Applied Mathematics and Computation 428 (2022) 127207

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

Effective computational discretization scheme for nonlinear dynamical systems

Priscila F.S. Guedes[a], Eduardo M.A.M. Mendes[b], Erivelton Nepomuceno[c,*]

[a] *Graduate Program in Electrical Engineering, Federal University of Minas Gerais, Brazil*
[b] *Department of Electronic Engineering, Federal University of Minas Gerais, Brazil*
[c] *Centre for Ocean Energy Research, Department of Electronic Engineering, Maynooth University, Ireland*

ARTICLE INFO

Article history:
Received 1 December 2021
Revised 20 April 2022
Accepted 24 April 2022
Available online 5 May 2022

Keywords:
Chaos
Nonlinear dynamics
Computer simulation
Computer arithmetic
Observability
Green algorithms

ABSTRACT

Numerical methods are essential to investigate and apply nonlinear continuous-time dynamical systems in many fields of science and engineering and discretization schemes are required to obtain the solutions of such dynamical systems. Although computing power has been speedily growing in recent decades, embedded and large-scale problems have motivated significant research to improve the computational efficiency. Nevertheless, few studies have focused on finite precision limitation on discretization schemes due to round-off effects in floating-point number representation. In this paper, a computational effective discretization scheme for nonlinear dynamical systems is introduced. By means of a theorem, it is shown that high-order terms in the Runge-Kutta method can be neglected with no accuracy loss. The proposed approach is illustrated using three well-known systems, namely the Rössler systems, the Lorenz equations and the Sprott B system. The number of mathematical operations and simulation time have reduced up to 81.1% and 90.7%, respectively. Furthermore, as the step-size decreases, the number of neglected terms increases due to the precision of the computer. Yet, accuracy, observability of dynamical systems and the largest Lyapunov are preserved. The adapted scheme is effective, reliable and suitable for embedded and large-scale applications.

1. Introduction

Numerical methods are of foremost importance for the study of nonlinear dynamic systems [8,10,26,41,50]. Since the renowned paper of Lorenz [25], numerical computations have played a central part to display and analyse solutions of nonlinear dynamical systems [26]. From bifurcation diagrams to the discovery of hidden attractors, a great sort of analyses have been carried out using digital computers. Additionally, application of nonlinear dynamical systems has been dramatically increased by exploring embedded platforms. For instance, chaotic systems have been successfully applied in optimization techniques [24], encryption schemes [35] and pseudo-random number generators [46].

Many nonlinear dynamic systems are described by continuous differential equations and it is necessary to study and apply methods to simulate the equations that represent such systems in order to understand their behaviour [37,42]. It is usually advisable to rewrite a set of differential equations with continuous dependence over time by a set of difference

[*] Corresponding author.
E-mail addresses: pri12_guedes@hotmail.com (P.F.S. Guedes), emmendes@cpdee.ufmg.br (E.M.A.M. Mendes), erivelton.nepomuceno@mu.ie (E. Nepomuceno).

https://doi.org/10.1016/j.amc.2022.127207

equations with a discrete time variable [22] to simulate continuous equations. Therefore, it is necessary to use discretization schemes, which have been extensively used in the literature such as the methods of Euler, Heunn and Runge-Kutta [5,38] which are based on the truncation of the Taylor series and therefore obtaining an approximation for the system under investigation. Other methods use information from the previous steps to obtain higher order approximations, such as the Adams Bashforth and Adams Bashforth Moulton methods [5,44]. There are also nonstandard techniques such as the Monaco and Normand-Cyrot method, which is based on representing the solution of the system in terms of exponential Lie expansion [28]; the Mickens method [27] that uses finite differences in the construction of the discrete model that represents the system, among others. Numerical methods for nonlinear dynamical systems are expected to be accurate and efficient for most, or preferable, all solutions [17]; otherwise, a technique is unreliable and hence not likely to be used in applications, even if any alternative is not as accurate and efficient [17]. In other words, a method should be robust.

Although computing power has been speedily growing in recent decades, embedded and large-scale problems have motivated significant research to improve computational efficiency of discretization schemes [23]. Dridi et al. [6] have implemented a secured chaos-based stream cipher in a FPGA-board using VHDL. The authors have shown that the obtained speed is competitive when compared with other works and the proposed technique is a suitable candidate for encrypting private data. With a crescent demand on secure for data flowing on the internet, pseudorandom number generators based on chaotic systems have received great attention. These systems should be fast enough to generate sets of pseudorandom numbers and avoid delays in communication or data storage. This feature becomes even more crucial for embedded systems, where computer power is usually more restricted. An example of application in this line can be seen in [49]. The authors applied a 5D hyperchaotic four-wing memristive system to implement a digital chaotic system in FPGA. According to those authors, experiments show that the design can be applied to various embedded password applications. Concern on computational cost does not only matter for embedded systems. Large-scale systems, such as the Internet of Things (IoT), poses a great challenge to simulate the dynamics of all entities and actors, which can be seen as dynamical systems [4]. A significant number of works to tackle these problems are related to innovative discretization schemes [28] or the use of parallel computer clusters [23].

There is no doubt that great progress has been observed in the simulation of nonlinear dynamical systems. However, concerns on finite precision effects are still present in literature [8,26]. Computers have properties and characteristics that result in non-exact numerical simulations. Nepomuceno [31] shows that the logistic map simulation may not converge to the exact fixed point due to the limitations of the computer. This limitation comes from the finite representation of real numbers, since computers are not able to represent all real numbers. If an input number is represented, after some mathematical operations, the result may no longer be representative, then the computer rounds each computational step, accumulating the error in the result [8]. Similar results can be found in [34], where rounding errors are investigated in the simulations but using a method based on interval extension. Nepomuceno et al. [32] used the lower bound error to simulate a chaotic system with lower and upper bounds. It has been shown that the widths of these bounds do not diverge, which is an advantage compared with other techniques based on arithmetic intervals [29]. Several works have been dedicated to the investigation of the effects of the finite precision of the computer [7,8,12,14,26,31,33,48].

Investigation on discretization schemes has received significant attention. Nevertheless, effects of computer finite precision onto discretization schemes for chaotic systems have received less attention. The normal practice is to implement the whole equation provided by a discretization scheme. Notwithstanding, round-off effects due in floating-point numbers have not yet changed the implementation of discretization schemes. In this paper, by means of a theorem, it is shown that high-order monomials in the Runge-Kutta of fourth-order (RK4) method can be excluded without accuracy loss. The proposed approach is illustrated with three systems: Rössler [40], Lorenz [25] and Sprott (B) [43]. Comparison between the number of operations is carried out showing that a reduction of up to 81.1% was achieved. The quality of simulation is validated by means of observability of dynamical systems, the plot of the attractors and the computation of the largest Lyapunov exponent.

The rest of the paper is organised as follows. In Section 2, a background for the manuscript is presented. A brief description on numerical computing, fourth-order Runge-Kutta discretization method and observability are reviewed. The proposed method based on exclusion of terms due to finite computer precision is presented in Section 3. To illustrate this approach, examples using the well-known Rössler equations, Lorenz equations and Sprott B are given in Section 4. Section 6 presents the conclusions.

2. Background

In this section, basic concepts of numerical computing, fourth-order Runge-Kutta discretization method and observability of dynamical systems are briefly described.

2.1. Numerical computing

The IEEE 754-2019 floating-point standard [13] aims at ensuring standards for the representation and operations with real numbers on computers. This standard specifies two essential formats: single format and double format. The investigation done here is based on the double format, whose bit layout is shown in Table 1.

