AIMS Mathematics

http://www.aimspress.com/journal/Math

AIMS Mathematics, 7(4): 5871–5894.
DOI: 10.3934/math.2022326
Received: 03 October 2021
Revised: 24 December 2021
Accepted: 31 December 2021
Published: 12 January 2022

*Research article*

# Poincaré maps for detecting chaos in fractional-order systems with hidden attractors for its Kaplan-Yorke dimension optimization

**Daniel Clemente-López[1], Esteban Tlelo-Cuautle[1,*], Luis-Gerardo de la Fraga[2], José de Jesús Rangel-Magdaleno[1] and Jesus Manuel Munoz-Pacheco[3]**

[1] Department of Electronics, Instituto Nacional de Astrofísica, Optica y Electrónica (INAOE), Luis Enrique Erro No. 1, Tonantzintla, Puebla 72840, Mexico

[2] Computer Science Department, Cinvestav, Av. IPN 2508, Mexico City 07360, Mexico

[3] Facultad de Ciencias de la Electrónica, Benemérita Universidad Autónoma de Puebla, Ciudad Universitaria, 18 Sur y Avenida San Claudio San Manuel, Puebla 72592, Mexico

* **Correspondence:** Email: etlelo@inaoep.mx; Tel: +522222663100.

**Abstract:** The optimization of fractional-order (FO) chaotic systems is challenging when simulating a considerable number of cases for long times, where the primary problem is verifying if the given parameter values will generate chaotic behavior. In this manner, we introduce a methodology for detecting chaotic behavior in FO systems through the analysis of Poincaré maps. The optimization process is performed applying differential evolution (DE) and accelerated particle swarm optimization (APSO) algorithms for maximizing the Kaplan-Yorke dimension ($D_{KY}$) of two case studies: a 3D and a 4D FO chaotic systems with hidden attractors. These FO chaotic systems are solved applying the Grünwald-Letnikov method, and the Numba just-in-time (jit) compiler is used to improve the optimization process's time execution in Python programming language. The optimization results show that the proposed method efficiently optimizes FO chaotic systems with hidden attractors while saving execution time.

**Keywords:** chaos; fractional calculus; Poincaré map; differential evolution algorithm; accelerated particle swarm optimization; Kaplan-Yorke dimension
**Mathematics Subject Classification:** 26A33, 90C29

## 1. Introduction

Chaos theory is the branch of mathematics that deals with complex behavior emerging in dynamical systems [1]. Chaotic systems possess special features such as extreme sensitivity to initial conditions, random-like unpredictable behavior and ergodicity. Improving these characteristics enhance applications in almost every field of science and engineering, such as: Biology [2], astrophysics [3], mechanics [4], economics [5], secure communications [6], cryptography [7], robotics [8], control [9] and so on. With the study of fractional calculus, it has been found that Fractional-Order (FO) derivatives serve as an extra-degree freedom for enhancing chaos-based applications, this thanks to the FO derivatives that have inheritance or memory properties that preserve complex phenomena with more accuracy [10, 11]. For instance, in cryptography and secure communications, FO chaotic systems can generate larger keyspace than their integer counterpart [12, 13].

According to the literature, attractors of dynamical systems can be divided into two categories depending on the localization of their equilibrium points in the corresponding basin of attraction [14]. On the one hand, chaotic systems with self-excited attractors are those whose basin of attraction intersects with an unstable equilibrium. On the other hand, chaotic systems with hidden attractors have a basin of attraction that does not intersect with any open neighborhoods of equilibria. Since the first hidden attractor was found in the classical Chua's circuit [15], much more attention has been shifted towards this new type of attractors. Firstly, the dynamics of hidden attractors are highly complex to analyze since they can lead to unexpected behavior. For instance, complex phenomena such as "Multistability" [16], which refers to when two or more different attractors coexist under the same parameters are likely to appear in dynamical systems with hidden attractors. The multistability phenomena depend on the initial conditions to switch between one attractor to a totally different one, and this property can be used as an additional source of randomness for potential applications. For instance, but not limited to, they might enhance the performance on cryptography [17], and secure communications [18].

There exist few studies related to FO chaotic systems with hidden attractors, but FO derivatives can generate several families of hidden chaotic attractors in their commensurate and incommensurate models where multistability phenomena can also be observed [19, 20]. Therefore, the research effort oriented to FO chaotic systems with hidden attractors is an area of opportunity to enhance chaos-based applications. One of the critical factors that guarantees the success of those applications, is the complexity degree of the chaotic attractor that can be quantified by its Kaplan-Yorke dimension ($D_{KY}$). In this manner, the application of metaheuristics, such as: Differential Evolution (DE) and Accelerated Particle Swarm Optimization (APSO) algorithms, has gained interest in recent years for solving optimization problems of FO chaotic systems [21]. In [22] the optimization through metaheuristics of the $D_{KY}$ is achieved on four FO chaotic systems with self-excited attractors, a similar approach is presented in [23]. To the best of the authors' knowledge, there are no works that explore the optimization of FO chaotic systems with hidden attractors.

Motivated by the above discussion, this paper aims to introduce the optimization of FO chaotic systems with hidden attractors. One of the challenges in the optimization process, is that hidden attractors have small basin of attraction [24–26], and thus, tiny variations in the parameters, and fractional-order derivatives or initial conditions can easily shift the chaotic behavior to a periodic dynamics. Hence, careful considerations needs to be taken to ensure that the optimization process is performing correctly. The literature reports several methods for detecting Chaos in dynamical systems, such as phase portraits, bifurcation diagrams, equilibrium points analysis through Shilnikov theorem, Lyapunov exponents (LEs), Poincaré maps, among others. It is well-known that LEs provide quantitative information over the stretching and folding of the chaotic attractors. However, accurate computation of Lyapunov exponents requires the analysis of large time series by specialized library packages such as TISEAN [27], which can be high computational expensive, but LEs are necessary for calculating the $D_{KY}$ of a chaotic attractor. For such problem, one can introduce constraints that ensures the chaotic motion for its LEs computation through TISEAN. In this manner, Poincaré maps provide a qualitative understating of the dynamical characteristics of a $n$-dimensional system into a $(n-1)$ representation [28]. This is done by the intersection analysis of a hyper-plane transverse to the flow of the dynamical system, then the Poincaré map corresponding to a chaotic system should display enough distribution over a 2-D plane. In this regard, the contribution of this work is to detect chaotic motion in FO chaotic systems with hidden attractors through the distribution quantification of Poincaré maps. Moreover, to save execution time, the optimization algorithms are implemented by using the Numba just-in-time (jit) compiler in the Python programming language.

The rest of the paper is organized as follows: Section 2 summarizes the background of this work. DE and APSO algorithms are described in Section 3. Section 4 shows the numerical analysis of the FO chaotic systems. Section 5 details the optimization process whereas the discussion of the optimization results is presented in Section 6. Finally, the authors' conclusions are given in Section 7.

## 2. Mathematical background

### 2.1. Grünwald-Letnikov method

Although one can cite various definitions of FO derivatives such as as the Grünwald-Letnikov [29], Riemman Liouville [30] and Caputo definitions [31], these mathematical interpretations do not allow a direct understanding of the derivative solution. Therefore, the solution of FO differential equations are often obtained through numerical approximations accordingly the used fractional derivative definition [32–35]. For instance, the Adams Bashforth Moulton (ABM) method is a predictor-corrector scheme highly recognized as a valuable fractional calculus tool due to the good approximation that it produces [36]. Drawbacks of this scheme are related to the complexity for representing the memory properties of FO derivatives, that requires from long time computation. The Grünwald Letnikov (GL) method is another numerical scheme very suitable for solving FO systems. It can be considered as a generalization of the Euler method for solving ordinary differential equations [37]. The GL method is of low computational cost and has approximately the same order of accuracy and good match of numerical solutions as ABM [38]. Therefore, this work uses the Grünwald-Letnikov method for solving the FO differential equations that comprises the FO chaotic systems that are the case study. The mathematical description is given below.

According to the explicit numerical approximation of the Grünwald-Letnikov derivative [39], the FO derivative $0 < q < 1$ for a discrete function $f(t_k)$, can be described by (2.1), $h$ is the integration step-size, and $C_j^q$ are binomial coefficients which are recursively calculated by (2.2).

$$D_{t_k}^{q} f(t_k) \approx h^{-q} \sum_{j=0}^{k} C_j^q f(t_{k-j}), \tag{2.1}$$

$$C_0^q = 1,\ C_j^q = \left(1 - \frac{1+q}{j}\right) C_{j-1}^q,\ j = 1, 2, ..., k. \tag{2.2}$$

The solution of a FO system in the form $D_{t_k}^{q} x(t_k) = f(x(t_k))$, can be obtained by (2.3), which ideally requires infinite memory length for numerical simulation since the summation term depends on the discretized time $t_k$.

$$x(t_k) = f(x(t_{k-1}))h^q - \sum_{j=1}^{k} C_j^q x(t_{k-j}). \tag{2.3}$$

### 2.2. *Poincaré maps*

One of the features of chaos behavior is the existence of a dense set of periodic orbits, which implies that any periodic trajectory of the orbit visits an arbitrarily small neighborhood of a non-periodic one [40]. In this manner, and since Poincaré maps preserve many of the orbits of dynamical systems, it is a viable tool for detecting chaos in both integer and FO systems [41, 42]. Below is described the elaboration of a Poincaré map.

Lets consider a FO system of the form given in (2.4), where $0 < q < 1$, $x(t) = [x_1(t), x_2(t), ..., x_n(t)]^T \in \mathbb{R}^n$ and $f(x(t)) = [f_1(x(t)), f_2(x(t)), ..., f_n(x(t))]^T$.

$$D_t^q x(t) = f(x(t)), \tag{2.4}$$

The Poincaré map of the system given in (2.4) is obtained as follows:

(1) Without loss of generality, let $\Sigma \in \mathbb{R}^{n-1}$ be the Poincaré section of system (2.4) such that $\Sigma \cap x(t) = [x_1(t), x_2(t), ..., x_{n-1}(t)]^T$.
(2) Next, the Poincaré section is settled on a given $x_n = \sigma$.
(3) Finally, all states $[x_1(t), x_2(t), ..., x_{n-1}(t)]^T$ are captured each time the dynamical flow intersects the Poincaré section.

