense of generalized Caputo fractional derivative has been introduced in Ref. 23.

The methods which can be used to solve fractional IVPs cannot be utilized to simulate fractional DDEs. For solving DDEs, number of specific methods are available in the literature. Authors in Ref. 24 proposed an advanced form of the Adams–Bashforth–Moulton approach for simulating fractional DDEs. They observed some numerical computations to show the method's utility. In Ref. 25, a new approach for simulating nonlinear fractional DDEs with extensive error analysis was introduced. In Ref. 26, a group of orthonormal polynomials were used to generate a numerical solution of fractional DDEs utilizing a novel type of Chelyshkov wavelet basis. In their work, wavelet bases and their features were utilized to enlist their operational matrix of fractional integration in the Riemann–Liouville (RL) sense with delay-type operational matrix. Analysis of convergence for such types of wavelet mapping were described along with error bound. In Ref. 27, shifted Jacobi polynomials were used to solve a fractional DDEs numerically. The shifted Jacobi polynomial approach was used to determine the solution outputs by establishing operational matrices for fractional integral and differentiation in the Caputo and RL sense, respectively. Recently, a PC method for solving fractional DDEs including multiple lags has been introduced in Ref. 28. A generalized form of PC method to solve fractional DDEs in the case of generalized Caputo derivative by using a non-uniform mess points has been given in Ref. 29.

In this research work, we generalie the L1-PC method (Ref. 30) to solve fractional-order multiple-delay systems. In Ref. 30, the authors have introduced the L1-PC method for solving fractional DDEs. They discussed a detailed error analysis for the proposed scheme along with solving a number of examples. Also, they compared the scheme with other existing methods. According to their conclusions, the method was more time efficient and worked for very small values of fractional orders. But their method is only applicable for single delay systems. Then what happens with multiple-delay cases? Now, we fill this research gap by modifying their method to solve the multiple-DDEs.

This paper is formulated as follows: In Sec. 2, we recall some preliminaries. In Sec. 3, we derive our main results, where we generalize the previously given L1-PC method[30] into a new version for solving multiple-DDEs. In Sec. 4, we derive the solution of a neural network to check the efficiency of the method and correctness of outputs. At the end, we conclude our findings.

## 2. PRELIMINARIES

Firstly, we recall the following preliminaries:

**Definition 1.** A real function $f(s)$, $s > 0$ belongs to the space:

(a) $C_\eta, \eta \in \mathbb{R}$ if there exists a real number $q > \eta$, such that $f(s) = s^q f_1(s)$, $f_1 \in C[0, \infty)$. Clearly, $C_\eta \subset C_\gamma$ if $\gamma \leq \eta$.
(b) $C_\eta^m, m \in \mathbb{N} \cup \{0\}$ if $f^m \in C_\eta$.

**Definition 2 (Ref. 2).** The RL fractional integral of the function $f(t) \in C_\eta (\eta \geq -1)$ is given by

$$J^\delta f(t) = \frac{1}{\Gamma(\delta)} \int_0^t (t-s)^{\delta-1} f(s) ds,$$
$$J^0 f(t) = f(t).$$

**Definition 3 (Ref. 2).** The Caputo fractional derivative of $f \in C_{-1}^c$ is given by

$$D_t^\theta f(t) = \begin{cases} \dfrac{d^c f(t)}{dt^c} & \text{if } \theta = c \in \mathbb{N}, \\ \dfrac{1}{\Gamma(c-\theta)} \displaystyle\int_0^\zeta (t-\vartheta)^{c-\theta-1} f^{(c)}(\vartheta) d\vartheta & \\ \qquad \text{if } c-1 < \theta < c, \quad c \in \mathbb{N}. \end{cases} \tag{1}$$

## 3. NEW VERSION OF THE L1-PC METHOD

In this section, we will formulate a new version of the L1-PC method by using the previously proposed results of L1-PC single delay version[30] along with L1 algorithm,[31] multiple delay PC scheme version,[28] and decomposition method.[32]

We consider the following multiple-delay type fractional differential equation:

$$\begin{aligned} {}^C D_0^\beta F(t) &= K(t, F(t), F(t-\tau_1), \ldots, F(t-\tau_k)) \\ & \qquad t \in [0, T], \quad k \in \mathbb{N}, \\ F(t) &= \phi(t), \quad t \in [-\tau, 0], \\ \tau &= \max\{\tau_1, \tau_2, \ldots, \tau_k\}, \end{aligned} \tag{2}$$

where $F(t) = (f_1, f_2, \ldots, f_N)$, $\phi(t) = (\phi_1, \phi_2, \ldots, \phi_N)$, $N \in \mathbb{N}$, $T > 0$, $0 < \beta \leq 1$ and $\tau_j \geq 0$ for all $j = 1, \ldots, k$ denote the delay coefficients.

Also, we assume that $K \in C([0,T] \times \mathbb{R}^{kN+N}, \mathbb{R}^N)$ and consider a uniform grid $\{t_i = ih, i = -m_j, -m_j+1, \ldots, -1, 0, 1, \ldots, N'\}$, where $N'$ and $m_j$ are integers such that $N' = [\frac{T}{h}]$ and $m_j = [\frac{\tau_j}{h}]$ for $j = 1, \ldots, k$.

Now, we use the L1 algorithm (see Ref. 31) for obtaining the numerical derivation of the fractional operator of order $0 < \beta \leq 1$ which is derived as follows:

$$\begin{aligned} &[{}^C D_0^\beta F(t)]_{t=t_i} \\ &\quad = \frac{1}{\Gamma(1-\beta)} \int_0^{t_i} (t_i - s)^{-\beta} F'(s) ds \\ &\quad = \frac{1}{\Gamma(1-\beta)} \sum_{n=0}^{i-1} \int_{t_n}^{t_{n+1}} (t_i - s)^{-\beta} F'(s) ds \\ &\quad \approx \frac{1}{\Gamma(1-\beta)} \sum_{n=0}^{i-1} \int_{t_n}^{t_{n+1}} (t_i - s)^{-\beta} \\ &\qquad \times \frac{F(t_{n+1}) - F(t_n)}{h} ds \\ &\quad = \sum_{n=0}^{i-1} b_{i-n-1} (F(t_{n+1}) - F(t_n)), \end{aligned} \tag{3}$$

where

$$b_{i-n-1} = \frac{h^{-\beta}}{\Gamma(2-\beta)} [(i-n)^{1-\beta} - (i-n-1)^{1-\beta}]. \tag{4}$$

Now, we assume that we have previously calculated the approximations $F(t_q)$, $(q = -m_j, -m_j +$

$1, \dots, -1, 0, 1, \dots, i-1$) and want to find the further $i$-th approximation $F(t_i)$. Thus, we approximate ${}^C D_0^\beta F(t)$ by Eq. (3) and using Eq. (2), we get

$$\begin{aligned}[{}^C D_0^\beta F(t)]_{t=t_i} &= \sum_{n=0}^{i-1} b_{i-n-1}(F_{n+1} - F_n) \\ &= K(t_i, F_i, F(t_i - \tau_1), \dots, \\ &\quad \times F(t_i - \tau_j)),\end{aligned} \tag{5}$$

where $F_n$ denotes the approximate solution of Eq. (2) at $t = t_n$. Further Eq. (5) can be written as

$$\begin{aligned} &b_{i-1}(F_1 - F_0) + b_{i-2}(F_2 - F_1) + \cdots \\ &\quad + b_0(F_i - F_{i-1}) \\ &= K(t_i, F_i, F(t_i - \tau_1), \dots, F(t_i - \tau_j)).\end{aligned} \tag{6}$$