Table 1
Bits arrangement according to the IEEE standard for 64-bit systems.

System	Double
Signal (±)	1 bit
Exponent (E)	11 bits
Mantissa (S)	52 bits

The precision of a floating-point system is associated with the number of bits present by the format, given by the number of bits of the mantissa, including the hidden bit [36]. A floating-point with precision ρ can be expressed by:

$$x = \pm(1.b_1b_2\dots b_{\rho-2}b_{\rho-1})_2 \times 2^E. \tag{1}$$

Using this representation, the following definition about precision is presented.

Definition 1. Precision (ρ) denotes the number of bits of the mantissa. The double precision ($\rho = 53$) corresponds to approximately $\rho_{10} = \log_{10}(2^{53}) \approx 16$ decimal digits [36].

2.2. Fourth-order Runge-Kutta

One of the most used and well-known methods of discretization is the fourth-order Runge-Kutta (RK4) [38,39,45]. Consider the boundary value problem determined by:

$$\dot{x} = f(t, x), \quad x(t_0) = x_0, \tag{2}$$

where x is some state variable. Let integration step $h > 0$, then RK4 can be expressed by Butcher and Goodwin [5]:

$$x_{k+1} = x_k + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4), \tag{3}$$

where

$$\begin{aligned} k_1 &= f_k, \\ k_2 &= f(t_k + \frac{h}{2}, x_k + \frac{h}{2}k_1), \\ k_3 &= f(t_k + \frac{h}{2}, x_k + \frac{h}{2}k_2), \\ k_4 &= f(t_{k+1}, x_k + hk_3), \end{aligned} \tag{4}$$

and f_k is the differential equation.

In order to apply for nonlinear dynamical systems, such as the Lorenz, the set of equations in (4) is rewritten as

$$\begin{aligned} k_1 &= f_k, \\ k_2 &= f(x_k + \frac{h}{2}k_1), \\ k_3 &= f(x_k + \frac{h}{2}k_2), \\ k_4 &= f(x_k + hk_3). \end{aligned} \tag{5}$$

Note that Eq. (5) does not depend on time explicitly (autonomous systems) and therefore a discrete model can be directly written from the continuous counterpart. With that and instead of applying the Runge-Kutta method as it is, one can use a specific discrete model for obtaining the numerical solutions (See Example 1). The advantage of this approach is that all tools available for analysing the discrete (nonlinear) models are readily available for the analysis to be performed in the next sections.

Example 1. Let the following nonlinear system represented by

$$\begin{cases} \dot{x} = y \\ \dot{y} = x^2 + y \end{cases} \tag{6}$$

Applying Eqs. (3) to (6) yields to the following nonlinear discrete Runge-Kutta model

$$\begin{cases} x_{k+1} = \frac{\left(x_k^2+y_k\right)^2 h^6}{96} + \frac{\left(4x_k^2 y_k+6y_k^2\right)h^5}{96} + \frac{\left(8y_k^2+(16x_k+4)y_k+8x_k^3+4x_k^2\right)h^4}{96} \\ \quad + \frac{\left((32x_k+16)y_k+16x_k^2\right)h^3}{96} + \frac{\left(48x_k^2+48y_k\right)h^2}{96} + hy_k + x_k \\ y_{k+1} = \frac{h^9 y_k^4}{384} + \frac{\left((8x_k+4)y_k^3+4x_k^2 y_k^2\right)h^8}{384} \\ \quad + \frac{\left(8y_k^3+\left(24x_k^2+16x_k+4\right)y_k^2+\left(16x_k^3+8x_k^2\right)y_k+4x_k^4\right)h^7}{384} \\ \quad + \frac{\left(16y_k^3+(32x_k+20)y_k^2+\left(32x_k^3+40x_k^2\right)y_k+20x_k^4\right)h^6}{384} \\ \quad + \frac{\left((80x_k+80)y_k^2+96x_k^2 y_k+24x_k^4\right)h^5}{384} \\ \quad + \frac{\left(128y_k^2+\left(160x_k^2+96x_k+16\right)y_k+64x_k^3+16x_k^2\right)h^4}{384} \\ \quad + \frac{\left(128y_k^2+(256x_k+64)y_k+128x_k^3+64x_k^2\right)h^3}{384} \\ \quad + \frac{\left((384x_k+192)y_k+192x_k^2\right)h^2}{384} + \frac{\left(384x_k^2+384y_k\right)h}{384} + y_k \end{cases} \tag{7}$$

By iterating Eq. (7), the Runge-Kutta solution of the system in Eq. (6) is obtained. Note that the discrete model is rather complex when compared to the continuous counterpart. That will be the case in all examples shown here.

2.3. Observability of dynamical systems

In this paper, our examples focus on systems with three dimensions, then the concept of observability will be developed for these systems.

Let be the dynamical system

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}), \quad r = g(\mathbf{x}), \tag{8}$$

where $\mathbf{x} \in \mathbb{R}^3$ is the state vector, $\mathbf{f}$ is nonlinear vector field and r is the observable acquired through the measurement function $g : \mathbb{R}^3 \mapsto \mathbb{R}$.

The portrait that is reproduced may be spanned by the derivative coordinates, such as

$$X = r, \quad Y = \dot{r}, \quad W = \ddot{r}. \tag{9}$$

It is possible to define a coordinate transformation Φ, when original states (x, y, w) and derivative coordinates (X, Y, W) are related. If $r = x$, then the transformation Φ is equivalent to

$$X = r, \quad Y = f_r, \quad W = \frac{\partial f_r}{\partial x} f_x + \frac{\partial f_r}{\partial y} f_y + \frac{\partial f_r}{\partial w} f_w, \tag{10}$$

where f_x, f_y, and f_w are the components of $\mathbf{f}$ [19,20].

The system can be rewritten as an explicit system

$$\dot{X} = Y, \quad \dot{Y} = W, \quad \dot{W} = F_r(X, Y, W), \tag{11}$$

where $F_r(X, Y, W)$ is the jerk equation [9,18,19]. For a given original system, the jerk equation F_r can be analytically derived using the coordinate transformation Φ [21].

3. Methodology

In this section, the proposed technique is described. First, we present the following definition for monomial:

Definition 2. Monomial is an algebraic expression formed by a real number, or a variable, or by a multiplication of numbers (coefficients) and variables.

Example 2. The algebraic expressions in (12)–(15) are examples of monomials

$$2, \tag{12}$$

$$4.7 \times 10^{-3}, \tag{13}$$

$$5x^2 y, \tag{14}$$

$$3.2 \times 10^2 xyz. \tag{15}$$

Addition and subtraction of numbers in scientific notation can only be performed when there are equal exponents. The monomial as in (13) can be represented by mantissa ($S = 4.7$), base ($B = 10$) and exponent ($E = -3$).

Example 3. Consider three monomials,

$$x_1 = -1.2 \times 10^2, \tag{16}$$

$$x_2 = 2.5 \times 10^3, \tag{17}$$

$$x_3 = 3.3 \times 10^{-1}. \tag{18}$$

To add them, it is necessary to convert to the same exponent. In this case, the conversion is performed for the highest value exponent (E = 3).

$$\begin{aligned} S &= x_1 + x_2 + x_3 \\ &= -0.12 \times 10^3 + 2.5 \times 10^3 + 0.00033 \times 10^3 \\ &= (-0.12 + 2.5 + 0.00033) \times 10^3 \\ &= 2.38033 \times 10^3. \end{aligned}$$

The key point of this work is to analyse monomials of systems discretized by the fourth-order Runge-Kutta method according to the rules of rounding-off of the IEEE-754-2019. As some monomials can be of high-order, it can be noticed that round-off operation may turn some of these monomials negligible. Thus, it is possible to safely determine that such monomials may be excluded or not. The result of excluding monomials, considering the double precision format, is presented in the following theorem.