The captured points by the above procedure portrait the Poincaré map of system (2.4), then, the behavior of the system is obtained by analyzing the distribution of the distinct points of the Poincaré map. If the Poincaré map consists neither of a small number of points or filling a continuous line but instead a large and irregular distribution, then it is a strong indicator of chaos [43].

### 2.3. *Kaplan Yorke dimension*

The Kaplan Yorke dimension $D_{KY}$ describes the fractal dimension of an attractor by means of its Lyapunov exponents as follows [44]:

$$D_{KY} = j + \frac{1}{|\lambda_{j+1}|} \sum_{k=1}^{j} \lambda_k, \tag{2.5}$$

herein, the Lyapunov exponents are sorted from highest to lowest $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{j+1}$ and $j \in \mathbb{N}$ is the largest integer for which $\lambda_1 + \lambda_2 + \cdots + \lambda_j \geq 0$. The $D_{KY}$ gives a meaningful measure to quantify the complexity of a chaotic attractor [45, 46]. Therefore, it is the interest of this optimization work to determine a set of parameters and fractional orders that leads to a higher $D_{KY}$.

## 3. DE and APSO algorithms

### 3.1. Differential Evolution algorithm

The Differential Evolution (DE) algorithm belongs to the family of algorithms based on biological evolution, which uses genetic cross and mutation operators to generate new candidate solutions [47]. The structure of DE is illustrated in the chart flow of Figure 1, whose description is as follows:

(1) In the first step, the initial population is generated within the range of interest.

$$X_i \in \mathbb{R}^n,\ i = 1, 2, ..., p, \tag{3.1}$$

(2) Next, the objective function is evaluated for each individual in the population

$$\forall X_i,\ do\ f(X_i) \in \mathbb{R}. \tag{3.2}$$

(3) In the next step, the mutation operation is carried out:

- (3.1) First, three randomly individuals $X_1$, $X_2$ and $X_3$ are selected.
- (3.2) Then, the following operation is performed:

$$w_i = X_1 + \mu(X_2 - X_3) \tag{3.3}$$

where $\mu \in \mathbb{R}$ is a scaling parameter.

(4) In the next step, individuals are crossed according to the condition

$$F_i(j) = \begin{cases} w_i(j), & \text{rand} \leq Cr, \\ X_i(j), & \text{other}. \end{cases} \tag{3.4}$$

Specifically, if the random number (rand) is less than or equal to the cross parameter $Cr$, then the term $F_i(j)$ takes the value $w_i(j)$, otherwise it will be $X_i(j)$. Next, the first generation of individuals generates new individuals through mutation and cross operations. Subsequently, the objective function is evaluated in the new individual.

(5) Steps 2–5 are repeated for a number of $k$ generations.

(6) In the last step, the lowest cost function is printed.

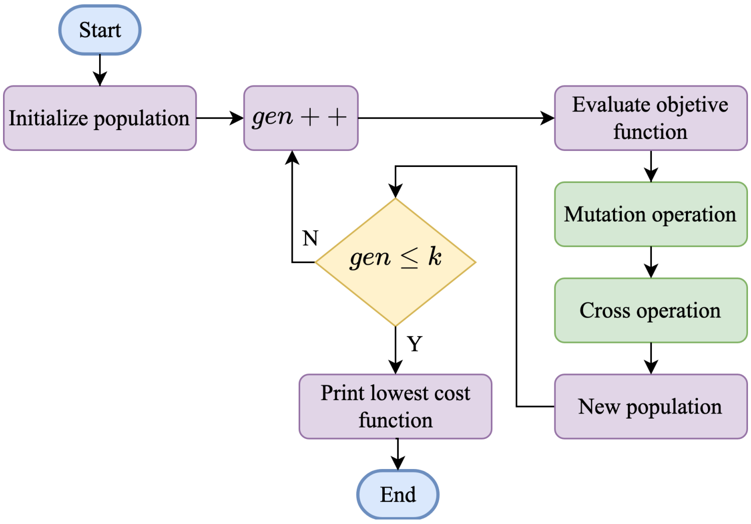


**Figure 1.** Chart flow of the Differential Evolution algorithm.

### 3.2. *Accelerated Particle Swarm Optimization algorithm*

In the Accelerated Particle Swarm Optimization (APSO) algorithm, the tentative solutions are called particles. This algorithm consists of disturbing the behavior of the particles in such a way that they can move within a search space and finding the solution to the optimization problem [48]. The algorithm is inspired by the flight of birds to find food in an unknown field with the lowest possible energy expenditure, which can be translated into finding the minimum value of a function. The structure of the APSO is illustrate in the chart flow of Figure 2, and its description is given as follows:

(1) In the first step, the initial population is generated, meaning the particles position within a range of interest.

$$X_i \in \mathbb{R}^n, \ i = 1, 2, ..., p. \tag{3.5}$$

(2) Next, the objective function is evaluated for each particle

$$\forall X_i, \ do \ f(X_i) \in \mathbb{R}. \tag{3.6}$$

(3) In the next step, the position of the particles is disturbed:

- (3.1) First, the best particles are selected

$$g^* = \min\left(f(X_i)\right). \tag{3.7}$$

- (3.2) Then, the position is updated

$$X_i(k+1) = (1-\beta) + \beta g^* + \alpha \epsilon_n. \tag{3.8}$$

Specifically, the new population is generated from the change in position of the particles. The value for $\epsilon_n$ is randomly initialized within the range $[0, 1]$ whereas $\alpha$ and $\beta$ are estimated experimentally.

(4) Steps 2–5 are repeated for a number of $k$ generations.

(5) In the last step, the lowest cost function is printed.

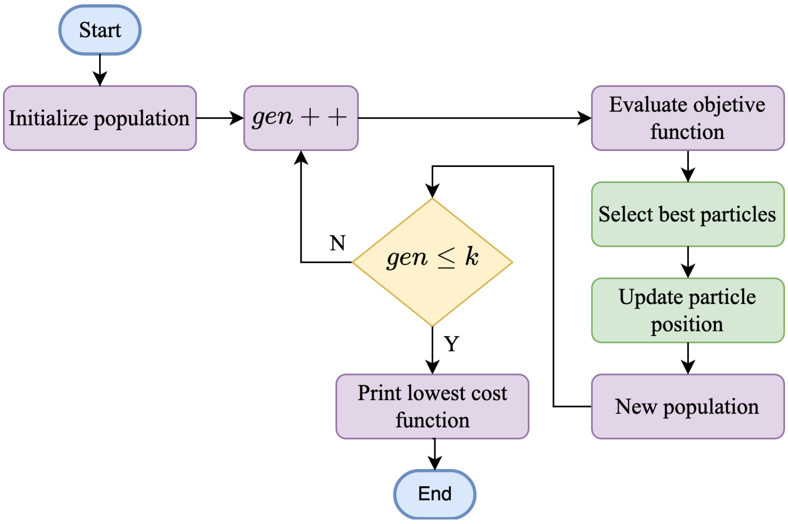


**Figure 2.** Chart flow of the Accelerated Particle Swarm Optimization algorithm.

## 4. Fractional order chaotic systems

This section shows the numerical simulations and Poincaré maps of the FO chaotic systems that are case study. For instance, the authors in [19] introduced the 3D FO chaotic system given in (4.1), where $0 < q_i < 1$, $(i = 1, 2, 3)$ and $a \in \mathbb{R}$. According to expression (4.2), it can be seen that by setting the parameter $a > 1/4$, the 3D FO chaotic system does not posses an equilibrium point ($E_p = (\bar{x}, \bar{y}, \bar{z})$), and thus, any existing basin of attraction is considered as hidden.

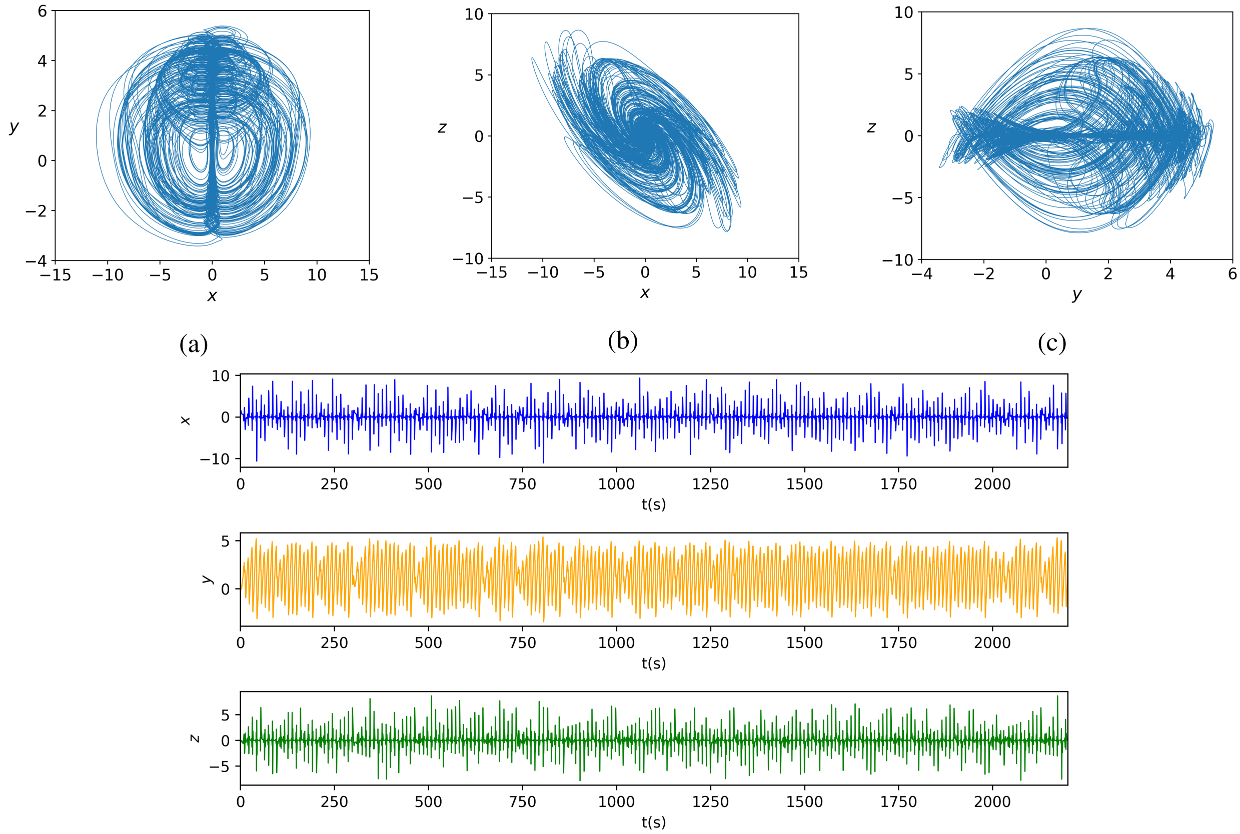
$$
\begin{aligned}
D_t^{q_1} x &= yz + x(y - a), \\
D_t^{q_2} y &= 1 - |x|, \\
D_t^{q_3} z &= -xy - z,
\end{aligned}
\tag{4.1}
$$