After rewriting the terms, Eq. (6) can be displayed as follows:

$$\begin{aligned} b_0 F_i = b_0 F_{i-1} - \sum_{n=0}^{i-2} b_{n+1} F_{i-1-n} + \sum_{n=1}^{i-1} b_n F_{i-1-n} \\ + K(t_i, F_i, F(t_i - \tau_1), \dots, F(t_i - \tau_j)).\end{aligned} \tag{7}$$

Using Eq. (4), we get

$$\begin{aligned} F_i &= (i^{1-\beta} - (i-1)^{1-\beta}) F_0 \\ &\quad + \sum_{n=1}^{i-1} [2(i-n)^{1-\beta} - (i-n+1)^{1-\beta} \\ &\quad - (i-n-1)^{1-\beta}] F_n + h^\beta \Gamma(2-\beta) \\ &\quad \times K(t_i, F_i, F(t_i - \tau_1), \dots, F(t_i - \tau_j)).\end{aligned}$$

Consequently, Eq. (3) can be formulated as

$$\begin{aligned} F_i &= a_{i-1} F_0 + \sum_{n=1}^{i-1} (a_{i-n-1} - a_{i-n}) F_n \\ &\quad + h^\beta \Gamma(2-\beta) \\ &\quad \times K(t_i, F_i, F(t_i - \tau_1), \dots, F(t_i - \tau_j)),\end{aligned}$$

where $a_n := (n+1)^{1-\beta} - n^{1-\beta}$. Here, given $a_n$ have the following characteristics:

- $a_n > 0$, $n = 0, 1, \dots, i-1$
- $a_0 = 1 > a_1 > \cdots > a_n$ and $a_n \to 0$ as $n \to \infty$.
- $\sum_{n=0}^{i-1}(a_n - a_{n+1}) + a_i = (1 - a_1) + \sum_{n=1}^{i-2}(a_n - a_{n+1}) + a_{i-1} = 1$.

We identify that Eq. (8) has the form of Eq. (3) of Ref. 30, that is, it labels into the structure.

$F = g + N(F)$ where $N(F)$ is a nonlinear operator defined on a Banach space $B$. In our case:

$$g = a_{i-1} F_0 + \sum_{n=1}^{i-1} (a_{i-n-1} - a_{i-n}) F_n$$

and $N(F_i) = h^\beta \Gamma(2-\beta) K(t_i, F_i, F(t_i - \tau_1), \dots, F(t_i - \tau_j))$. Then we can apply the DJ scheme[32] (see for instance Refs. 33–35) to find the approximate solution. The DJ scheme gives the approximate value of $F_i$ as follows:

$$\begin{cases} F_{i,0} = g = a_{i-1} F_0 + \sum_{n=1}^{i-1} (a_{i-n-1} - a_{i-n}) F_n, \\ F_{i,1} = N(F_{i,0}) = h^\beta \Gamma(2-\beta) \\ \qquad \times K(t_i, F_i, F(t_i - \tau_1), \dots, F(t_i - \tau_j)), \\ F_{i,2} = N(F_{i,0} + F_{i,1}) - N(F_{i,0}). \end{cases} \tag{8}$$

The three-term approximation of the function is $F_i \approx F_{i,0} + F_{i,1} + F_{i,2}$. The delay term is given by

$$\begin{cases} F(t_q - \tau_j) = F(qh - m_j h) = F((q - m_j)h) \\ \qquad = F(t_{q-m_j}), \quad q = 0, 1, \dots, N', \\ \qquad\qquad j = 1, \dots, k, \\ F(t_q) = \phi(t_q), \quad q = -m_j, -m_j + 1, \dots, 0. \end{cases} \tag{9}$$

Equations (8) and (9) provide a new version of L1-PCM for solving multiple-delay type fractional differential equations:

$$\begin{cases} F_i^p = a_{i-1} F_0 + \sum_{n=1}^{i-1} (a_{i-n-1} - a_{i-n}) F_n, \\ z_i^p = N(F_i^p) = h^\beta \Gamma(2-\beta) \\ \qquad \times K(t_i, F_i^p, F(t_{i-m_1}), \dots, F(t_{i-m_j})), \\ F_i^c = F_i^p + h^\beta \Gamma(2-\beta) \\ \qquad \times K(t_i, F_i^p + z_i^p, F(t_{i-m_1}), \dots, F(t_{i-m_j})), \end{cases} \tag{10}$$

where $F_i^p, z_i^p$ are predictors and $F_i^c$ is the corrector.

**Remark 4.** The approximation of the delayed term $F(t_q - \tau_j)$ 9 can be defined as follows. Indeed, let $(m_j - \delta_j)h = \tau_j$ with $0 < \delta_j \leq 1$. When $\delta_j = 0$,

$F(t_q - \tau_j)$ can be given by

$$F(t_q - \tau_j) = \begin{cases} F_{q-m_j} & \text{if } q > m_j, \\ \phi_q & \text{if } q \le m_j, \end{cases} \quad j = 1, \dots, k. \tag{11}$$

When $0 < \delta_j < 1$, $j = 1, \dots, k$ it cannot be calculated directly. Assume $\omega_{q+1,j}$ is the approximation of $F(t_{q+1} - \tau_j)$ for the case when $(m_j - 1)h < \tau_j < m_j h$, $j = 1, \dots, k$. When interpolating it by the two closest points, that is,

$$\omega_{q+1,j} = \delta_j F_{q-m_j+2} + (1 - \delta_j) F_{q-m_j+1} \tag{12}$$

the last expression implies the implicit of the numerical equation if $m_j > 1$ which can be directly calculated. However, when $m_j = 1$ and $\delta_j \neq 0$, that is, $\tau_j < h$, the first term in the right side of Eq. (12) is $\delta_j F_{q+1}$. Further predication is needed in this case, that is,

$$\omega_{q+1,j} = \delta_j F^p_{q+1} + (1 - \delta_j) F_q. \tag{13}$$

**Remark 5 (Error Analysis).** Comparing the proposed model version of L1-PCM method with the L1-PCM scheme provided in Ref. 30, we get the error behaves as (see Theorem 1 in Ref. 30):

$$\max_{0 \le n \le N'} |F(t_n) - F_n| \le O(h^2),$$

where $N' = [\frac{T}{h}]$ and $F_n$ defines the approximated solution at $t = t_n$ given in 10 for $0 < \beta \le 1$ and sufficiently small $h$.