Theorem 1. *Let γ be the set of monomials, that is, $\gamma = \{\alpha_1 10^{\beta_1}, \alpha_2 10^{\beta_2}, \ldots, \alpha_n 10^{\beta_n}\}$, with $1 \leq \alpha_n \leq 9$ and with $\beta_n > \beta_{n-1} > \beta_{n-2} > \ldots > \beta_1$. Let Ω be the set of difference between β_n and other exponents, that is, $\Omega = \{\Omega_1, \Omega_2, \ldots, \Omega_{n-1}\} = \{(\beta_n - \beta_1), (\beta_n - \beta_2), \ldots, (\beta_n - \beta_{n-1})\}$. If $\Omega_i > \rho$, then the monomial γ_i may be excluded in the implementation of the discretization scheme.*

Proof. Definition 1 shows that a number represented in the decimal base for double precision is approximately $\rho \approx 16$ decimal digits. Then, if the necessary adjustment to perform the sum (or subtraction) is greater than 16, the number will be represented by zeros multiplied by 10^β, that is, $0.0000000000000000 \times 10^\beta$, which confirms its exclusion. This rationale can be applied for any number of bits or decimal digits. □

Example 4. Consider the following equation

$$\begin{aligned} X &= 0.05 + 6.5104 \times 10^{-20} - 3.90625 \times 10^{-10} \\ &= 5 \times 10^{-2} + 6.5104 \times 10^{-20} - 3.90625 \times 10^{-10} \end{aligned}$$

According to Theorem 1, the set of monomials are as follows:

$$\begin{aligned} \gamma_1 &= \alpha_1 10^{\beta_1} = 6.5104 \times 10^{-20} \\ \gamma_2 &= \alpha_2 10^{\beta_2} = -3.90625 \times 10^{-10} \\ \gamma_3 &= \alpha_3 10^{\beta_3} = 5.0 \times 10^{-2}. \end{aligned}$$

The set Ω is given by

$$\begin{aligned} \Omega_1 &= \beta_3 - \beta_1 = -2 - (-20) = 18 \\ \Omega_2 &= \beta_3 - \beta_2 = -2 - (-10) = 8. \end{aligned}$$

Since $\Omega_1 > 16$ (for double precision), γ_1 may be excluded without loss of accuracy. This is verified in Eq. (19). Figure 1a and b present this operation using Matlab for long and hexadecimal format, respectively. As can seen, the hexadecimal remains the same after the exclusion of the monomial.

$$\begin{aligned} X &= 5 \times 10^{-2} + 0.0000000000000000 \times 10^{-2} \\ &\quad -0.0000000390625 \times 10^{-2} \\ &= 4.9999999609375 \times 10^{-2} \end{aligned} \tag{19}$$

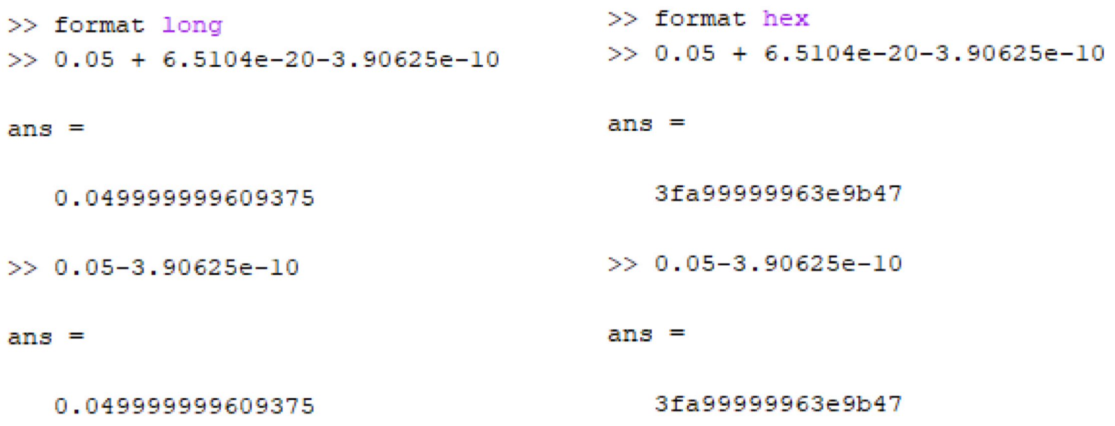
To show what happens when there are monomials with the same exponent value, an example is now given (Example 5).

Example 5. Consider the following equation

$$\begin{aligned} X &= 0.04 + 6.5 \times 10^{-20} - 3.9 \times 10^{-10} + 0.02 \\ &= 5 \times 10^{-2} + 6.5 \times 10^{-20} - 3.9 \times 10^{-10} + 2 \times 10^{-2} \end{aligned}$$

According to Theorem 1, the set of monomials are as follows:

$$\gamma_1 = \alpha_1 10^{\beta_1} = 6.5 \times 10^{-20}$$



```
>> format long
>> 0.05 + 6.5104e-20-3.90625e-10

ans =

   0.049999999609375

>> 0.05-3.90625e-10

ans =

   0.049999999609375
```

```
>> format hex
>> 0.05 + 6.5104e-20-3.90625e-10

ans =

   3fa99999963e9b47

>> 0.05-3.90625e-10

ans =

   3fa99999963e9b47
```

(a) Format Long

(b) Format Hexadecimal

Fig. 1. Example 4 was run on Matlab$^{\text{TM}}$. The result is presented in long and hexadecimal formats, as described in Eq. (19). The exclusion of monomial $\gamma_1 = 6.5104 \times 10^{-20}$ does not change the final result.

$$\gamma_2 = \alpha_2 10^{\beta_2} = -3.9 \times 10^{-10}$$
$$\gamma_3 = \alpha_3 10^{\beta_3} = 2.0 \times 10^{-2}$$
$$\gamma_4 = \alpha_4 10^{\beta_4} = 5.0 \times 10^{-2}.$$

The set Ω is given by

$$\Omega_1 = \beta_4 - \beta_1 = -2 - (-20) = 18$$
$$\Omega_2 = \beta_4 - \beta_2 = -2 - (-10) = 8$$
$$\Omega_3 = \beta_4 - \beta_3 = -2 - (-2) = 0.$$

Since $\Omega_1 > 16$ (for double precision), γ_1 may be excluded without loss of accuracy. The monomial represented by Ω_3 will be kept in the representation, since only monomials where $\Omega > 16$ will be excluded.

4. Numerical experiments

In this section, the proposed approach is illustrated for three systems: the Rössler system [40], the Lorenz equations [25] and Sprott B [43]. For each system, the reduced RK4, here named as RRK4, according to Theorem 1 is calculated. The quality of RRK4 is evaluated by means of the observability of the dynamical systems, plot of the projections on the xy plane of the discretized systems and computation of the largest Lyapunov exponent. The step size h is chosen according to the usual values found in the literature.