$$
\begin{aligned}
E_{p_1} &= (1, (1 + \sqrt{1 - 4a})/2, -(1 + \sqrt{1 - 4a})/2), \\
E_{p_2} &= ( - 1, (1 - \sqrt{1 - 4a})/2, (1 - \sqrt{1 - 4a})/2), \\
E_{p_3} &= (1, (1 - \sqrt{1 - 4a})/2, -(1 - \sqrt{1 - 4a})/2), \\
E_{p_4} &= ( - 1, (1 + \sqrt{1 - 4a})/2, (1 - \sqrt{1 - 4a})/2).
\end{aligned}
\tag{4.2}
$$

By taking into account (2.3), the solution of the 3D FO chaotic system by means of the Grünwald-Letnikov method is given in (4.3), whereas Figure 3 shows the simulation results. Figure 4 shows the LEs spectra by varying the fractional derivative.

$$
\begin{aligned}
x(t_k) &= \Big(y(t_{k-1})z(t_{k-1}) + x(t_{k-1})\big(y(t_{k-1}) - a\big)\Big)h^{q_1} - \sum_{j=1}^{k} C_j^{q_1} x(t_{k-j}), \\
y(t_k) &= (1 - |x(t_{k-1})|)h^{q_2} - \sum_{j=1}^{k} C_j^{q_2} y(t_{k-j}), \\
z(t_k) &= ( - x(t_{k-1})y(t_{k-1}) - z(t_{k-1}))h^{q_3} - \sum_{j=1}^{k} C_j^{q_3} z(t_{k-j}).
\end{aligned}
\tag{4.3}
$$

To demonstrate the usefulness of Poincaré maps in detecting chaotic behavior, a Poincaré section $\Sigma \in \mathbb{R}^2$ on $y = 2$ is defined as depicted in Figure 5 (a), then, by following the steps of Section 2.2, the Poincaré map of Figure 5 (b) is obtained. It can be seen that the displayed points resembles the $xz$ phase portrait of Figure 3 (b), the markers illustrates if the flow intersects the Poincaré section on $y > 2$ (blue, in-direction) and if it intersects on $y < 2$ (red, out-direction).



(a)

(b)

(c)

(d)

**Figure 3.** Numerical simulation of the 3D FO chaotic system with hidden attractors, with step size $h = 0.015$, simulation time $T_{sim} = 2200s$, initial conditions $(x(t_0), y(t_0), z(t_0)) = (1, 1, 1)$, parameter $a = 0.35$, and fractional order $q_i = 0.97$, $(i = 1, 2, 3)$. Hidden chaotic attractor depicted in (a) $xy$ phase portrait, (b) $xz$ phase portrait, (c) $yz$ phase portrait and (d) the corresponding time series.

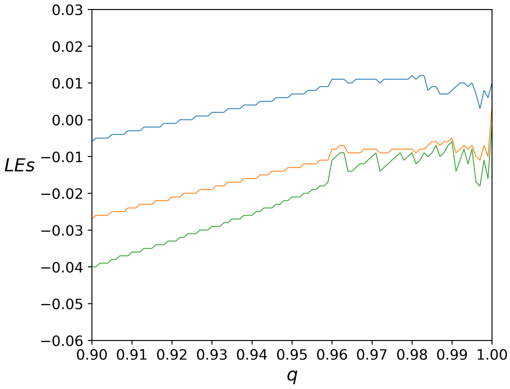
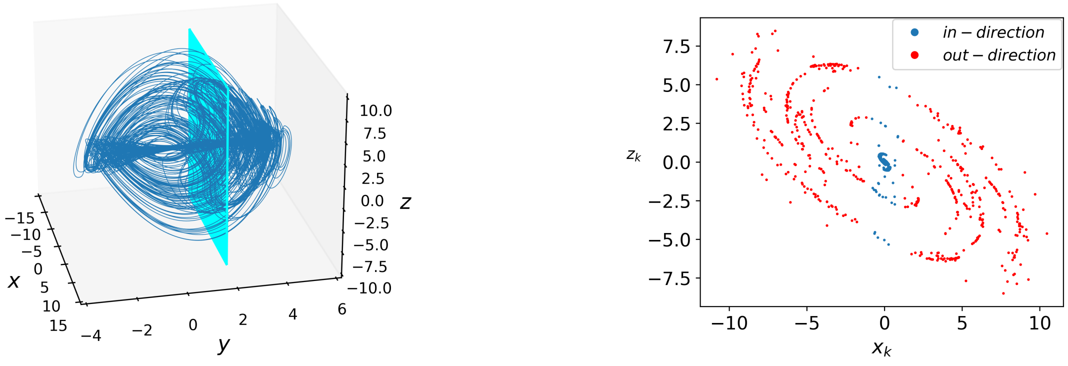


**Figure 4.** LEs spectra of the 3D FO system with hidden attractors, with $a = 0.35$ and varying the fractional-order derivative $q$. $\lambda_1$ is in blue color, $\lambda_2$ is in orange color and $\lambda_3$ is in green color.



(a) (b)

**Figure 5.** (a) Chaotic flow of the 3D FO chaotic system intersecting the Poincaré section $\Sigma \in \mathbb{R}^2$ on $y = 2$, and (b) its corresponding Poincaré map.

It is worthy to mention that the number of points portrayed in the Poincaré map is in relation to the amount of used data. However, the number of points itself says nothing about the dynamical behavior. For instance, lets us consider the dynamical behavior of the system (4.1) illustrated in Figure 6, then, a Poincaré section $\Sigma \in \mathbb{R}^2$ on $y = 2$ is defined as depicted in Figure 7 (a). The obtained Poincaré map in Figure 7 (b) displays only two points, indicating the flow intersects twice in different in-out directions of the Poincaré section, and thus, resembling a periodic behavior. Nevertheless, this is only apparent, due to the fact of the non-existence of exactly periodic solutions in FO systems [49, 50], thereby, the importance of the distribution quantization of the Poincaré map. The next step is the distribution quantization of the Poincaré map. Herein, a simple procedure is proposed as follows:

(1) Verify that the solution of a system (4.1) remains bounded but does not converge to a fixed point.

(2) Next, the first 10 % samples of the time series are deleted to avoid the initial transient.

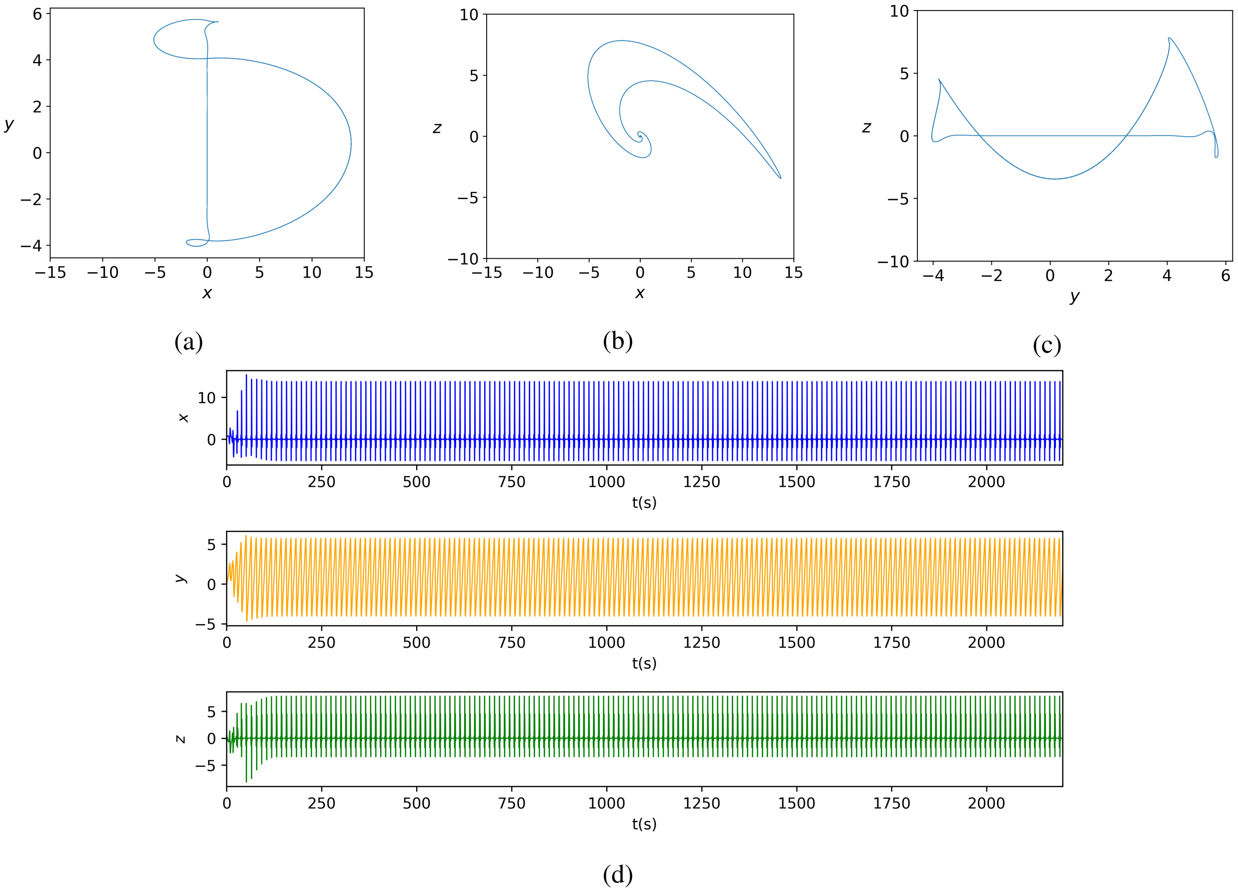
(3) Then, the Poincaré map is portrayed with the obtained points when the flow intersects on $y < 2$ (out-direction).