## 4. EXPERIMENTAL SIMULATIONS

In this section, for checking the correctness and accuracy of our method, we solve a multiple delay neural network (MDNN) in the form of Caputo fractional derivative. The fractional MDNN involving two time delays is considered as follows[36]:

$$\begin{aligned} {}^C D_0^\beta z_1(t) &= -n_1 z_1(t - \tau_1) + w_{11} \tanh(z_1(t - \tau_1)) \\ &\quad + w_{12} \tanh(z_2(t - \tau_2)), \\ {}^C D_0^\beta z_2(t) &= -n_2 z_2(t - \tau_2) + w_{21} \tanh(z_1(t - \tau_1)) \\ &\quad + w_{22} \tanh(z_2(t - \tau_2)). \end{aligned} \tag{14}$$

Here, $z_1(t)$ and $z_2(t)$ are the state variables, $n_1,\ n_2 > 0$ are the automatic adjustment of parameters for neurons, $w_{11}, w_{12}, w_{21}, w_{22}$ are the connection weights, and $\tau_1, \tau_2$ show the time delays. Function $\tanh(\cdot)$ is taken as an activation function. ${}^C D_0^\beta$ is the Caputo derivative operator of order $\beta \in (0, 1]$.

Now by using the above proposed version of L1-PC method Eq. (10), we get the following solution of the proposed system Eq. (14):

$$\begin{cases} z^p_{1_i} = a_{i-1} z_{1_0} + \sum_{n=1}^{i-1} (a_{i-n-1} - a_{i-n}) z_{1_n}, \\ z^p_{2_i} = a_{i-1} z_{2_0} + \sum_{n=1}^{i-1} (a_{i-n-1} - a_{i-n}) z_{2_n}, \\ \nu^p_{1_i} = N(z^p_{1_i}) = h^\beta \Gamma(2 - \beta)[-n_1 z_1(t_{i-m_1}) + w_{11} \\ \qquad \times \tanh(z_1(t_{i-m_1})) + w_{12} \tanh(z_2(t_{i-m_2}))], \\ \nu^p_{2_i} = N(z^p_{2_i}) = h^\beta \Gamma(2 - \beta)[-n_2 z_2(t_{i-m_2}) \\ \qquad + w_{21} \tanh(z_1(t_{i-m_1})) \\ \qquad + w_{22} \tanh(z_2(t_{i-m_2}))], \end{cases} \tag{15}$$

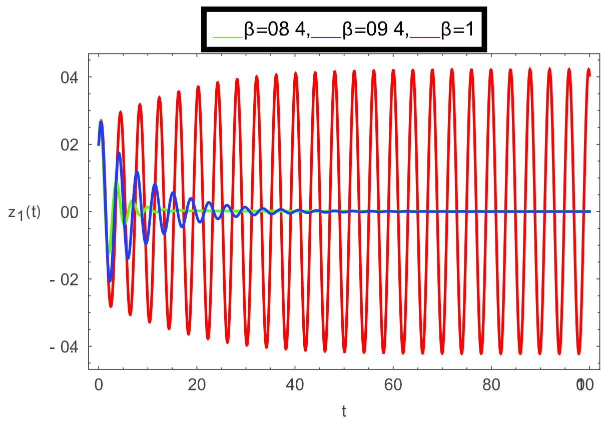
and

$$\begin{cases} z^c_{1_i} = z^p_{1_i} + h^\beta \Gamma(2 - \beta)[-n_1 z_1(t_{i-m_1}) + w_{11} \\ \qquad \times \tanh(z_1(t_{i-m_1})) + w_{12} \tanh(z_2(t_{i-m_2}))], \\ z^c_{2_i} = z^p_{2_i} + h^\beta \Gamma(2 - \beta)[-n_2 z_2(t_{i-m_2}) + w_{21} \\ \qquad \times \tanh(z_1(t_{i-m_1})) + w_{22} \tanh(z_2(t_{i-m_2}))], \end{cases} \tag{16}$$

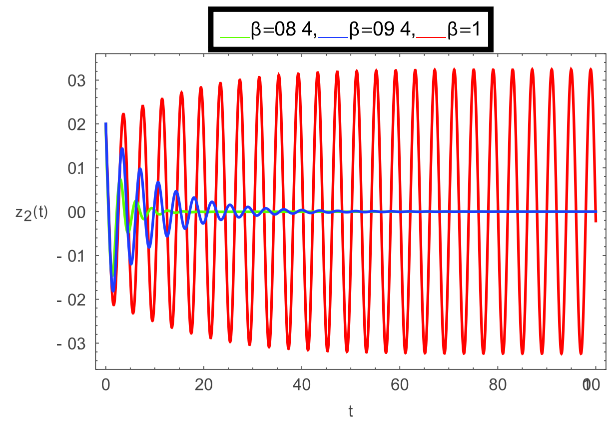
where $f^p_i, \nu^p_i$ are predictors and $f^c_i$ is the corrector. The values of the terms $z_1(t_{i-m_1})$ and $z_2(t_{i-m_2})$ are calculated as explained in Remark 4.

**Remark 6.** Here, we notice from the proposed neural network system (14) that there is no separate involvement of non-delay terms $z_1(t)$ and $z_2(t)$ in the right-hand side. In that case, there is no significance of the predictor terms $\nu^p_{1_i}$ and $\nu^p_{2_i}$ in the corrector formula (16). So, we can neglect them. Therefore, the revised solution of the given system (14) from Eqs. (15) and (16) can be written by

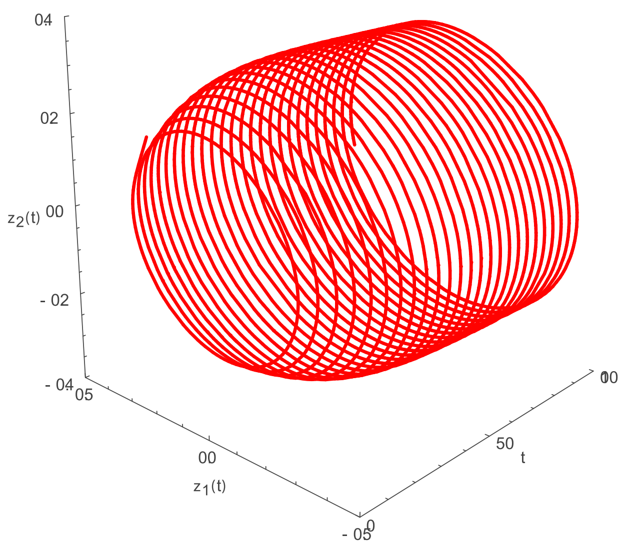
$$\begin{cases} z^p_{1_i} = a_{i-1} z_{1_0} + \sum_{n=1}^{i-1} (a_{i-n-1} - a_{i-n}) z_{1_n}, \\ z^p_{2_i} = a_{i-1} z_{2_0} + \sum_{n=1}^{i-1} (a_{i-n-1} - a_{i-n}) z_{2_n}, \\ z^c_{1_i} = z^p_{1_i} + h^\beta \Gamma(2 - \beta)[-n_1 z_1(t_{i-m_1}) \\ \qquad + w_{11} \tanh(z_1(t_{i-m_1})) \\ \qquad + w_{12} \tanh(z_2(t_{i-m_2}))], \\ z^c_{2_i} = z^p_{2_i} + h^\beta \Gamma(2 - \beta)[-n_2 z_2(t_{i-m_2}) \\ \qquad + w_{21} \tanh(z_1(t_{i-m_1})) \\ \qquad + w_{22} \tanh(z_2(t_{i-m_2}))]. \end{cases} \tag{17}$$