4.1. The Rössler system

Let the Rössler system be represented by Rössler [40]:

$$\begin{cases} \dot{x} = -y - z, \\ \dot{y} = x + ay, \\ \dot{z} = b + z(x - c), \end{cases} \tag{20}$$

where bifurcation parameters are described by (a, b, c). Rössler equations were discretized by the Runge-Kutta scheme of fourth-order. Theorem 1 was applied considering initial condition $(x_0, y_0, z_0) = (-1.0, 1.0, 1.0)$, parameters $a = 0.15$, $b = 0.20$ and $c = 10.0$, and step-size of 10^{-2}. The number of monomials for each equation of the Rössler system is shown in Table 2. RK4 represents the discretization performed by the fourth-order Runge-Kutta method, and RRK4 represents the fourth-order Runge-Kutta method when Theorem 1 is applied and some terms are excluded.

To investigate the observability of each dynamical variable, it is necessary to begin with a measurement component so that $r = g(x, y, z) = y$. The coordinate transformation Φ_y peruses at that point as

$$\Phi_y = \begin{cases} X = y, \\ Y = ay + x, \\ Z = -y - z + ax + a^2 y, \end{cases} \tag{21}$$

Table 2
Number of monomials for each of the discretized equations for the systems Lorenz, Rössler and Sprott B. The comparison is made between conventional RK4 and Reduced RK4 (RRK4). Initial conditions, parameters and step size for each system are as follows: 1) Rössler:$(x_0, y_0, z_0) = (-1.0, 1.0, 1.0)$, $a = 0.15$, $b = 0.20$ and $c = 10.0$, and step-size of 10^{-2}; 2) Lorenz: $(x_0, y_0, z_0) = (1.0, 0.5, 0.9)$, $\sigma = 16.0$, $\beta = 4.0$, $\rho = 45.92$, $h = 10^{-3}$; 3) Sprott B: $(x_0, y_0, z_0) = (0.05, 0.05, 0.05)$ and $h = 10^{-2}$. The highest reduction in the number of monomials occurs for the Lorenz equations, where the total is decreased in 75.5%.

Equations	Rössler		Lorenz		Sprott B	
	RK4	RRK4	RK4	RRK4	RK4	RRK4
x_{k+1}	117	101	116	79	175	41
y_{k+1}	38	38	886	192	38	26
z_{k+1}	903	290	963	210	167	37

and the equivalent jerk equation F_y is

$$\begin{aligned} F_y &= -b - cX + (ac-1)Y + (a-c)Z - aX^2 \\ &\quad + (a^2+1)XY - aXZ - aY^2 + YZ. \end{aligned} \tag{22}$$

When the observable is variable x of the Rössler system, the coordinate transformation Φ_x is as follows

$$\Phi_x = \begin{cases} X = x, \\ Y = -y - z, \\ Z = -x - ay - z(x-c) - b, \end{cases} \tag{23}$$

and the corresponding jerk equation F_x is

$$\begin{aligned} F_x &= ab - cX + X^2 - aXY + acY + XZ + (a-c)Z \\ &\quad - \frac{(a+c+Z-aY+b)}{a+c-X}. \end{aligned} \tag{24}$$

For variable z of the Rössler system as the observable, the coordinate transformation Φ_z is as follows

$$\Phi_z = \begin{cases} X = z, \\ Y = z(x-c) + b, \\ Z = (b + z(x-c))(x-c) + z(-y-z), \end{cases} \tag{25}$$

and the associated jerk equation is

$$\begin{aligned} F_z &= b - cX - Y + aZ + aX^2 - XY \\ &\quad + \frac{(ab+3Z)Y - aY^2 - bZ}{Z} + \frac{2bY^2 - 2Y^3}{X^2}. \end{aligned} \tag{26}$$

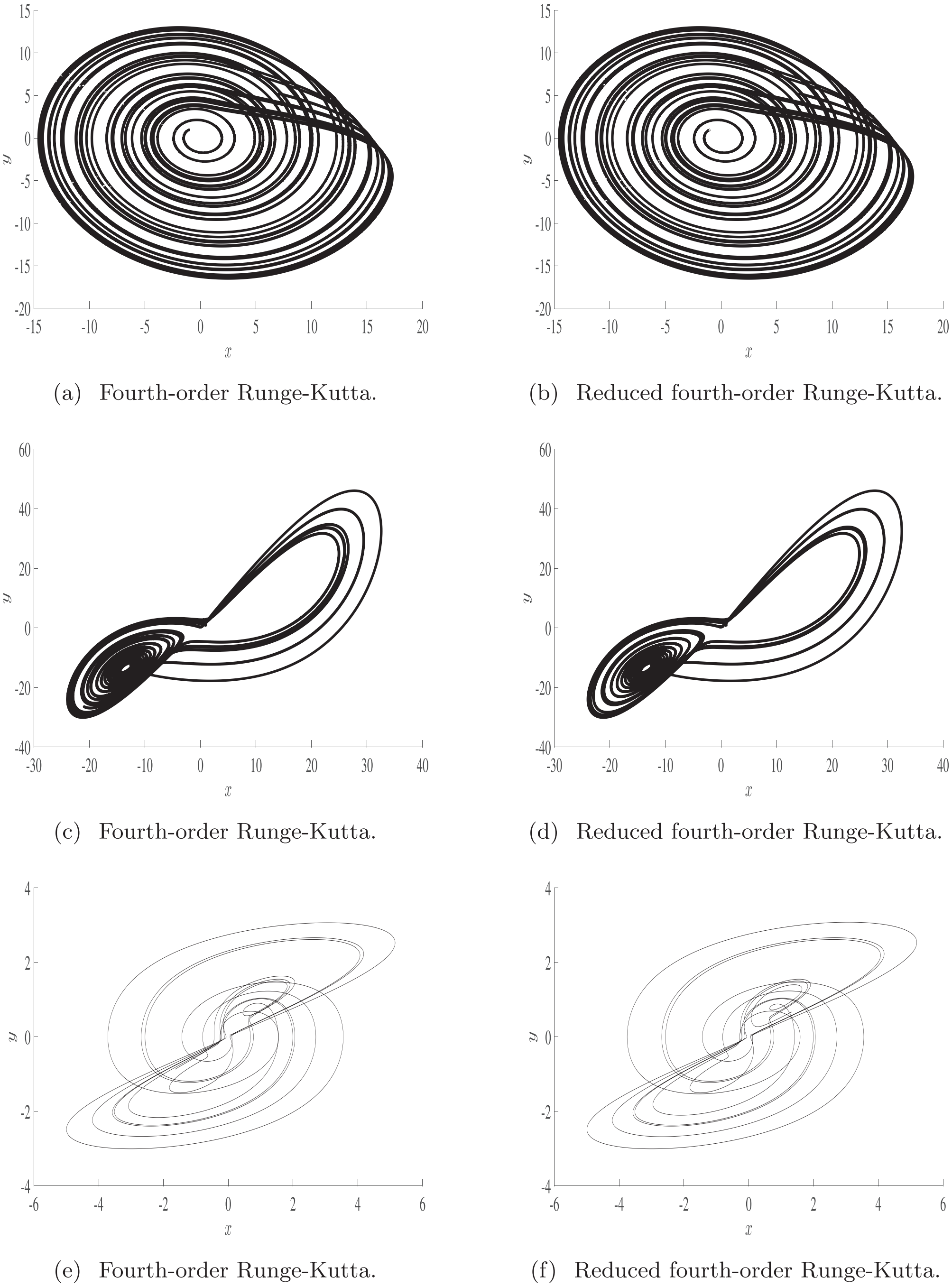
Analysing Eqs. (22), (24) and (26) it is possible to state that Eq. (26) is more complex than Eq. (24) and therefore is more complex than Eq. (22). Similar to [18], we consider complexity associated with the number of monomials included in the jerk equation, order of the nonlinearities and poles. Note that the jerk equation indicates the connections between the states of the original system “seen from one observable point of view”.