(4) In the next step, only the points that satisfies the condition (4.4) are considered, where $(u_j, v_j)$ and $(u_k, v_k)$ denotes the coordinates of the $P_j$ and $P_k$ points respectively, $p$ is the total number of points intersecting the Poincaré section and $\epsilon = 0.1$, this ensures to preserve only the points distributed over the plane and remove those located closer than $(\epsilon, \epsilon)$,

$$|P_j(u_j, v_j) - P_k(u_k, v_k)| \geq (\epsilon, \epsilon), \quad k = j + 1, \cdots, p, \quad \forall j = 1, 2, \cdots p - 1. \tag{4.4}$$

(5) Finally, the number of points remaining in the Poincaré map is co-related with the dynamical behavior.

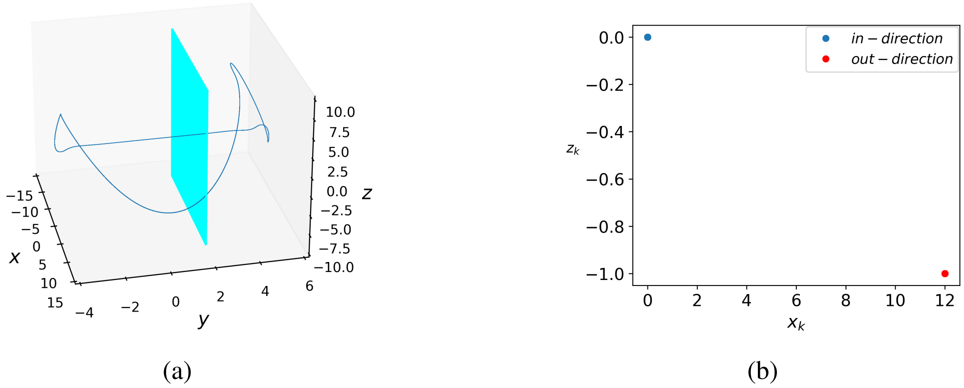
For instance, Figures 8 and 9 illustrates the hidden attractors and its corresponding Poincaré maps when system (4.1) behaves quasi-periodic. After applying 1–5 from above procedure, the obtained results in Table 1 shows that the remaining points are 10 and 3 respectively. On the other hand, when system (4.1) behaves chaotic such as in Figure 5, the remaining points in the Poincaré map must portrait $\approx$ 100 points.



(a) (b) (c)

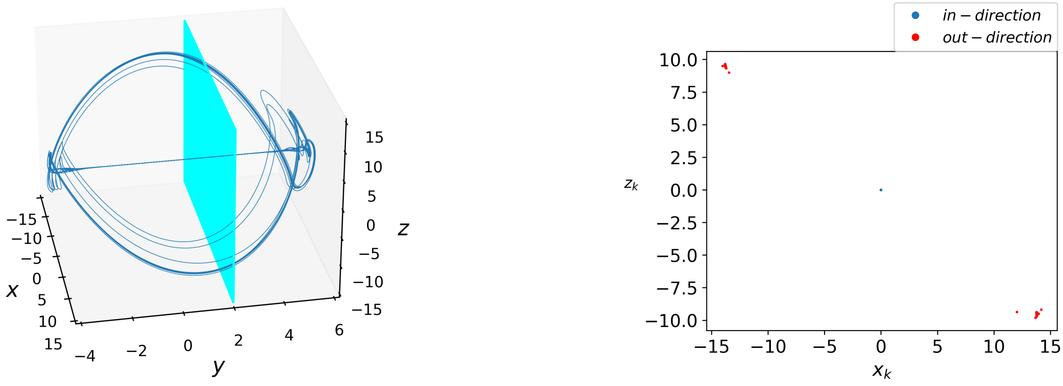
(d)

**Figure 6.** Numerical simulation of the 3D FO system (4.1) with hidden attractors, with step size $h = 0.015$, simulation time $T_{sim} = 2200s$, initial conditions $(x(t_0), y(t_0), z(t_0)) = (1, 0, 1)$, parameter $a = 0.35$, and fractional order $q_i = 0.996$, $(i = 1, 2, 3)$. Hidden chaotic attractor depicted in (a) $xy$ phase portrait, (b) $xz$ phase portrait, (c) $yz$ phase portrait and (d) the corresponding time series.



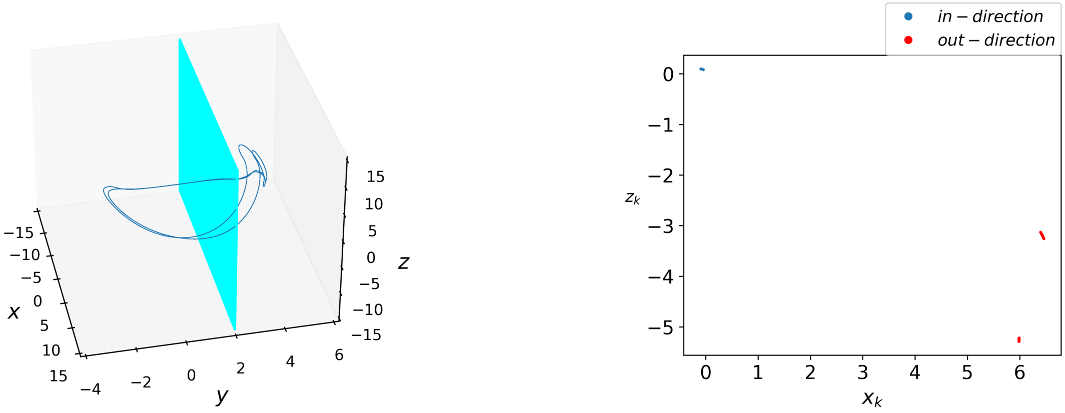
(a) (b)

**Figure 7.** (a) Quasi-periodic flow of the 3D FO system intersecting the Poincaré section $\Sigma \in \mathbb{R}^2$ on $y = 2$, and (b) its corresponding Poincaré map.



(a) (b)

**Figure 8.** (a) Quasi-periodic flow of the 3D FO sytem (4.1) with configuration $a = 0.436$, $(q_1, q_2, q_3) = (0.998, 0.989, 0.986)$ and initial conditions $(x(t_0), y(t_0), z(t_0)) = (1, 0, 1)$, intersecting the Poincaré section $\Sigma \in \mathbb{R}^2$ on $y = 2$, (b) shows its corresponding Poincaré map.



(a) (b)

**Figure 9.** (a) Quasi-periodic flow of the 3D FO sytem (4.1) with configuration $a = 0.277$, $(q_1, q_2, q_3) = (0.988, 0.948, 0.881)$ and initial conditions $(x(t_0), y(t_0), z(t_0)) = (1, 0, 1)$, intersecting the Poincaré section $\Sigma \in \mathbb{R}^2$ on $y = 2$, (b) shows its corresponding Poincaré map.

**Table 1.** Relation between the remainder number of points in the Poincaré map of the 3D FO system (4.1) and its dynamical behavior. Obtained results with step size $h = 0.025$, simulation time $T_{sim} = 2200s$, and initial conditions $(x(t_0), y(t_0), z(t_0)) = (1, 0, 1)$.

| Parameters | Fractional order derivative | Number of points | LEs | $D_{KY}$ | Dynamical behavior |
|---|---|---|---|---|---|
| $a = 0.35$ | $q_1 = 0.97$,<br>$q_2 = 0.97$,<br>$q_3 = 0.97$ | 107 | $\lambda_1 = 0.0011$,<br>$\lambda_2 = -0.0006$,<br>$\lambda_3 = -0.0036$ | 2.1504 | chaotic |
| $a = 0.3$ | $q_1 = 0.993$,<br>$q_2 = 1$,<br>$q_3 = 0.75$ | 96 | $\lambda_1 = 0.0010$,<br>$\lambda_2 = -0.0009$,<br>$\lambda_3 = -0.0047$ | 2.0208 | chaotic |
| $a = 0.325$ | $q_1 = 1$,<br>$q_2 = 0.99$,<br>$q_3 = 0.85$ | 104 | $\lambda_1 = 0.0018$,<br>$\lambda_2 = -0.0006$,<br>$\lambda_3 = -0.0057$ | 2.2071 | chaotic |
| $a = 0.337$ | $q_1 = 0.966$,<br>$q_2 = 0.968$,<br>$q_3 = 0.926$ | 105 | $\lambda_1 = 0.0013$,<br>$\lambda_2 = -0.0008$,<br>$\lambda_3 = -0.0036$ | 2.1452 | chaotic |
| $a = 0.6$ | $q_1 = 0.992$,<br>$q_2 = 0.938$,<br>$q_3 = 0.968$ | 94 | $\lambda_1 = 0.0022$,<br>$\lambda_2 = -0.0009$,<br>$\lambda_3 = -0.0051$ | 2.2452 | chaotic |
| $a = 0.437$ | $q_1 = 0.993$,<br>$q_2 = 0.956$,<br>$q_3 = 0.874$ | 105 | $\lambda_1 = 0.0010$,<br>$\lambda_2 = -0.0009$,<br>$\lambda_3 = -0.0043$ | 2.003 | chaotic |
| $a = 0.322$ | $q_1 = 0.982$,<br>$q_2 = 0.955$,<br>$q_3 = 0.985$ | 96 | $\lambda_1 = 0.0017$,<br>$\lambda_2 = -0.0009$,<br>$\lambda_3 = -0.0052$ | 2.1593 | chaotic |
| $a = 0.29$ | $q_1 = 0.986$,<br>$q_2 = 0.967$,<br>$q_3 = 0.981$ | 17 | $\lambda_1 = 0.0008$,<br>$\lambda_2 = -0.0017$,<br>$\lambda_3 = -0.0018$ | 1.5219 | quasi-periodic |
| $a = 0.436$ | $q_1 = 0.998$,<br>$q_2 = 0.989$,<br>$q_3 = 0.986$ | 10 | $\lambda_1 = 0.0007$,<br>$\lambda_2 = -0.0012$,<br>$\lambda_3 = -0.0028$ | 1.8214 | quasi-periodic |
| $a = 0.277$ | $q_1 = 0.988$,<br>$q_2 = 0.948$,<br>$q_3 = 0.881$ | 3 | $\lambda_1 = 0.0009$,<br>$\lambda_2 = -0.0038$,<br>$\lambda_3 = -0.0183$ | 1.8415 | quasi-periodic |
| $a = 0.382$ | $q_1 = 0.994$,<br>$q_2 = 0.922$,<br>$q_3 = 97$ | 2 | $\lambda_1 = -0.0003$,<br>$\lambda_2 = -0.0011$,<br>$\lambda_3 = -0.0015$ | 1.06 | quasi-periodic |
| $a = 0.387$ | $q_1 = 0.976$,<br>$q_2 = 0.968$,<br>$q_3 = 0.804$ | 2 | $\lambda_1 = 0.0001$,<br>$\lambda_2 = -0.0011$,<br>$\lambda_3 = -0.0016$ | 1.375 | quasi-periodic |