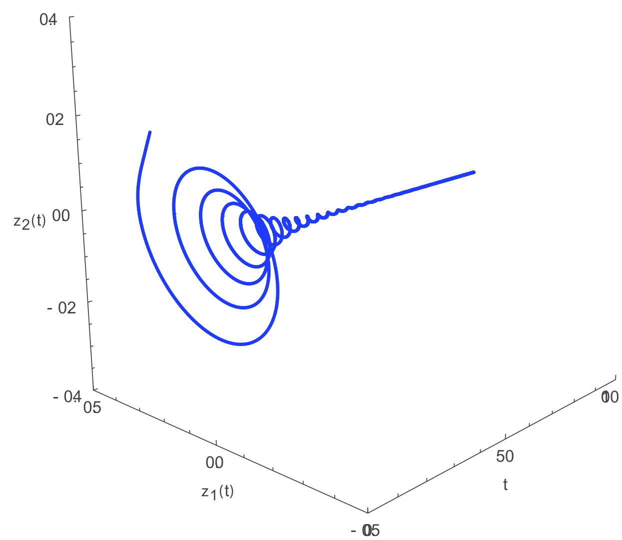

**(a)** $z_1$ versus $t$



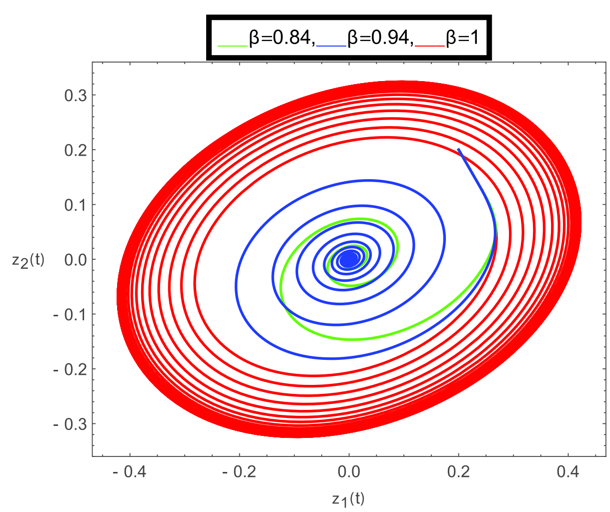

**(b)** $z_2$ versus $t$



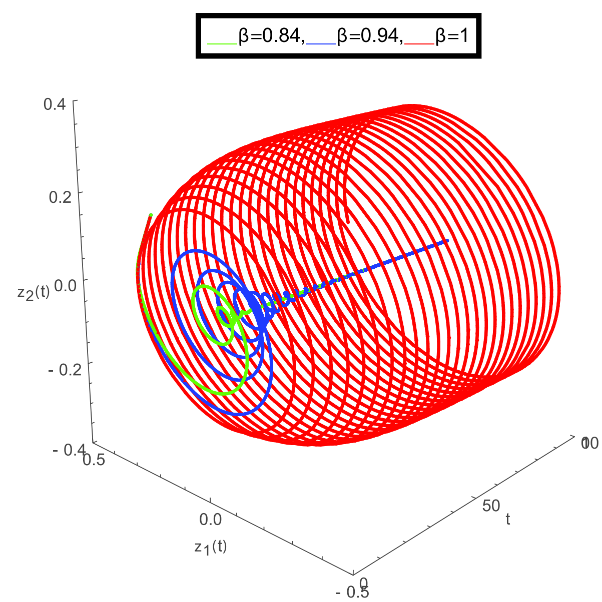

**(c)** 3D plot at $\beta = 1$




**(d)** 3D plot at $\beta = 0.94$

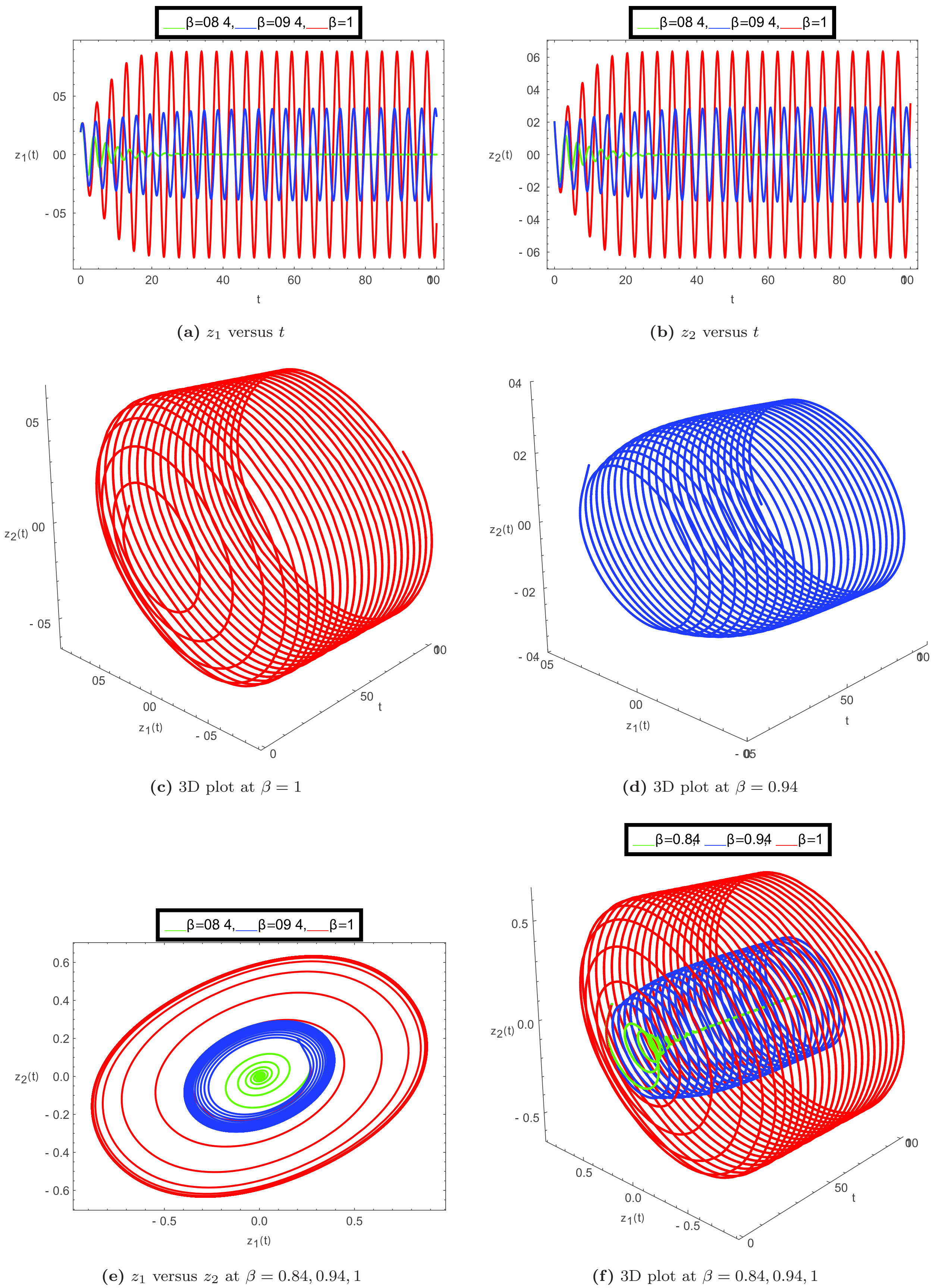



**(e)** $z_1$ versus $z_2$ at $\beta = 0.84, 0.94, 1$




**(f)** 3D plot at $\beta = 0.84, 0.94, 1$

**Fig. 1** Dynamical behavior of the system (14) for delays $\tau_1 = 0.35$ and $\tau_2 = 0.2$.




(a) $z_1$ versus $t$

(b) $z_2$ versus $t$

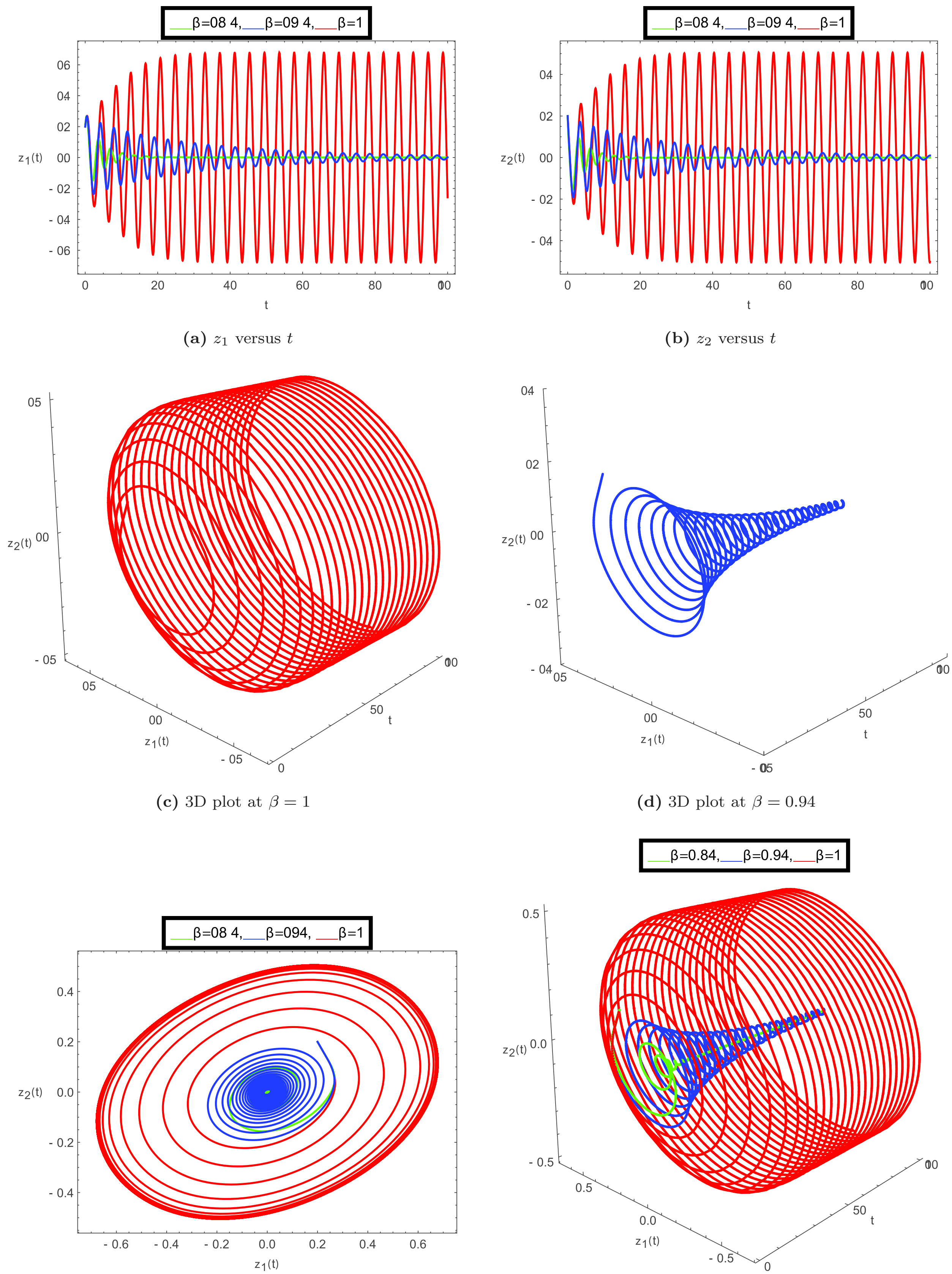
(c) 3D plot at $\beta = 1$

(d) 3D plot at $\beta = 0.94$

(e) $z_1$ versus $z_2$ at $\beta = 0.84, 0.94, 1$

(f) 3D plot at $\beta = 0.84, 0.94, 1$

**Fig. 2** Dynamical behavior of the system (14) for delays $\tau_1 = 0.45$ and $\tau_2 = 0.2$.




**(a)** $z_1$ versus $t$

**(b)** $z_2$ versus $t$

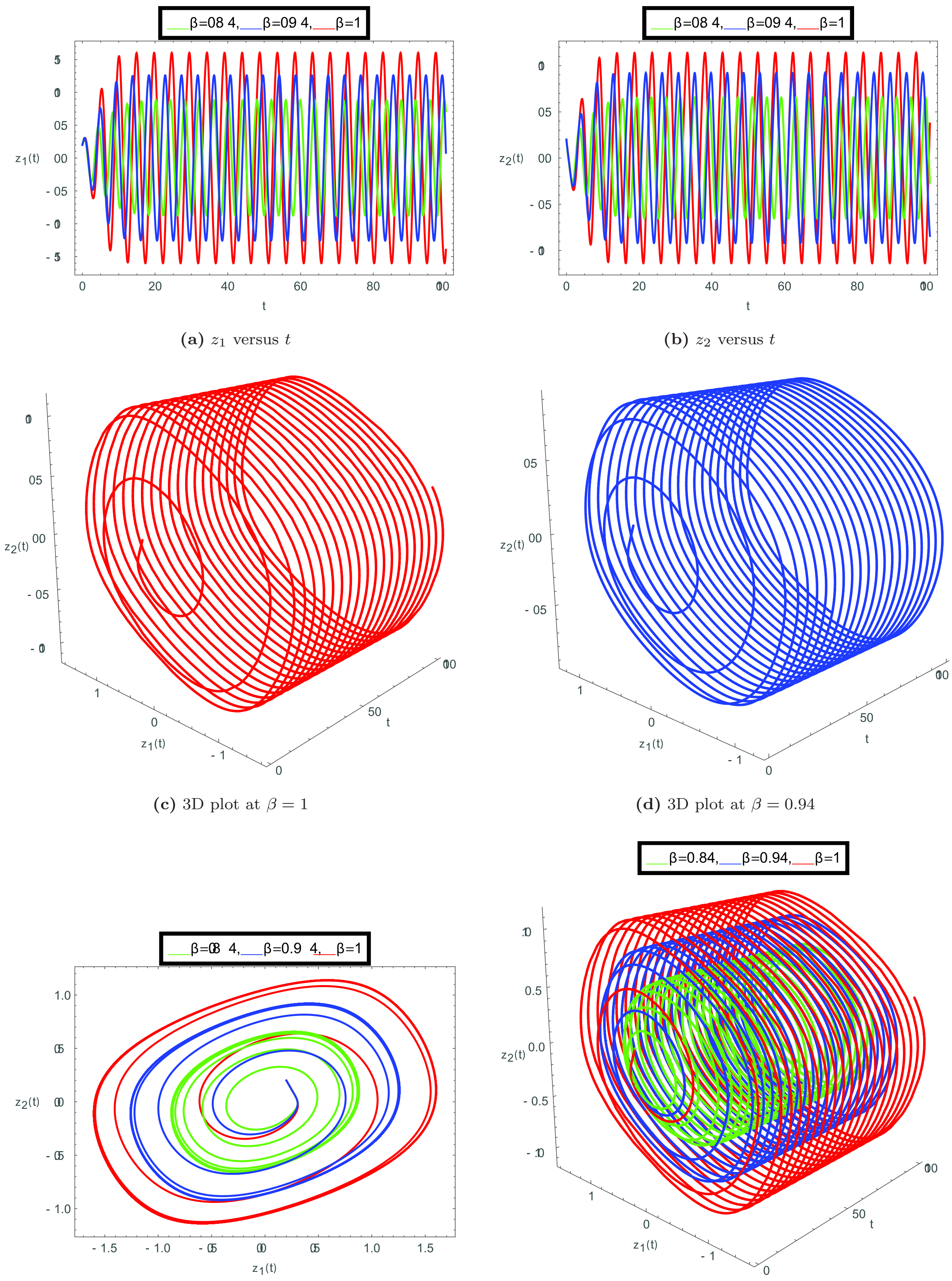
**(c)** 3D plot at $\beta = 1$

**(d)** 3D plot at $\beta = 0.94$

**(e)** $z_1$ versus $z_2$ at $\beta = 0.84, 0.94, 1$

**(f)** 3D plot at $\beta = 0.84, 0.94, 1$

**Fig. 3** Dynamical behavior of the system (14) for delays $\tau_1 = 0.40$ and $\tau_2 = 0.2$.




**(a)** $z_1$ versus $t$

**(b)** $z_2$ versus $t$

**(c)** 3D plot at $\beta = 1$

**(d)** 3D plot at $\beta = 0.94$

**(e)** $z_1$ versus $z_2$ at $\beta = 0.84, 0.94, 1$

**(f)** 3D plot at $\beta = 0.84, 0.94, 1$

**Fig. 4** Dynamical behavior of the system (14) for delays $\tau_1 = 0.56$ and $\tau_2 = 0.4$.

In order to explore the graphical solutions of the given system (14) by using the algorithm (17), we consider the following numerical values of the proposed parameters: $n_1 = n_2 = 1,\ w_{11} = 0.2, w_{12} = 1.8, w_{21} = -1.2, w_{22} = 0.5$. The initial constraints are fixed as $z_1(0) = z_2(0) = 0.2$. Now, we perform our simulations for the fractional order values $\beta = 1, 0.94, 0.84$. For simulating the effects of delay parameters $\tau_1$ and $\tau_2$, we use various cases of delay values. We use *Mathematica* software to code the given algorithm.