Based on the complexity of F_x, F_y e F_z, it can be said that variable y is more observable than variable x, which is more observable than variable z, that is, $y \triangleright x \triangleright z$. From the results presented in Table 2, it is conceivable to evaluate the observability and compare it with what has already been exposed before. Variable y is more observable than variable x, which is more observable than the variable z. Since variable z is the one with the greatest exclusion of terms, it does not contribute much to the description of the system. This statement corroborates the observability analysis carried out previously.

Figures 2a and b show the projections on the xy plane of the Rössler equations for the fourth-order Runge-Kutta method and the reduced fourth-order Runge-Kutta method. The trajectories exhibit very close values in both cases, so Fig. 2a and b are practically the same, illustration Theorem 1.

As Fig. 2a and b appear to be the same; it is difficult to visualise the small difference, let alone compare them. To mitigate this problem, Fig. 3a shows the difference between RK4 and RRK4 by using a logarithmic scale. Note that RK4 and RRK4 represent two interval extensions of the same system. The distance between these two interval extensions is the lower bound error [30]. And for chaotic systems, the distance is exponentially divergent.

As a final criterion, the largest Lyapunov exponent was calculated using the method proposed by Nepomuceno and Mendes [34] for both RK4 and RRK4, as shown in Table 3. This method uses the interval extensions to calculate the lower bound error which is a measure of the distance between the simulated dynamical systems (or pseudo-orbit) and the real orbit. When a system behaves chaotically the distance between these two orbits is exponentially divergent, and therefore a



(a) Fourth-order Runge-Kutta.

(b) Reduced fourth-order Runge-Kutta.

(c) Fourth-order Runge-Kutta.

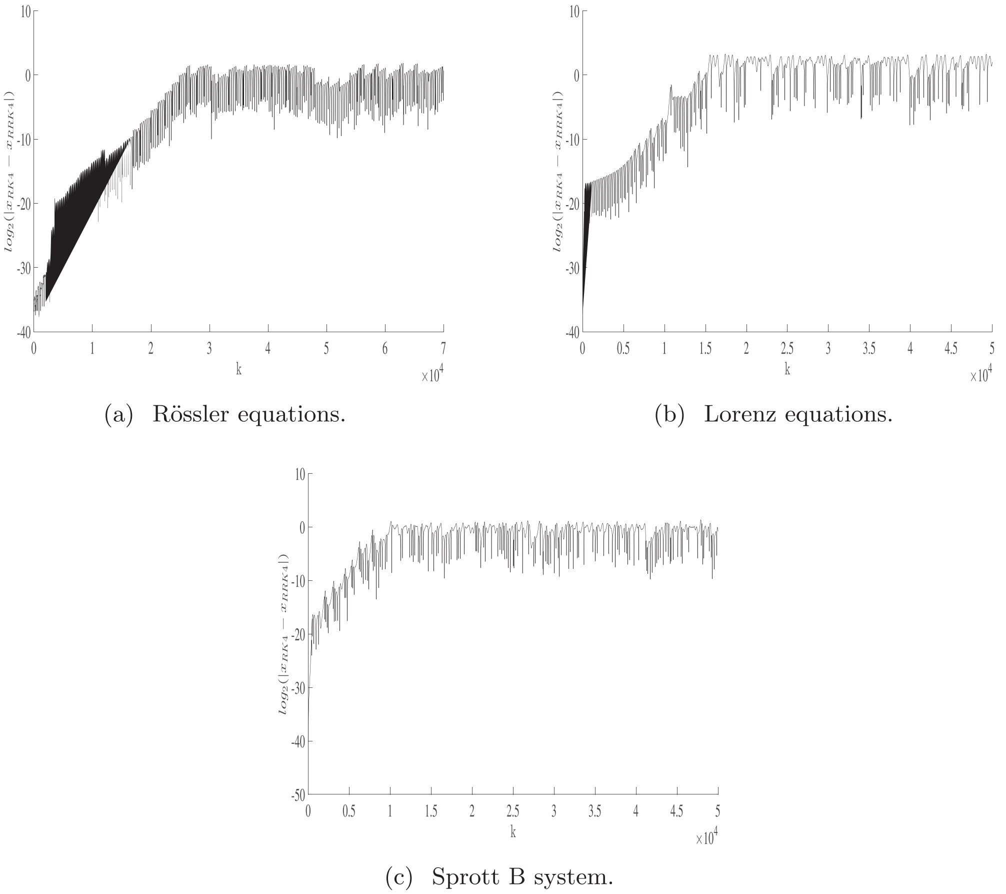
(d) Reduced fourth-order Runge-Kutta.

(e) Fourth-order Runge-Kutta.

(f) Reduced fourth-order Runge-Kutta.

Fig. 2. Projections on the *xy* plane of the attractors of the discretized systems. (a) and (b): Projections on the *xy* plane of the discretized Rössler equations, with initial condition $(x_0, y_0, z_0) = (-1.0, 1.0, 1.0)$, parameters $a = 0.15$, $b = 0.20$ and $c = 10.0$, and step-size $h = 10^{-2}$. (c) and (d): Projections on the *xy* plane of the discretized Lorenz equations, with initial condition $(x_0, y_0, z_0) = (1.0, 0.5, 0.9)$, parameters $\sigma = 16.0$, $\beta = 4.0$ and $\rho = 45.92$, and step-size $h = 10^{-3}$. (e) and (f): Projections on the *xy* plane of the Sprott B, with initial condition $(x_0, y_0, z_0) = (0.05, 0.05, 0.05)$ and step-size $h = 10^{-2}$.

slope in a logarithm plot of the lower bound error captures the divergence and quantifies it as a number which is the positive Lyapunov exponent. The positive exponent was found to be 0.0897 and 0.0909 for RK4 and RRK4, respectively, which is in a good agreement with the literature value of 0.090 nat/iter [47]. Table 4 shows the number of points needed to calculate the Lyapunov exponent using the method proposed in [34]. It is possible to observe that RK4 uses a slightly larger number for this estimation.



(a) Rössler equations.

(b) Lorenz equations.

(c) Sprott B system.

Fig. 3. Exponentially growing of the difference between the two pseudo-orbits RK4 and RRK4. (a): Difference between the two pseudo-orbits of the discretized Rössler equations, with initial condition $(x_0, y_0, z_0) = (-1.0, 1.0, 1.0)$, parameters $a = 0.15$, $b = 0.20$ and $c = 10.0$, and step-size $h = 10^{-2}$. (b): Difference between the two pseudo-orbits of the discretized Lorenz equations, with initial condition $(x_0, y_0, z_0) = (1.0, 0.5, 0.9)$, parameters $\sigma = 16.0$, $\beta = 4.0$ and $\rho = 45.92$, and step-size $h = 10^{-3}$. (c): Difference between the two pseudo-orbits of the Sprott B, with initial condition $(x_0, y_0, z_0) = (0.05, 0.05, 0.05)$ and step-size $h = 10^{-2}$.

Table 3
Calculation of Lyapunov exponent. The expected values are obtained in [47] for Rössler and Lorenz equations and in [43] for Sprott B. The unit of the Lyapunov exponent is indicated in nat/iter.