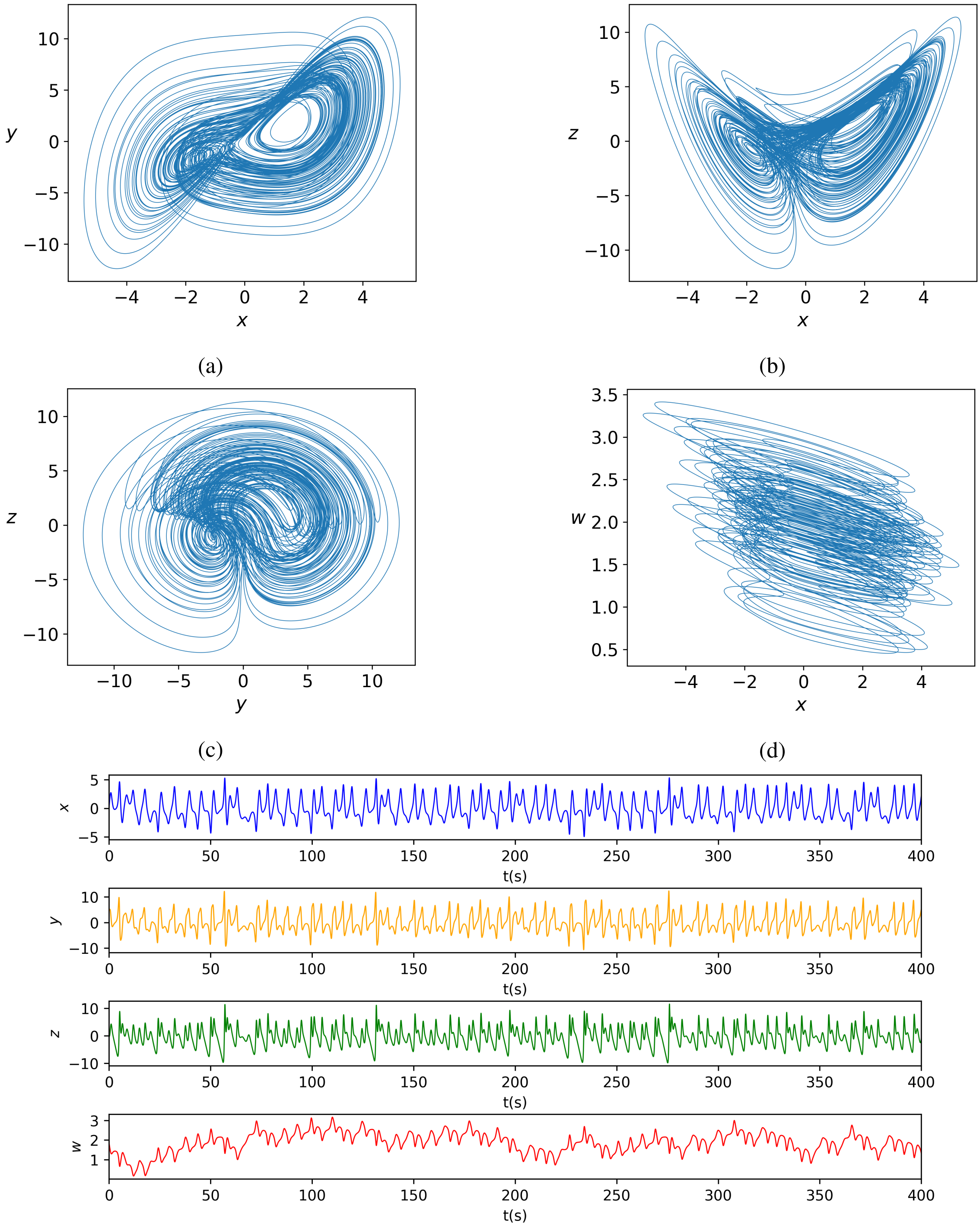
The second case study is a 4D FO chaotic system introduced in [51] as a 4D integer chaotic system with coexisting hidden attractors. Later on, in [52] the FO version given in (4.5) was proposed, where $0 < q_i < 1$, $(i = 1, 2, 3)$ and $a \in \mathbb{R}$. According to expression (4.6), it can be seen that for $a \neq 0$, the 4D FO chaotic system does not present an equilibrium point ($E_p = (\bar{x}, \bar{y}, \bar{z}, \bar{w})$), and thus, any existing basin of attraction is considered as hidden.

$$\begin{aligned} D_t^{q_1} x &= y - x, \\ D_t^{q_2} y &= -xz + w, \\ D_t^{q_3} z &= xy - a, \\ D_t^{q_4} z &= -by, \end{aligned} \tag{4.5}$$

$$\begin{aligned} y - x &= 0, \\ -xz + w &= 0, \\ xy - a &= 0, \\ -by &= 0. \end{aligned} \tag{4.6}$$

By taking into account (2.3), the solution of the 4D FO chaotic system by means of the Grünwald-Letnikov method is given in (4.7), whereas Figure 10 shows the simulation results. Figure 11 shows the LEs spectra by varying the fractional derivative.

$$\begin{aligned} x(t_k) &= (y(t_{k-1}) - x(t_{k-1}))h^{q_1} - \sum_{j=1}^{k} C_j^{q_1} x(t_{k-j}), \\ y(t_k) &= ( - x(t_{k-j})z(t_{k-j}) + w(t_{k-j}))h^{q_2} - \sum_{j=1}^{k} C_j^{q_2} y(t_{k-j}), \\ z(t_k) &= (x(t_{k-1})y(t_{k-1}) - a))h^{q_3} - \sum_{j=1}^{k} C_j^{q_3} z(t_{k-j}), \\ z(t_k) &= ( - by(t_{k-1}))h^{q_3} - \sum_{j=1}^{k} C_j^{q_4} w(t_{k-j}). \end{aligned} \tag{4.7}$$



(a)

(b)

(c)

(d)

(e)

**Figure 10.** Numerical simulation of the 4D FO chaotic system with hidden attractors, with step size $h = 0.003$, simulation time $T_{sim} = 400s$, initial conditions $(x(t_0), y(t_0), z(t_0), w(t_0)) = (0.1, 5, 0, 2)$, parameters $a = 3.5$, $b = 0.1$ and fractional order $q_i = 0.98$, $(i = 1, \cdots, 4)$. Hidden chaotic attractor depicted in (a) $xy$ phase portrait, (b) $xz$ phase portrait, (c) $yz$ phase portrait, (d) $xw$ phase portrait and (e) the corresponding time series.

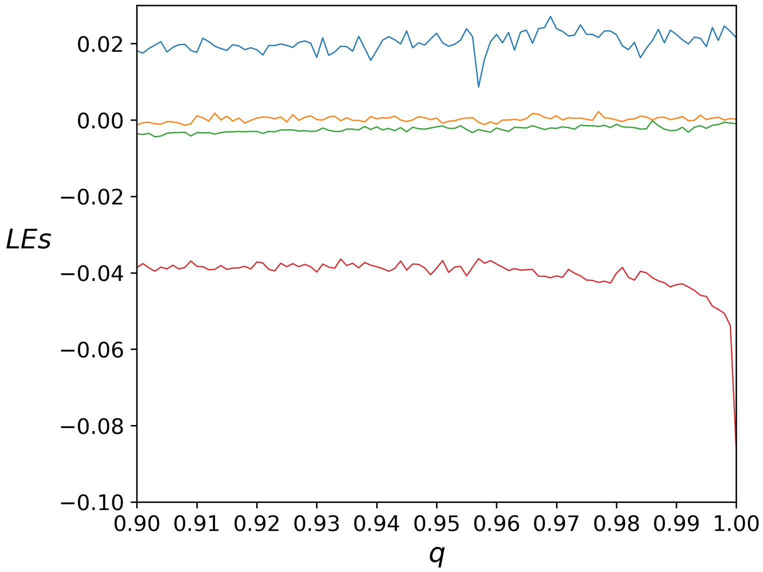
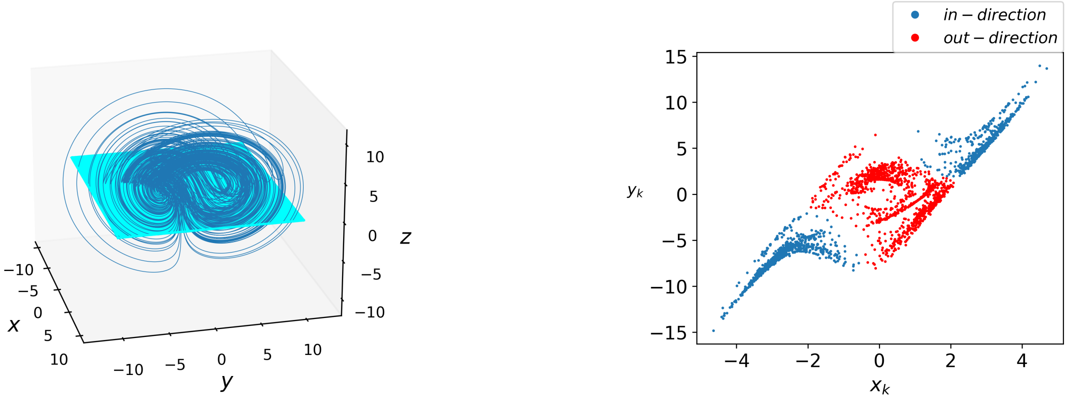


**Figure 11.** LEs spectra of the 4D FO system with hidden attractors (4.5), with $a = 0.35$ and varying the fractional-order derivative $q$. $\lambda_1$ is in blue color, $\lambda_2$ is in orange color, $\lambda_3$ is in green color and $\lambda_4$ is in red color.