In the first case, we take $\tau_1 = 0.35$ and $\tau_2 = 0.2$. For better accuracy in the outputs, we choose the step size $h = 0.025$. Then, $m_1 = [\frac{\tau_1}{h}] = [\frac{0.35}{0.025}] = 14$ and $m_2 = [\frac{\tau_2}{h}] = [\frac{0.2}{0.025}] = 8$. The graphical outputs for this case are given in the company of Figs. 1a–1f. From Figs. 1a and 1b, we notice that at fractional order $\beta = 1$, the oscillations are constant after some time but not converging to any fixed value, but when we decrease the fractional order then after $\beta = 0.94$, the oscillations converge to the origin which justifies the correctness of our outputs compared to the case of $\beta = 0.94$ given in Ref. 36. Three-dimensional (3D) trajectories of the given system at $\beta = 1$ and $\beta = 0.94$ are plotted in Figs. 1c and 1d, simultaneously. Plots of Figs. 1e and 1f define the 2D and 3D graphs of the given classes $z_1$ and $z_2$ at all given fractional order values $\beta$, respectively.

In the second case, we take $\tau_1 = 0.45$ and $\tau_2 = 0.2$. Again, we consider the same step size $h = 0.025$. Then, $m_1 = [\frac{\tau_1}{h}] = [\frac{0.45}{0.025}] = 18$ and $m_2 = [\frac{\tau_2}{h}] = [\frac{0.2}{0.025}] = 8$. The graphical outputs for this case are given in the group of Figs. 2a–2f. From Figs. 2a and 2b, we notice that at both fractional orders $\beta = 1$ and $\beta = 0.94$, the oscillations are constant after some time but not converging to any fixed value, but when we decrease the fractional order then after $\beta = 0.84$, the oscillations converge to the origin which again shows the correctness of our outputs comparing with the outputs of Ref. 36. 3D trajectories of the given system at $\beta = 1$ and $\beta = 0.94$ are plotted in Figs. 2c and 2d, simultaneously. Plots of Figs. 2e and 2f show the 2D and 3D graphs of the given classes $z_1$ and $z_2$ at all given fractional order values of $\beta$, respectively.