System	Literature λ	Calculated λ for RK4	Calculated λ for reduced RK4
Rössler	0.0900	0.0897	0.0909
Lorenz	1.5000	1.4702	1.4778
Sprott B	0.2100	0.1710	0.1854

Table 4
Number of points necessary to estimate the Lyapunov exponent for each system. The following parameters were considered for each system: 1) Rössler: $(x_0, y_0, z_0) = (-1.0, 1.0, 1.0)$, $a = 0.15$, $b = 0.20$ and $c = 10.0$, and step-size of 10^{-2}; 2) Lorenz: $(x_0, y_0, z_0) = (1.0, 0.5, 0.9)$, $\sigma = 16.0$, $\beta = 4.0$, $\rho = 45.92$, $h = 10^{-3}$; 3) Sprott B: $(x_0, y_0, z_0) = (0.05, 0.05, 0.05)$ and $h = 10^{-2}$.

System λ	RK4	reduced RK4
Rössler	35218	33331
Lorenz	21276	19678
Sprott B	14893	12687

Table 5
Mean and standard deviation for the estimation of the Lyapunov exponent using 50 samples with different initial conditions. The expected values are obtained in [47] for Rössler and Lorenz equations and in [43] for Sprott B. The unit of the Lyapunov exponent is indicated in nat/iter.

System	Literature λ	Calculated λ for RK4	Calculated λ for reduced RK4
Rössler	0.0900	0.0899 ± 0.0091	0.0901 ± 0.0089
Lorenz	1.5000	1.4847 ± 0.0451	1.4877 ± 0.0487
Sprott B	0.2100	0.2183 ± 0.0229	0.2051 ± 0.0121

To emphasise the advantage in using the proposed method, the Lyapunov exponent was estimated for different initial conditions and the mean and standard deviation were calculated. For this calculation, 50 samples with different initial conditions were considered. The following parameters were considered for the Rössler system: $(x_0, y_0, z_0) = (-15.0 : -1.0, 1.0 : 15.0, 1.0 : 15.0)$, $a = 0.15$, $b = 0.20$ and $c = 10.0$, and step-size of 10^{-2}. Table 5 shows the results and it is possible to verify that they are in accordance with the literature.

4.2. Lorenz equations

Consider the Lorenz equations [25]

$$\begin{cases} \dot{x} = \sigma(y - x), \\ \dot{y} = x(\rho - z) - y, \\ \dot{z} = xy - \beta z, \end{cases} \tag{27}$$

where σ, ρ and β are parameters. Lorenz equations were discretized by the fourth-order Runge-Kutta method. To apply Theorem 1, initial condition $(x_0, y_0, z_0) = (1.0, 0.5, 0.9)$, parameters $\sigma = 16.0$, $\beta = 4.0$ and $\rho = 45.92$, and step-size of 10^{-3} were considered. The number of monomials for each equation of the Lorenz equations is shown in Table 2. As it occurred to Rössler's equations, the reduced RK4 varies as iterations of Lorenz equations occur.

The observability of each dynamic variable was investigated, and the associated model for variable x is

$$F_x = X\beta\rho\sigma - X^3\sigma - X\beta\sigma - \beta\sigma Y - X^2Y - \beta Y - \beta Z - \sigma Z + \frac{\sigma Y^2}{X} - Z + \frac{Y^2}{X} + \frac{YZ}{X} \tag{28}$$

whereas for the other coordinates the respective equations have a large number of monomials and, more importantly they are of the following implicit form

$$\dot{X} = Y, \quad \dot{Y} = Z, \quad F_r(X, Y, Z, \dot{Z}) = 0. \tag{29}$$

Equations relating to F_x, F_y, and F_z have 12, 588, and 211 monomials, respectively. Based on the complexity of the jerk equations, it can be concluded that variable x is more observable than variable z, which is more observable than variable y, that is, $x \triangleright z \triangleright y$. This is slightly different from the results in Table 2, variable x is more observable than variable y, which is more observable than variable z, that is, $x \triangleright y \triangleright z$. The result is in accordance with what is stated in [1,18] and indicates that the variable z is the least observable, based on symmetry considerations. Nonetheless the result presented in Table 2 is in accordance with the observability analysis performed previously using the jerk equations, which determines that the variable x is the most observable.

Figure 2c and d show the projections on the xy plane of the Lorenz equations. The trajectories exhibit very close values in both cases. And Fig. 3b shows the difference between RK4 and RRK4 orbits on a logarithmic scale. The Lyapunov exponent was calculated as shown in Tables 3 and 4 shows the number of points needed to calculate the Lyapunov exponent. Values obtained from RK4 and RRK4 are similar to the values found in the literature. Additionally, the Lyapunov exponent was calculated for 50 samples with different initial conditions, $(x_0, y_0, z_0) = (1.0 : 35.0, 0.5 : 35.0, 0.9 : 35.0)$, system parameters set to $\sigma = 16.0$, $\beta = 4.0$, $\rho = 45.92$ and step size $h = 10^{-3}$. The mean and standard deviation were calculated as shown in Table 5 and the estimated Lyapunov exponent is in agreement with values found in the literature.

4.3. Sprott B

Consider the Sprott B system [43]:

$$\begin{cases} \dot{x} = yz, \\ \dot{y} = x - y, \\ \dot{z} = 1 - xy. \end{cases} \tag{30}$$

The Sprott B system was discretized using the fourth-order Runge-Kutta method. In order to apply Theorem 1, $(x_0, y_0, z_0) = (0.05, 0.05, 0.05)$ and step-size of 10^{-2} were considered as indicated in [15,16,43]. The number of monomials for each equation of the Sprott B system is shown in Table 2. RRK4, as shown in Table 2, is achieved for some iterations since, as the Sprott B equations are iterated, their values change. Thus, the values of the monomials are also changed, causing the number of monomials to vary for each equation.

For the investigation of the observability of each dynamic variable, consider $r = g(x, y, z) = y$ as and coordinate transformation Φ_y such as

$$\Phi_y = \begin{cases} X = y, \\ Y = x - y, \\ Z = -x + (1 + z)y, \end{cases} \tag{31}$$

and the corresponding jerk equation F_y is

$$F_y = -Z + (1 - XY + X^2)X - (Y + Z)\frac{Y}{X}. \tag{32}$$

When the observable is variable z of the Sprott B, the coordinate transformation Φ_z reads as

$$\Phi_z = \begin{cases} X = z, \\ Y = 1 - xy, \\ Z = xy - x^2 - y^2 z, \end{cases} \tag{33}$$

and the corresponding jerk equation F_z is

$$\begin{aligned} F_z = \frac{1}{2X}\{&8X^2Y - 8X^2 - 2XY - 4XZ + Y^2 + YZ \\ &+2X - Y \pm (2X - Y)(-4XY^2 + 8XY + Y^2 \\ &+2YZ + Z^2 - 4X - 2Y - 2Z + 1)^{\frac{1}{2}}\}. \end{aligned} \tag{34}$$