In the same manner, a Poincaré map is elaborated for analyzing the behavior of the 4D FO chaotic system. Herein, the Poincaré section $\Sigma \in \mathbb{R}^2$ is settled on $z = 0$ as depicted in Figure 12 (a), then, by following the steps of Section 2.2, the Poincaré map of Figure 12 (b) is obtained.



(a)

(b)

**Figure 12.** (a) Chaotic flow of the 4D FO chaotic system intersecting the Poincaré section $\Sigma \in \mathbb{R}^2$ on $z = 2$, and (b) its corresponding Poincaré map.

The distribution quantization of the Poincaré map of the 4D FO chaotic system is quite similar to the quantization of the Poincaré map of the 3D FO chaotic system. Herein, only the obtained points when the flow intersects on $z > 0$ (in-direction) are considered with a separation of $\epsilon = 0.1$. This procedure is repeated for different parameters and fractional order for the system (4.5). The obtained results with a number of data of $1.20\times10^5$ are shown in Table 2. It can be seen that for chaotic behavior to exist in the system (4.1), its corresponding Poincaré map must portrait $\approx 40 - 60$ points.

**Table 2.** Relation between the remainder number of points in the Poincaré map of the 4D FO system (4.1) and its dynamical behavior. Obtained results with step size $h = 0.003$, simulation time $T_{sim} = 400s$, and initial conditions $(x(t_0), y(t_0), z(t_0), w(t_0)) = (0.1, 5, 0, 2)$.

| Parameters | Fractional order derivative | Number of points | LEs | $D_{KY}$ | Dynamical behavior |
|---|---|---|---|---|---|
| $a = 3.5$,<br>$b = 0.1$ | $q_1 = 0.98, q_2 = 0.98$,<br>$q_3 = 0.98, q_4 = 0.98$ | 43 | $\lambda_1 = 0.0011, \lambda_2 = 0$,<br>$\lambda_3 = -0.0001, \lambda_4 = -0.0024$ | 3.4257 | chaotic |
| $a = 3.5$,<br>$b = 0.1$ | $q_1 = 1.0, q_2 = 0.997$,<br>$q_3 = 1.0, q_4 = 0.45$ | 50 | $\lambda_1 = 0.0030, \lambda_2 = 0$,<br>$\lambda_3 = -0.0017, \lambda_4 = -0.0040$ | 3.3368 | chaotic |
| $a = 2.9$,<br>$b = 0.1$ | $q_1 = 0.45, q_2 = 0.95$,<br>$q_3 = 1.0, q_4 = 0.985$ | 40 | $\lambda_1 = 0.0008, \lambda_2 = 0$,<br>$\lambda_3 = -0.0003, \lambda_4 = -0.0030$ | 3.1666 | chaotic |
| $a = 3.162$,<br>$b = 0.182$ | $q_1 = 0.980, q_2 = 0.942$,<br>$q_3 = 0.978, q_4 = 0.975$ | 42 | $\lambda_1 = 0.0008, \lambda_2 = 0.0001$,<br>$\lambda_3 = 0, \lambda_4 = -0.0018$ | 3.5029 | hyper-chaotic |
| $a = 3.128$,<br>$b = 0.285$ | $q_1 = 0.946, q_2 = 0.964$,<br>$q_3 = 0.957, q_4 = 0.923$ | 41 | $\lambda_1 = 0.0003, \lambda_2 = 0.0001$,<br>$\lambda_3 = 0, \lambda_4 = -0.0006$ | 3.5103 | hyper-chaotic |
| $a = 3.131$,<br>$b = 0.285$ | $q_1 = 0.945, q_2 = 0.963$,<br>$q_3 = 0.956, q_4 = 0.922$ | 41 | $\lambda_1 = 0.0004, \lambda_2 = 0.0002$,<br>$\lambda_3 = 0,\lambda_4 = -0.0011$ | 3.5282 | hyper-chaotic |
| $a = 3.088$,<br>$b = 0.12$ | $q_1 = 0.925, q_2 = 0.948$,<br>$q_3 = 0.933, q_4 = 0.932$ | 38 | $\lambda_1 = 0.0016, \lambda_2 = 0$,<br>$\lambda_3 = -0.0004,\lambda_4 = -0.0031$ | 3.387 | chaotic |
| $a = 3.601$,<br>$b = 0.3$ | $q_1 = 0.999, q_2 = 0.999$,<br>$q_3 = 0.999, q_4 = 0.962$ | 48 | $\lambda_1 = 0.0005, \lambda_2 = 0$,<br>$\lambda_3 = -0.0001, \lambda_4 = -0.0022$ | 3.2014 | chaotic |
| $a = 9.4$,<br>$b = 0.01$ | $q_1 = 0.936, q_2 = 0.935$,<br>$q_3 = 0.958, q_4 = 0.946$ | 60 | $\lambda_1 = 0.0009, \lambda_2 = 0$,<br>$\lambda_3 = -0.0001, \lambda_4 = -0.0013$ | 3.5972 | chaotic |
| $a = 8.4$,<br>$b = 0.1$ | $q_1 = 0.990, q_2 = 0.980$,<br>$q_3 = 0.980, q_4 = 0.981$ | 8 | $\lambda_1 = 0, \lambda_2 = -0.0002$,<br>$\lambda_3 = -0.0014, \lambda_4 = -0.0039$ | 2.5897 | quasi-periodic |
| $a = 7.8$,<br>$b = 0.1$ | $q_1 = 0.990, q_2 = 0.980$,<br>$q_3 = 0.970, q_4 = 0.981$ | 10 | $\lambda_1 = 0, \lambda_2 = -0.0002$,<br>$\lambda_3 = -0.0011, \lambda_4 = -0.0041$ | 2.6829 | quasi-periodic |
| $a = 7.3$,<br>$b = 0.08$ | $q_1 = 0.989, q_2 = 0.998$,<br>$q_3 = 0.967, q_4 = 0.981$ | 11 | $\lambda_1 = -0.0001, \lambda_2 = -0.0002$,<br>$\lambda_3 = -0.0007, \lambda_4 = -0.0042$ | 2.7619 | quasi-periodic |

## 5. Description of the optimization process

This section details the maximization process of the $D_{KY}$ for the 3D and 4D FO chaotic systems. Given the nature of the DE and APSO, the optimization problem is given as follows:

$$\text{Maximize}(f(X)) = \text{Minimize}(-f(X)), \ f(X) \in \mathbb{R}, \tag{5.1}$$

where $X \in \mathbb{R}^n$ are the parameters and fractional order derivative values of the FO chaotic system of interest and $f(X)$ returns the $D_{KY}$.

The optimization process is performed applying DE and APSO who are programmed in Python, because it is a multi-paradigm programming language highly versatile for creating all kind of applications and, therefore, it has become one of the most used in recent years. However, Python based applications pays a penalty in the form of speed execution, thereby complex projects can benefit from the use of the Numba JIT compiler [53], which allows to translate Python functions to optimized machine code and thus obtaining better speed performance [54]. Moreover, the implementation of the Numba JIT compiler only requires from slight modifications in the original Python code and, therefore, maintaining the flexibility of the language. Table 3 shows a comparison of the speed performance between a normal Python code and one Python code using the Numba JIT compiler. In both executions, the argument is to solve the proposed FO chaotic system and elaborate its corresponding Poincaré maps for the chaotic behavior validation. It can be seen, that the optimized Python code performs ≈ 375 times faster than the non-optimized. Thereby the interest of using this compiler in the optimization process.

**Table 3.** Execution time in seconds, between a Python code and a Python code using the Numba Jit compiler.

| FO chaotic system | Python code | Code using the Numba Jit compiler | Number of generated data |
|---|---|---|---|
| 3D | $1.025 \times 10^3 s$ | $2.61 s$ | $146,666$ |
| 4D | $2.085 \times 10^3 s$ | $5.55 s$ | $133,332$ |

Algorithms 1 and 2 show the pseudocodes for the implementation of DE and APSO, respectively. It can be seen that the implementation is quite similar in both algorithms. The Numba JIT compiler is used in most of the Python code with the exception of the Lyapunov exponents (LEs) computation $D_{KY}$ computation, in this step, the computation of the Lyapunov exponents $\lambda_i$, $(i = 1, 2, ..., n)$ is made with the TISEAN package by removing the first 10% samples of the time series.