In our next case, we take $\tau_1 = 0.40$ and $\tau_2 = 0.2$ with the same step size $h = 0.025$. Then, $m_1 = [\frac{\tau_1}{h}] = [\frac{0.40}{0.025}] = 16$ and $m_2 = [\frac{\tau_2}{h}] = [\frac{0.2}{0.025}] = 8$. The graphical outputs for this case are given in the group of Fig. 3 (Figs. 3a–3f). From Figs. 3a and 3b, we notice that at fractional order $\beta = 0.94$, the oscillations start to converge but less slowly as compared to the first case (Figs. 1a and 1b). 3D trajectories of the given system at $\beta = 1$ and $\beta = 0.94$ are plotted in graphics Figs. 3c and 3d, respectively. Plots Figs. 3e and 3f show the 2D and 3D graphs of the given classes $z_1$ and $z_2$ at all given fractional order values $\beta$, respectively.

In our last case, we fix $\tau_1 = 0.56$ and $\tau_2 = 0.4$ with the same step size $h = 0.025$. Then, $m_1 = [\frac{\tau_1}{h}] = [\frac{0.56}{0.025}] = 22$ and $m_2 = [\frac{\tau_2}{h}] = [\frac{0.4}{0.025}] = 16$. The graphical outputs for this case are given in the group of Figs. 4a–4f. From Figs. 4a and 4b, we notice that at all considered fractional orders $\beta = 1, 0.94, 0.84$, the oscillations do not converge to any particular value which was also mentioned in Ref. 36. 3D trajectories of the given system at $\beta = 1$ and $\beta = 0.94$ are plotted in Figs. 4c and 4d, simultaneously. Plots of Figs. 4e and 4f show the 2D and 3D graphs of the given classes $z_1$ and $z_2$ at all given fractional order values $\beta$, respectively.

From the above given graphical observations, we can say that the proposed method works well and it is suitable for simulating the given neural network system. The asymptotic steadiness and the volatility of the origin of the given system (14), which were previously studied in Ref. 36, can be investigated much clearly considering the proposed method.