The final case is to consider the variable x of the Sprott B as the observable. The coordinate transformation Φ_x reads as

$$\Phi_x = \begin{cases} X = x, \\ Y = yz, \\ Z = (x - y)z + (1 - xy)y, \end{cases} \tag{35}$$

and the associated jerk equation is

$$\begin{aligned} &F_x^3X^3 + (4X^3Y - 3X^4 + 7X^3Z - 3X^2Y^2 - 3X^2YZ - X^2 + XY)F_x^2 + (22X^3Y^2 + 27X^6Y + 9X^6Z - 18X^4Y \\ &-22X^4Z + 28X^3YZ + 15X^3Z^2 + 4X^3 - 8X^2Y^3 - 22X^2Y^2Z - 14X^2YZ^2 - 5X^2Y - Y^3 - 3X^2Z + 3XY^4 \\ &+6XY^3Z + 3XY^2Z^2 + 2XY^2 + 4XYZ - Y^2Z)F_x + 27X^9Y - 45X^7Y - 18X^7Z + 4X^6 + 31X^6Y^2 + 39X^6YZ \\ &+9X^6Z^2 - 27X^5Y^3 - 36X^5Y^2Z - 9X^5YZ^2 + 17X^5Y + 16X^5Z - 22X^4Y^2 - 42X^4YZ - 27X^4Z^2 - 4X^4 \\ &+40X^3YZ^2 + 5X^3Y + 9X^3Z^3 + 6X^3Z + 31X^3Y^3 + 60X^3Y^2Z - 19X^2Y^4 - 44X^2Y^3Z - 40X^2Y^2Z^2 - 5X^2Y^2 \\ &-15X^2YZ^3 - 9X^2YZ - 2X^2Z^2 + 4XY^5 + 15XY^4Z + 18XY^3Z^2 + 4XY^3 + 7XY^2Z^3 + 6XY^2Z - 3Y^5Z \\ &-3Y^4Z^2 + 4XYZ^2 - Y^6 - Y^3Z^3 - 3Y^3Z - 2Y^2Z^2 = 0 \end{aligned} \tag{36}$$

It can be seen that Eq. (36) is more complex than Eq. (34), just as this equation is more complex than Eq. (32). Therefore, one can guarantee that the variable y is more observable than the variable z, which is more observable than the variable x, that is, $y \triangleright z \triangleright x$. According to the results presented in Table 2, it is possible to assess the observability of the system. Variable y is the most observable since there is a lesser exclusion of terms, followed by variables z and x. All this is in conformity with the observability analysis carried out previously.

Figure 2 shows the projections on the xy plane of the Sprott B for RK4 and RRK4. Note that Fig. 2e and f appear to be the same since their pseudo-orbits exhibit very close values. This highlights the concept of monomial exclusion. Figure 3c show the small difference between RK4 and RRK4.

In addition, using both RK4 and RRK4, a Lyapunov exponent of 0.1710 nat/iter and 0.1854 nat/iter was found, respectively, as shown in Table 3, which is also in good agreement with literature. Table 4 shows the number of points needed to calculate the Lyapunov exponent. It is possible to observe again that RK4 needs a slightly larger number of iterations to estimate the exponent. Table 5 shows the mean and standard deviation of the Lyapunov exponent considering 50 samples with different initial conditions within the range $(x_0, y_0, z_0) = (0.05 : 15.0, 0.05 : 15.0, 0.05 : 15.0)$ and $h = 10^{-2}$.

4.4. Algorithm complexity analysis

In order to analyse the computational complexity of the systems under study, the mathematical operations of each system were counted, the simulation time of each system has been calculated, the first three iterations for RK4 and RRK4 are shown. Since it is known that the discretized system depends on the step-size (h), it will be shown that the number of monomials varies with the chosen step. Table 6 shows the summary of the basic operations used in the calculations. The number of operations is per iteration that outfits a computational complexity of $O(n)$.

Table 6
Summary of computational complexity. The basic operations used throughout each system were analysed, that is, Sum/Subtraction, Multiplication/Division and Power. For each system, all the operators for variables x, y, z are added and the reduction was calculated. The proposed method can reduce up to 81.1% of the required operations for the Lorenz equations.

		RK4			Reduced RK4			Reduction
	Operations	x_{k+1}	y_{k+1}	z_{k+1}	x_{k+1}	y_{k+1}	z_{k+1}	
Rössler	Sum/Subtraction	116	37	902	100	37	289	
	Multiplication/Division	527	141	4878	445	141	1269	
	Power	192	47	2240	161	47	606	
	Summation of operators	835	225	8020	706	225	2164	
	Total		9080			3095		65.9%
Lorenz	Sum/Subtraction	115	885	962	78	191	209	
	Multiplication/Division	605	5260	5739	325	877	980	
	Power	289	2881	3190	172	425	514	
	Summation of operators	1009	9026	9891	575	1493	1703	
	Total		19926			3771		81.1%
Sprott B	Sum/Subtraction	174	37	166	40	25	36	
	Multiplication/Division	726	139	710	118	64	115	
	Power	404	60	393	62	33	64	
	Summation of operators	1304	236	1269	220	122	215	
	Total		2809			557		80.2%

Table 7
Average time of a thousand attempts to execute the proposed algorithm. We have also presented one standard deviation in order to consider the intrinsic fluctuation of time consumption in a computer. 1) Rössler:$(x_0, y_0, z_0) = (-1.0, 1.0, 1.0)$, $a = 0.15$, $b = 0.20$ and $c = 10.0$, and step-size of 10^{-2}; 2) Lorenz: $(x_0, y_0, z_0) = (1.0, 0.5, 0.9)$, $\sigma = 16.0$, $\beta = 4.0$, $\rho = 45.92$, $h = 10^{-3}$; 3) Sprott B: $(x_0, y_0, z_0) = (0.05, 0.05, 0.05)$ and $h = 10^{-2}$.

System	*RK*4	*RRK*4	Reduction
Rössler	4.1729 ± 0.5054	1.4500 ± 0.0991	65.3%
Lorenz	18.7971 ± 0.1078	1.7481 ± 0.0994	90.7%
Sprott B	1.6427 ± 0.0567	0.2540 ± 0.0085	84.5%

Rössler equations present a total of 9080 operations per iteration using RK4 whereas, using RRK4, the number of operations was 3095, representing a reduction of approximately 65.9% of operations performed per iteration. The Lorenz equations presented 19926 and 3771 mathematical operations per iteration, when discretized using RK4 and RRK4, respectively. The reduction in the number of mathematical operations was 81.1%. Likewise, the Sprott B system presented a total of 2809 operations when discretized using RK4 and 557 operations using RRK4. Therefore, there was a reduction of approximately 80.2% in the number of mathematical operations when Theorem 1 was applied.

The simulation time is shown in Table 7. Each system was simulated a thousand times and the time shown in the table is the average over the outcome of the simulations. For the reduced fourth-order Runge-Kutta it is evident that there was a significant reduction in the simulation time, especially for the Lorenz equations. Since the decrease in time is directly related to the amount of operations performed, the contribution to the decrease in computational cost is worth considering. Such a decrease is specially important, as there is a desire to develop fast and light algorithms specially with ever increasing volume of data created. In addition to that, it is necessary to guarantee the security for data that need to be kept confidential. Security, combined with speed (short execution time) and low computational cost is discussed in several works [2,3,11].

Table 8 presents the three first values of each system for variable x. The Rössler equations and the Sprott B system present the same values for both RK4 and RRK4. For the case of simulation of Lorenz equations, it is possible to observe that from the second iteration on there is a divergence between the values of RK4 and of RRK4 (See Table 8).

The accumulated truncation error for RK4 is $O(h^4)$ and the error between RK4 and RRK4 for Lorenz equations is 1.0137×10^{-15} and 1.0199×10^{-15}, in the second and third iteration, respectively. That is, this error is smaller than the accumulated error validating the use of the proposed theorem, since the found error was smaller than the error of the discretization scheme.

The fourth-order Runge-Kutta discretization scheme depends on the step-size, as a consequence the proposed term exclusion method significantly depends on the chosen value of h. To illustrate this dependence, Table 9 shows the number of monomials that is considered in the simulation for each selected step-size. As expected, the number of monomials that represent the systems decreases as the step-size decreases.