**Algorithm 1** Maximizing $D_{KY}$ applying DE

```
Require: $\mu$, $C_r$, $p$, $gen$.
Ensure: best $D_j$, $j = 0, 2, ..., gen$
  Initialize population $X_i \in \mathbb{R}^n$, $i = 1, 2, ..., p$
  for $i = 1 : p$ do
    Compute numerical solution for each $X_i$ using (2.3)
    Calculate $D_{KY_i}$ using (2.5)
  store $D_0 \leftarrow$ best $D_{KY_i}$
  for $j = 1 : gen$ do
    for $k = 1 : p$ do
      generate $X_k$ using (3.3) and (3.4)
      Compute numerical solution for each new $X_i$ using (2.3)
      Detect chaotic behavior using Poincaré map
      Compute LEs using Tisean
      Calculate $D_{KY_k}$ using (2.5)
    store $D_j \leftarrow$ best $D_{KY_k}$
```

**Algorithm 2** Maximizing $D_{KY}$ applying APSO

```
Require: $\alpha, \beta, p, gen.$
Ensure: Best $D_j$, for $j = 0, 2, \ldots, gen.$
  Initialize population $X_i \in \mathbb{R}^n,\ i = 1, 2, ..., p$
  for $i = 1 : p$ do
      Compute numerical solution for each $X_i$ using (2.3)
      Calculate $D_{KY_i}$ using (2.5)
  Store $D_0 \leftarrow$ best $D_{KY_i}$
  for $j = 1 : gen$ do
      for $k = 1 : p$ do
          Generate $X_k$ using (3.8)
          Compute numerical solution for each new $X_i$ using (2.3)
          Detect chaotic behavior using Poincaré map
          Compute LEs using Tisean
          Calculate $D_{KY_k}$ using (2.5)
      Store $D_j \leftarrow$ best $D_{KY_k}$
```

## 6. Optimization results

The DE and APSO were executed for a 100 population and for 50–100 generations. Table 4 shows the ranges of interest and the configuration of the metaheuristics for the optimization of the 3D and 4D FO chaotic systems, respectively. The values of the parameters and fractional-order derivatives in the search space, were encoded within three decimal places. For convenience, the initial population is set with parameters and fractional-order derivatives that lead to chaotic behavior and, therefore, the elaboration of Poincaré maps is not necessary for this step. In Tables 5 and 6 are shown the optimization results of the 3D and 4D FO chaotic systems by DE and APSO, respectively. It can be seen that in all cases, the obtained results lead to higher $D_{KY}$ than those shown in Tables 1 and 2, thus validating the optimization process. The best results for the 3D FO chaotic system are those obtained APSO by using 50 generations whereas the best results for the 4D FO chaotic system are obtained with DE by using 100 generations.

**Table 4.** Range of parameters and fractional order derivative values for the optimization of the 3D and 4D FO chaotic systems.

| FO chaotic system | Parameters | FO derivative | DE parameters | APSO parameters |
|---|---|---|---|---|
| 3D | $a \in [0.251, 0.5]$ | $q_i \in [0.3, 0.999]$ | $\mu = 0.5, C_r = 0.6$ | $\alpha = 0.1, \beta = 0.1$ |
| 4D | $a \in [0.251, 0.45]$, $b \in [-0.1, 0.3]$ | $q_i \in [0.3, 0.999]$ | $\mu = 0.5, C_r = 0.6$ | $\alpha = 0.3, \beta = 0.1$ |

**Table 5.** Optimization results of the 3D and 4D FO chaotic systems by DE.

| DE optimization results | | | | | | | |
|---|---|---|---|---|---|---|---|
| FO chaotic system | Generations | Parameters | Fractional order derivative | Number of points | LEs | $D_{KY}$ | Dynamical behavior |
| 3D | 50 | $a = 0.351$ | $q_1 = 0.976$,<br>$q_2 = 0.956$,<br>$q_3 = 0.906$ | 101 | $\lambda_1 = 0.0023$,<br>$\lambda_2 = -0.0004$,<br>$\lambda_3 = -0.0034$ | 2.5282 | chaotic |
| | 100 | $a = 0.322$ | $q_1 = 0.986$,<br>$q_2 = 0.927$,<br>$q_3 = 0.971$ | 92 | $\lambda_1 = 0.0029$,<br>$\lambda_2 = -0.0005$,<br>$\lambda_3 = -0.0032$ | 2.6120 | chaotic |
| 4D | 50 | $a = 3.291$,<br>$b = 0.099$ | $q_1 = 0.936$,<br>$q_2 = 0.965$,<br>$q_3 = 0.936$,<br>$q_4 = 0.887$ | 42 | $\lambda_1 = 0.0020$,<br>$\lambda_2 = 0$,<br>$\lambda_3 = -0.0003$,<br>$\lambda_4 = -0.0023$ | 3.6803 | chaotic |
| | 100 | $a = 2.688$,<br>$b = 0.152$ | $q_1 = 0.951$,<br>$q_2 = 0.970$,<br>$q_3 = 0.910$,<br>$q_4 = 0.970$ | 41 | $\lambda_1 = 0.0144$,<br>$\lambda_2 = 0.0027$,<br>$\lambda_3 = 0$,<br>$\lambda_4 = -0.0201$ | 3.8543 | hyperchaotic |

**Table 6.** Optimization results of the 3D and 4D FO chaotic systems by APSO.

| APSO optimization results | | | | | | | |
|---|---|---|---|---|---|---|---|
| FO chaotic system | Generations | Parameters | Fractional order derivative | Number of points | LEs | $D_{KY}$ | Dynamical behavior |
| 3D | 50 | $a = 0.378$ | $q_1 = 0.999$,<br>$q_2 = 0.969$,<br>$q_3 = 0.902$ | 101 | $\lambda_1 = 0.0026$,<br>$\lambda_2 = -0.0005$,<br>$\lambda_3 = -0.0032$ | 2.66 | chaotic |
| | 109 | $a = 0.38$ | $q_1 = 0.998$,<br>$q_2 = 0.970$,<br>$q_3 = 0.902$ | 113 | $\lambda_1 = 0.0034$,<br>$\lambda_2 = -0.0006$,<br>$\lambda_3 = -0.0045$ | 2.5978 | chaotic |
| 4D | 50 | $a = 3.193$,<br>$b = 0.299$ | $q_1 = 0.965$,<br>$q_2 = 0.980$,<br>$q_3 = 0.985$,<br>$q_4 = 0.967$ | 40 | $\lambda_1 = 0.0007$,<br>$\lambda_2 = 0.0003$,<br>$\lambda_3 = -0.0001$,<br>$\lambda_4 = -0.0011$ | 3.7633 | hyperchaotic |
| | 100 | $a = 3.106$,<br>$b = 0.19$ | $q_1 = 0.878$,<br>$q_2 = 0.957$,<br>$q_3 = 0.945$,<br>$q_4 = 0.970$ | 41 | $\lambda_1 = 0.0008$,<br>$\lambda_2 = 0.0002$,<br>$\lambda_3 = 0$,<br>$\lambda_4 = -0.0015$ | 3.6358 | hyperchaotic |

## 7. Conclusions

In this paper, the optimization of the $D_{KY}$ of a 3D and a 4D FO chaotic systems was performed by applying two metaheuristics, DE and APSO. The main contribution of this work is the use of Poincaré maps for validating the chaotic behavior in FO chaotic systems with hidden attractors. Herein a simple numerical procedure was proposed for the analysis of points distribution in Poincaré maps. It has been shown that with this simple procedure, it is possible to differentiate a chaotic attractor from a periodic or quasi-periodic attractor. Moreover, the use of the Numba JIT compiler in the Python programming language has been an essential tool to save execution time, while preserving the versatility in the implementation of the metaheuristics. This last part is essential and it has been demonstrated that it facilitates the optimization process of FO chaotic systems with hidden attractors. Finally, it is worthy to mention that Poincaré maps can be used in both FO and integer chaotic systems regardless of the attractor nature (hidden or self-excited). The proposed approach can be applied in FO chaotic systems with self-excited attractor as done in integer chaotic systems with hidden and self-excited attractors.

## Acknowledgments

Daniel Clemente-López acknowledges CONACYT for PhD scholarship with CVU no. 859036.

## Conflict of interest

The authors declare no conflicts of interests.

## References

1. Y. Bolotin, A. Tur, V. Yanovsky, *Chaos: Concepts, control and constructive use*, Springer, 2009.
2. B. Dubey, Sajan, A. Kumar, Stability switching and chaos in a multiple delayed prey-predator model with fear effect and anti-predator behavior, *Math. Comput. Simulat.*, **188** (2021), 164–192. https://doi.org/10.1016/j.matcom.2021.03.037
3. J. K. Zink, K. Batygin, F. C. Adams, The great inequality and the dynamical disintegration of the outer solar system, *Astron. J.*, **160** (2020), 232.
4. M. Sajid, Recent developments on chaos in mechanical systems, *Int. J. Theor. Appl. Res. Mech. Eng.*, **2** (2013), 121–124.
5. B. A. Idowu, S. Vaidyanathan, A. Sambas, O. I. Olusola, O. S. Onma, A new chaotic finance system: Its analysis, control, synchronization and circuit design, In: *Nonlinear dynamical systems with self-excited and hidden attractors*, Springer, 2018, 271–295. https://doi.org/10.1007/978-3-319-71243-7_12
6. B Wang, S. M. Zhong, X. C. Dong, On the novel chaotic secure communication scheme design, *Commun. Nonlinear Sci. Numer. Simul.*, **39** (2016), 108–117. https://doi.org/10.1016/j.cnsns.2016.02.035

7. D. Arroyo, F. Hernandez, A. B. Orúe, Cryptanalysis of a classical chaos-based cryptosystem with some quantum cryptography features, *Int. J. Bifurcat. Chaos*, **27** (2017), 1750004. https://doi.org/10.1142/S0218127417500043

8. X. Zang, S. Iqbal, Y. Zhu, X. Liu, J. Zhao, Applications of chaotic dynamics in robotics, *Int. J. Adv. Robotic Syst.*, **13** (2016), 60. https://doi.org/10.5772/62796

9. K. Tian, C. Grebogi, H. P. Ren, Chaos generation with impulse control: Application to non-chaotic systems and circuit design, *IEEE T. Circuits Syst. I*, **68** (2021), 3012–3022. https://doi.org/10.1109/TCSI.2021.3075550

10. V. E. Tarasov, Generalized memory: Fractional calculus approach, *Fractal Fract.*, **2** (2018), 23. https://doi.org/10.3390/fractalfract2040023

11. K. M. Owolabi, A. Atangana, J. F. Gómez-Aguilar, Fractional adams-bashforth scheme with the liouville-caputo derivative and application to chaotic systems, *Discrete Cont. Dyn. Syst.-S*, **14** (2021), 2455–2469. https://doi.org/10.3934/dcdss.2021060

12. M. Ahmad, U. Shamsi, I. R. Khan, An enhanced image encryption algorithm using fractional chaotic systems, *Procedia Comput. Sci.*, **57** (2015), 852–859. https://doi.org/10.1016/j.procs.2015.07.494

13. M. Bettayeb, U. M. Al-Saggaf, S. Djennoune, Single channel secure communication scheme based on synchronization of fractional-order chaotic Chua's systems, *T. I. Meas. Control*, **40** (2018), 3651–3664. https://doi.org/10.1177/0142331217729425

14. H. Natiq, M. R. M. Said, M. R. K. Ariffin, S. He, L. Rondoni, S. Banerjee, Self-excited and hidden attractors in a novel chaotic system with complicated multistability, *Eur. Phys. J. Plus*, **133** (2018), 1–12. https://doi.org/10.1140/epjp/i2018-12360-y