To justify the correctness of the proposed method, we compare our outputs with the outputs obtained from the PC method given in Ref. 28. In this regard, the comparison of numerical solutions

**Table 1** Numerical Results for $z_1(t)$ at $\tau_1 = 0.35$ and $\tau_2 = 0.2$.

| $t$ | $\beta = 0.84$ Ref. 28 | $\beta = 0.84$ DJ Scheme | $\beta = 0.94$ Ref. 28 | $\beta = 0.94$ DJ Scheme | $\beta = 1$ Ref. 28 | $\beta = 1$ DJ Scheme |
|---|---|---|---|---|---|---|
| 1 | 0.135509 | 0.115926 | 0.178629 | 0.155844 | 0.201393 | 0.182729 |
| 10 | 0.028441 | 0.013452 | −0.204470 | −0.076149 | −0.290946 | −0.274684 |
| 50 | 0.001154 | 0.001149 | 0.042533 | −0.001452 | −0.487985 | −0.413402 |
| 100 | 0.000644 | 0.000644 | 0.056202 | 0.000153 | 0.149454 | 0.405077 |

**Table 2** **Numerical Results for $z_2(t)$ at $\tau_1 = 0.35$ and $\tau_2 = 0.2$.**

| $t$ | $\beta = 0.84$ Ref. 28 | $\beta = 0.84$ DJ Scheme | $\beta = 0.94$ Ref. 28 | $\beta = 0.94$ DJ Scheme | $\beta = 1$ Ref. 28 | $\beta = 1$ DJ Scheme |
|---|---|---|---|---|---|---|
| 1 | −0.127472 | −0.122511 | −0.121820 | −0.121740 | −0.116033 | −0.120500 |
| 10 | 0.017759 | 0.002526 | 0.009148 | 0.024957 | −0.331580 | −0.177418 |
| 50 | −0.000231 | −0.000230 | −0.066069 | −0.000291 | −0.231437 | −0.104650 |
| 100 | −0.000122 | −0.000122 | −0.003579 | −0.000033 | −0.494628 | −0.021466 |

**Table 3** **Numerical Results for $z_1(t)$ at $\tau_1 = 0.45$ and $\tau_2 = 0.2$.**

| $t$ | $\beta = 0.84$ Ref. 28 | $\beta = 0.84$ DJ Scheme | $\beta = 0.94$ Ref. 28 | $\beta = 0.94$ DJ Scheme | $\beta = 1$ Ref. 28 | $\beta = 1$ DJ Scheme |
|---|---|---|---|---|---|---|
| 1 | 0.144854 | 0.123629 | 0.187299 | 0.164025 | 0.209369 | 0.190825 |
| 10 | 0.029705 | 0.047832 | −0.473395 | −0.307563 | −0.055475 | −0.297652 |
| 50 | 0.023563 | 0.001378 | 0.690455 | 0.389423 | 1.023091 | 0.273100 |
| 100 | 0.004446 | 0.000644 | 0.698661 | 0.330610 | −1.030329 | −0.594233 |

**Table 4** **Numerical Results for $z_2(t)$ at $\tau_1 = 0.45$ and $\tau_2 = 0.2$.**

| $t$ | $\beta = 0.84$ Ref. 28 | $\beta = 0.84$ DJ Scheme | $\beta = 0.94$ Ref. 28 | $\beta = 0.94$ DJ Scheme | $\beta = 1$ Ref. 28 | $\beta = 1$ DJ Scheme |
|---|---|---|---|---|---|---|
| 1 | −0.122945 | −0.121958 | −0.114695 | −0.117323 | −0.108409 | −0.114204 |
| 10 | 0.109780 | 0.043989 | −0.260330 | −0.048859 | −0.602581 | −0.501341 |
| 50 | 0.016010 | 0.000026 | 0.215599 | 0.082937 | 0.252413 | 0.634137 |
| 100 | −0.001242 | −0.000121 | −0.094733 | −0.078154 | 0.152623 | 0.310413 |

**Table 5** **CPU Time in Seconds for $z_1(t)$ at $\tau_1 = 0.35$ and $\tau_2 = 0.2$.**

| $t$ | $\beta = 0.84$ | $\beta = 0.94$ | $\beta = 1$ |
|---|---|---|---|
| 1 | 0.015625 | 0.015625 | 0.015622 |
| 10 | 0.484375 | 0.453125 | 0.031280 |
| 50 | 11.218750 | 11.312500 | 0.062513 |
| 100 | 45.500000 | 46.484375 | 0.109379 |

**Table 6** **CPU Time in Seconds for $z_2(t)$ at $\tau_1 = 0.35$ and $\tau_2 = 0.2$.**

| $t$ | $\beta = 0.84$ | $\beta = 0.94$ | $\beta = 1$ |
|---|---|---|---|
| 1 | 0.031250 | 0.015625 | 0.015591 |
| 10 | 0.468750 | 0.468750 | 0.031270 |
| 50 | 11.296875 | 11.218750 | 0.062514 |
| 100 | 45.906250 | 47.234375 | 0.093723 |

**Table 7** **CPU Time in Seconds for $z_1(t)$ at $\tau_1 = 0.45$ and $\tau_2 = 0.2$.**

| $t$ | $\beta = 0.84$ | $\beta = 0.94$ | $\beta = 1$ |
|---|---|---|---|
| 1 | 0.078087 | 0.015648 | 0.015644 |
| 10 | 0.453047 | 0.484240 | 0.031271 |
| 50 | 11.138056 | 11.169268 | 0.078141 |
| 100 | 44.989515 | 45.520625 | 0.140595 |

**Table 8** **CPU Time in Seconds for $z_2(t)$ at $\tau_1 = 0.45$ and $\tau_2 = 0.2$.**

| $t$ | $\beta = 0.84$ | $\beta = 0.94$ | $\beta = 1$ |
|---|---|---|---|
| 1 | 0.015626 | 0.015648 | 0.015622 |
| 10 | 0.484294 | 0.499912 | 0.031257 |
| 50 | 11.263027 | 11.153678 | 0.078092 |
| 100 | 46.801602 | 46.628350 | 0.125000 |

of $z_1$ and $z_2$ at various delay and fractional-order values by using PC method (Ref. 28) and proposed scheme (DJ scheme) is given in Tables 1–4. Here, we see that both outputs are similar.

The CPU processing times for the numerical solutions of $z_1$ and $z_2$ at various delay and fractional-order values are given in Tables 5–8.

## 5. CONCLUSION

In this work, we have derived a new form of the L1-PC method by using some previously proposed methods to solve multiple delay-type fractional differential equations. We have considered the Caputo fractional derivative with singular type kernel to establish the results. Some important remarks related to the delay term estimation and error analysis have been given. For justifying the accuracy and correctness of our method, we have given the solution of a neural network system with two delay parameters. A number of graphs are plotted to justify the role of delays as well as the accuracy of the algorithm. Overall, we conclude that the proposed method is fully reliable and accurate to solve multiple delay type fractional order systems. The *Mathematica* software was used to code the given algorithm. In future, some novel attempts to prove the stability and convergence of the proposed new scheme can be given to increase the popularity and strength of the scheme. Also, in future research the proposed method can be formulated in the sense of non-singular kernels by using Caputo–Fabrizio, Atangana–Baleanu or any other type of fractional derivatives.

## ACKNOWLEDGMENT

M. Murillo-Arcila is supported by MCIN/AEI/10.13039/501100011033, Project PID2019-105011GBI00 and by Generalitat Valenciana, Project PROMETEU/2021/070.