15. G. A. Leonov, N. V. Kuznetsov, Hidden attractors in dynamical systems. From hidden oscillations in hilbert-kolmogorov, aizerman, and kalman problems to hidden chaotic attractor in chua circuits, *Int. J. Bifurcat. Chaos*, **23** (2013), 1330002. https://doi.org/10.1142/S0218127413300024

16. P. R. Sharma, M. D. Shrimali, A. Prasad, N. V. Kuznetsov, G. A. Leonov, Control of multistability in hidden attractors, *Eur. Phys. J. Spec. Top.*, **224** (2015), 1485–1491. https://doi.org/10.1140/epjst/e2015-02474-y

17. A. K. Farhan, R. S. Ali, H. Natiq, N. M. G. Al-Saidi, A new s-box generation algorithm based on multistability behavior of a plasma perturbation model, *IEEE Access*, **7** (2019), 124914–124924. https://doi.org/10.1109/ACCESS.2019.2938513

18. F. Yu, Z. Zhang, L. Liu, H. Shen, Y. Huang, C. Shi, et al., Secure communication scheme based on a new 5D multistable four-wing memristive hyperchaotic system with disturbance inputs, *Complexity*, **2020** (2020), 5859273. https://doi.org/10.1155/2020/5859273

19. J. M. Munoz-Pacheco, E. Zambrano-Serrano, C. Volos, S. Jafari, J. Kengne, K. Rajagopal, A new fractional-order chaotic system with different families of hidden and self-excited attractors, *Entropy*, **20** (2018), 564. https://doi.org/10.3390/e20080564

20. N. Debbouche, S. Momani, A. Ouannas, M. T. Shatnawi, G. Grassi, Z. Dibi, et al., Generating multidirectional variable hidden attractors via newly commensurate and

incommensurate non-equilibrium fractional-order chaotic systems, *Entropy*, **23** (2021), 261. https://doi.org/10.3390/e23030261

21. D. A. Yousri, A. M. AbdelAty, L. A. Said, A. S. Elwakil, B. Maundy, A. G. Radwan, Parameter identification of fractional-order chaotic systems using different meta-heuristic optimization algorithms, *Nonlinear Dynam.*, **95** (2019), 2491–2542. https://doi.org/10.1007/s11071-018-4703-2
22. A. Silva-Juárez, E. Tlelo-Cuautle, L. G. de la Fraga, R. Li, Optimization of the kaplan-yorke dimension in fractional-order chaotic oscillators by metaheuristics, *Appl. Math. Comput.*, **394** (2021), 125831. https://doi.org/10.1016/j.amc.2020.125831
23. J. C. Nunez-Perez, V. A. Adeyemi, Y. Sandoval-Ibarra, F. J. Perez-Pinal, E. Tlelo-Cuautle, Maximizing the chaotic behavior of fractional order chen system by evolutionary algorithms, *Mathematics*, **9** (2021), 1194. https://doi.org/10.3390/math9111194
24. S. Jafari, J. C. Sprott, F. Nazarimehr, Recent new examples of hidden attractors, *Eur. Phys. J. Spec. Top.*, **224** (2015), 1469–1476. https://doi.org/10.1140/epjst/e2015-02472-1
25. W. S. Sayed, A. G. Radwan, Self-reproducing hidden attractors in fractional-order chaotic systems using affine transformations, *IEEE Open J. Circuits Syst.*, **1** (2020), 243–254. https://doi.org/10.1109/OJCAS.2020.3030756
26. M. F. Danca, P. Bourke, N. Kuznetsov, Graphical structure of attraction basins of hidden chaotic attractors: The rabinovich-fabrikant system, *Int. J. Bifurcat. Chaos*, **29** (2019), 1930001. https://doi.org/10.1142/S0218127419300015
27. C. W. Kulp, B. J. Niskala, *Characterization of time series data*, 2017.
28. S. Zamen, E. Dehghan-Niri, Observation and diagnosis of chaos in nonlinear acoustic waves using phase-space domain, *J. Sound Vib.*, **463** (2019), 114959. https://doi.org/10.1016/j.jsv.2019.114959
29. M. S. Abdelouahab, N. E. Hamri, The grünwald-letnikov fractional-order derivative with fixed memory length, *Mediterr. J. Math.*, **13** (2016), 557–572. https://doi.org/10.1007/s00009-015-0525-3
30. A. Carpinteri, F. Mainardi, *Fractals and fractional calculus in continuum mechanics*, Vol. 378, Springer, 2014.
31. K. M. Owolabi, B. Karaagac, Chaotic and spatiotemporal oscillations in fractional reaction-diffusion system, *Chaos Solitons Fract.*, **141** (2020), 110302. https://doi.org/10.1016/j.chaos.2020.110302
32. L. F. Ávalos-Ruiz, J. F. Gomez-Aguilar, A. Atangana, K. M. Owolabi, On the dynamics of fractional maps with power-law, exponential decay and Mittag-Leffler memory, *Chaos, Solitons Fract.*, **127** (2019), 364–388. https://doi.org/10.1016/j.chaos.2019.07.010
33. K. M. Owolabi, J. F. Gómez-Aguilar, G. Fernández-Anaya, J. E. Lavín-Delgado, E. Hernández-Castillo, Modelling of chaotic processes with caputo fractional order derivative, *Entropy*, **22** (2020), 1027. https://doi.org/10.3390/e22091027
34. K. M. Owolabi, Computational techniques for highly oscillatory and chaotic wave problems with fractional-order operator, *Eur. Phys. J. Plus*, **135** (2020), 1–23. https://doi.org/10.1140/epjp/s13360-020-00873-z

35. K. M. Owolabi, Robust synchronization of chaotic fractional-order systems with shifted chebyshev spectral collocation method, *J. Appl. Anal.*, **27**, 2021. https://doi.org/10.1515/jaa-2021-2053

36. C. Li, C. Tao, On the fractional adams method, *Comput. Math. Appl.*, **58** (2009), 1573–1588. https://doi.org/10.1016/j.camwa.2009.07.050

37. R. Scherer, S. L. Kalla, Y. Tang, J. Huang, The Grünwald-Letnikov method for fractional differential equations, *Comput. Math. Appl.*, **62** (2011), 902–917. https://doi.org/10.1016/j.camwa.2011.03.054

38. I. Petráš, *Fractional-order nonlinear systems: Modeling, analysis and simulation*, Springer Science & Business Media, 2011.

39. S. Pooseh, R. Almeida, D. F. M. Torres, Discrete direct methods in the fractional calculus of variations, *Comput. Math. Appl.*, **66** (2013), 668–676. https://doi.org/10.1016/j.camwa.2013.01.045

40. P. A. Cook, *Nonlinear dynamical systems*, 2 Eds., Prentice Hall International (UK) Ltd., GBR, 1994.

41. M. H. Arshad, M. Kassas, A. E. Hussein, M. A. Abido, A simple technique for studying chaos using jerk equation with discrete time sine map, *Appl. Sci.*, **11** (2021), 437. https://doi.org/10.3390/app11010437

42. L. Chen, W. Pan, J. A. T. Machado, A. M. Lopes, R. Wu, Y. He, Design of fractional-order hyper-chaotic multi-scroll systems based on hysteresis series, *Eur. Phys. J. Spec. Top.*, **226** (2017), 3775–3789. https://doi.org/10.1140/epjst/e2018-00012-8

43. M. Z. De la Hoz, L. Acho, Y. Vidal, A modified chua chaotic oscillator and its application to secure communications, *Appl. Math. Comput.*, **247** (2014), 712–722. https://doi.org/10.1016/j.amc.2014.09.031

44. J. Theiler, Estimating fractal dimension, *JOSA A*, **7** (1990), 1055–1073. https://doi.org/10.1364/JOSAA.7.001055

45. S. Haykin, S. Puthusserypady, Chaotic dynamics of sea clutter, *Chaos: Interdisc. J. Nonlinear Sci.*, **7** (1997), 777–802. https://doi.org/10.1063/1.166275

46. K. E. Chlouverakis, Color maps of the kaplan–yorke dimension in optically driven lasers: Maximizing the dimension and almost-hamiltonian chaos, *Int. J. Bifurcat. Chaos*, **15** (2005), 3011–3021. https://doi.org/10.1142/S0218127405013848

47. C. R. Suribabu, Differential evolution algorithm for optimal design of water distribution networks, *J. Hydroinform.*, **12** (2010), 66–82. https://doi.org/10.2166/hydro.2010.014

48. G. G. Wang, A. H. Gandomi, X. S. Yang, A. H. Alavi, A novel improved accelerated particle swarm optimization algorithm for global numerical optimization, *Eng. Comput.*, **31** (2014).

49. M. S. Tavazoei, M. Haeri, A proof for non existence of periodic solutions in time invariant fractional order systems, *Automatica*, **45** (2009), 1886–1890. https://doi.org/10.1016/j.automatica.2009.04.001

50. E. Kaslik, S. Sivasundaram, Non-existence of periodic solutions in fractional-order dynamical systems and a remarkable difference between integer and fractional-order derivatives of periodic functions, *Nonlinear Anal.: Real World Appl.*, **13** (2012), 1489–1497. https://doi.org/10.1016/j.nonrwa.2011.11.013

51. C. Li, J. C. Sprott, Coexisting hidden attractors in a 4-d simplified lorenz system, *Int. J. Bifurcat. Chaos*, **24** (2014), 1450034. https://doi.org/10.1142/S0218127414500345

52. M. Wang, X. Liao, Y. Deng, Z. Li, Y. Su, Y. Zeng, Dynamics, synchronization and circuit implementation of a simple fractional-order chaotic system with hidden attractors, *Chaos, Solitons Fract.*, **130** (2020), 109406. https://doi.org/10.1016/j.chaos.2019.109406

53. Numba: A high performance python compiler, 2021. Available from: `http://numba.pydata.org/`.

54. N. Watkinson, P. Tai, A. Nicolau, A. Veidenbaum, Numbasummarizer: A python library for simplified vectorization reports, In: *2020 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, IEEE, 2020, 1–7. https://doi.org/10.1109/IPDPSW50202.2020.00058