## REFERENCES

1. A. Kilbas, H. M. Srivastava and J. J. Trujillo, *Theory and Applications of Fractional Differential Equations* (Elsevier Science, 2006).
2. I. Podlubny, *Fractional Differential Equations: An Introduction to Fractional Derivatives, Fractional Differential Equations, to Methods of their Solution and Some of their Applications* (Elsevier, 1998).
3. M. Caputo and M. Fabrizio, A new definition of fractional derivative without singular kernel, *Progr. Fract. Differ. Appl.* **1**(2) (2015) 1–13.
4. V. S. Erturk and P. Kumar, Solution of a COVID-19 model via new generalized Caputo-type fractional derivatives, *Chaos Solitons Fractals* **139** (2020) 110280.
5. H. Abboubakar, P. Kumar, V. S. Erturk and A. Kumar, A mathematical study of a Tuberculosis model with fractional derivatives, *Int. J. Model. Simul. Sci. Comput.* **12**(4) (2021) 2150037.
6. H. Abboubakar, P. Kumar, N. A. Rangaig and S. Kumar, A Malaria model with Caputo–Fabrizio and Atangana–Baleanu derivatives, *Int. J. Model. Simul. Sci. Comput.* **12**(2) (2020) 2150013.
7. P. Kumar, V. S. Erturk, A. Yusuf, K. S. Nisar and S. F. Abdelwahab, A study on canine distemper virus (CDV) and rabies epidemics in the red fox population via fractional derivatives, *Results Phys.* **25** (2021) 104281.
8. P. Kumar, V. S. Erturk and H. Almusawa, Mathematical structure of mosaic disease using microbial biostimulants via Caputo and Atangana–Baleanu derivatives, *Results Phys.* **24** (2021) 104186.
9. P. Kumar, V. S. Erturk, R. Banerjee, M. Yavuz and V. Govindaraj, Fractional modeling of plankton-oxygen dynamics under climate change by the application of a recent numerical algorithm, *Phys. Scr.* **96**(12) (2021) 124044.
10. P. Kumar and V. S. Erturk, Environmental persistence influences infection dynamics for a butterfly pathogen via new generalised Caputo type fractional derivative, *Chaos Solitons Fractals* **144** (2021) 110672.
11. P. Kumar, V. S. Erturk and M. Murillo-Arcila, A complex fractional mathematical modeling for the love story of Layla and Majnun, *Chaos Solitons Fractals* **150** (2021) 111091.
12. S. Ciupe, B. De Bivort, D. Bortz and P. Nelson, Estimates of kinetic parameters from HIV patient data during primary infection through the eyes of three different models, *Math. Biosci.* **200**(1) (2006) 1–27.
13. K. Cooke, Y. Kuang and B. Li, Analyses of an antiviral immune response model with time delays, *Canad. Appl. Math. Quart.* **6**(4) (1998) 321–354.
14. P. W. Nelson, J. D. Murray and A. S. Perelson, A model of HIV-1 pathogenesis that includes an intracellular delay, *Math. Biosci.* **163**(2) (2000) 201–215.
15. P. Kumar, V. S. Erturk, A. Yusuf and S. Kumar, Fractional time-delay mathematical modeling of Oncolytic Virotherapy, *Chaos Solitons Fractals* **150** (2021) 111123.
16. T. Zhao, Global periodic-solutions for a differential delay system modeling a microbial population in the

chemostat, *J. Math. Anal. Appl.* **193**(1) (1995) 329–352.

17. M. Villasana and A. Radunskaya, A delay differential equation model for tumor growth, *J. Math. Biol.* **47**(3) (2003) 270–294.
18. P. Smolen, D. A. Baxter and J. H. Byrne, A reduced model clarifies the role of feedback loops and time delays in the drosophila circadian oscillator, *Biophys. J.* **83**(5) (2002) 2349–2359.
19. B. Vielle and G. Chauvet, Delay equation analysis of human respiratory stability, *Math. Biosci.* **152**(2) (1998) 105–122.
20. S. A. Campbell, R. Edwards and P. van den Driessche, Delayed coupling between two neural network loops, *SIAM J. Appl. Math.* **65**(1) (2004) 316–335.
21. C. Li and F. Zeng, The finite difference methods for fractional ordinary differential equations, *Numer. Funct. Anal. Optim.* **34**(2) (2013) 149–179.
22. Z. Odibat and D. Baleanu, Numerical simulation of initial value problems with generalized Caputo-type fractional derivatives, *Appl. Numer. Math.* **156** (2020) 94–105.
23. P. Kumar, V. S. Erturk and A. Kumar, A new technique to solve generalized Caputo type fractional differential equations with the example of computer virus model, *J. Math. Ext.* **15** (2021) 1–23.
24. S. Bhalekar and V. Daftardar-Gejji, A predictor–corrector scheme for solving nonlinear delay differential equations of fractional order, *J. Fract. Calc. Appl.* **1**(5) (2011) 1–9.
25. V. Daftardar-Gejji, Y. Sukale and S. Bhalekar, Solving fractional delay differential equations: A new approach, *Fract. Calc. Appl. Anal.* **18**(2) (2015) 400–418.
26. F. Mohammadi, Numerical solution of systems of fractional delay differential equations using a new kind of wavelet basis, *Comput. Appl. Math.* **37**(4) (2018) 4122–4144.
27. P. Muthukumar and B. Ganesh Priya, Numerical solution of fractional delay differential equation by shifted Jacobi polynomials, *Int. J. Comput. Math.* **94**(3) (2017) 471–492.
28. S. Abdelmalek and R. Douaifia, A predictor-corrector method for fractional delay-differential system with multiple lags, *Commun. Nonlinear Anal.* **6**(1) (2019) 78–88.
29. Z. Odibat, V. S. Erturk, P. Kumar and V. Govindaraj, Dynamics of generalized Caputo type delay fractional differential equations using a modified Predictor-Corrector scheme, *Phys. Scr.* **96**(12) (2021) 125213.
30. A. Jhinga and V. Daftardar-Gejji, A new numerical method for solving fractional delay differential equations, *Comput. Appl. Math.* **38**(4) (2019) 1–18.
31. K. Oldham and J. Spanier, *The Fractional Calculus Theory and Applications of Differentiation and Integration to Arbitrary Order* (Elsevier, 1974).
32. V. Daftardar-Gejji and H. Jafari, An iterative method for solving nonlinear functional equations, *J. Math. Anal. Appl.* **316**(2) (2006) 753–763.
33. M. A. AL-Jawary, G. H. Radhi and J. Ravnik, Daftardar–Jafari method for solving nonlinear thin film flow problem, *Arab J. Basic Appl. Sci.* **25**(1) (2018) 20–27.
34. H. G. Taher, H. Ahmad, J. Singh, D. Kumar and H. K. Jassim, Solving fractional PDEs by using Daftardar–Jafari method, *AIP Conf. Proc.* **2386**(1) (2022) 060002.
35. I. Ullah, M. T. Rahim and H. Khan, Application of Daftardar–Jafari method to first grade MHD squeezing fluid flow in a porous medium with slip boundary condition, in *Abstract and Applied Analysis*, Vol. 2014 (Hindawi, 2014).
36. C. Huang, H. Liu, X. Shi, X. Chen, M. Xiao, Z. Wang and J. Cao, Bifurcations in a fractional-order neural network with multiple leakage delays, *Neural Netw.* **131** (2020) 115–126.