Table 8
Simulation for the first three iterates: 1) Rössler:$(x_0, y_0, z_0) = (-1.0, 1.0, 1.0)$, $a = 0.15$, $b = 0.20$ and $c = 10.0$, and step-size of 10^{-2}; 2) Lorenz: $(x_0, y_0, z_0) = (1.0, 0.5, 0.9)$, $\sigma = 16.0$, $\beta = 4.0$, $\rho = 45.92$, $h = 10^{-3}$; 3) Sprott B: $(x_0, y_0, z_0) = (0.05, 0.05, 0.05)$ and $h = 10^{-2}$.

	Rössler	
k	*x RK4*	*x RRK4*
0	−1.000000000000000	−1.000000000000000
1	−1.019436103095310	−1.019436103095310
2	−1.037815068635247	−1.037815068635247
3	−1.055234737344272	−1.055234737344272
	Lorenz	
k	*x RK4*	*x RRK4*
0	1.000000000000000	1.000000000000000
1	0.992416886935038	0.992416886935038
2	0.985654757485587	0.985654757485586
3	0.979695097099713	0.979695097099712
	Sprott B	
k	*x RK4*	*x RRK4*
0	0.050000000000000	0.050000000000000
1	0.050027493774038	0.050027493774038
2	0.050059975228164	0.050059975228164
3	0.050097444650059	0.050097444650059

Table 9
Number of monomials for each of the discretized equations for the systems Lorenz, Rössler and Sprott B. The comparison is made between conventional RK4 and Reduced RK4 (RRK4) for different values of step-size. Initial conditions, parameters and step size for each system are as follows: 1) Rössler:$(x_0, y_0, z_0) = (-1.0, 1.0, 1.0)$, $a = 0.15$, $b = 0.20$ and $c = 10.0$; 2) Lorenz: $(x_0, y_0, z_0) = (1.0, 0.5, 0.9)$, $\sigma = 16.0$, $\beta = 4.0$, $\rho = 45.92$; 3) Sprott B: $(x_0, y_0, z_0) = (0.05, 0.05, 0.05)$.

			Rössler		
	RK4	RRK4	RRK4	RRK4	RRK4
		$(h = 10^{-1})$	$(h = 10^{-2})$	$(h = 10^{-3})$	$(h = 10^{-4})$
x_{k+1}	117	117	101	47	24
y_{k+1}	38	38	38	33	16
z_{k+1}	903	896	290	87	38
			Lorenz		
	RK4	RRK4	RRK4	RRK4	RRK4
		$(h = 10^{-2})$	$(h = 10^{-3})$	$(h = 10^{-4})$	$(h = 10^{-5})$
x_{k+1}	116	116	71	37	20
y_{k+1}	886	687	192	79	38
z_{k+1}	963	715	210	71	27
			Sprott B		
	RK4	RRK4	RRK4	RRK4	RRK4
		$(h = 10^{-1})$	$(h = 10^{-2})$	$(h = 10^{-3})$	$(h = 10^{-4})$
x_{k+1}	175	102	41	22	13
y_{k+1}	38	38	26	16	10
z_{k+1}	167	100	37	18	11

5. Discussion

Theorem 1 was applied to a discretization scheme, namely the fourth-order Runge-Kutta scheme with fixed step-size. This method is still widely used in many applications. Additionally, Runge-Kutta has been shown to be a good example to notice a significant amount of excluded (or neglected) terms (or monomials).

It could be argued that the proposed computational discretization scheme does not improve the performance of classical Runge-Kutta numerical methods since the removal of the neglected terms in the method is automatically performed by the computer and therefore the advantage of not including them in the general scheme has no effect on the results. Unfortunately that is not the case since the computer will perform the operations related to the neglected terms anyhow before returning zero. That certainly has an impact on the overall performance as shown here.

Stretching this argument a bit further, an advantage related to the number of operations performed by computers could be considered as not very important, if their current performance is taken into account. That is not entirely true, if a discretization scheme is to be implemented in a FPGA device for instance, the code length is certainly a factor to be taken into consideration. A shorter algorithm will certainly have an appeal in this case.

Another criticism is the use of a fixed step size Runge-Kutta of low order (considered outdated) when a variety of discretization schemes with variable step size are currently available. The change in the step will aggravate the problem when its value becomes smaller. More terms will be neglected and the final operation will be performed by far less terms which again shows the importance of the present study.

Finally showing the chaotic attractor as was done here could be considered as a simplistic way of analysing a difficult problem as no error control was used (and therefore could affect the simulation of the unstable invariant). Although a definite answer cannot be given before an exhaustive study, the idea that using a more elaborated discretization scheme (such as a variable step scheme) would guarantee such a simulation does not seem entirely possible as the user does not known which terms will be neglected in the simulation and therefore the unstable invariant will be calculated differently from what is expected.

6. Conclusion

This paper has introduced an effective computational discretization scheme for the solution of nonlinear dynamical systems. Based on a theorem, it has been shown how to safely exclude monomials in an implemented Runge-Kutta of fourth-order with almost no accuracy loss. The reduced Runge-Kutta of fourth-order (RRK4) has been successfully applied to the three classical nonlinear dynamical systems: Rössler equations, Lorenz equations and Sprott B.

The quality of simulations has been verified by means of the concept of observability of nonlinear dynamical systems, projections on plane *xy* of the attractors and computation of the largest Lyapunov exponent. Figure 2 reinforces the efficiency of RRK4 by showing trajectories practically identical. To demonstrate the computational effectiveness of the proposed technique the number of mathematical operations between RK4 and RRK4 was used for comparison. Table 6 shows that the achieved reduction is up to 81.8%. Consequently, simulation time has also significantly decreased, reaching 90.7% for the Lorenz case. Table 9 shows that decreasing the step-size, the number of neglected terms increases, further reducing the number of operations performed. This is a striking outcome that allows the user of such systems to speed up simulations and save energy and time in many applications. We hope that this method can be continuously developed and expanded in order to create novel algorithms for reducing the carbon footprint of scientific computation.

The proposed technique can be applied to a wide range of simulation applications. This finding is promising and should be applied to other nonlinear dynamical systems and extended to different discretization schemes. It is also believed that RRK4 would be beneficial for embedded employments, such as image encryption. Furthermore, as a future work, we intend to evaluate the use of the theorem in variable step-size discretization and in other discretization schemes.

Acknowledgements

Priscila F. S. Guedes was supported by Brazilian agency CAPES. Eduardo M. A. M. Mendes recognises financial support from CAPES, CNPQ (Grant 308129/2017-2) and FAPEMIG (Grant APQ-03197-18). Erivelton Nepomuceno was financially supported by Brazilian Research Agencies: CNPq (Grant No. 425509/2018-4 and Grant No. 311321/2020-8), CNPq/INERGE (Grant No. 465704/2014-0) and FAPEMIG (Grant No. APQ-00870-17). Erivelton Nepomuceno also received technical and financial support from the Centre for Ocean Energy Research and the Department of Electronic Engineering in the Maynooth University.

References

[1] L.A. Aguirre, S.B. Bastos, M.A. Alves, C. Letellier, Observability of nonlinear dynamics: normalized results and a time-series approach, Chaos 18 (1) (2008) 013123.

[2] J.I.M. Bezerra, V.V. de Almeida Camargo, A. Molter, A new efficient permutation-diffusion encryption algorithm based on a chaotic map, Chaos Solitons Fractals 151 (2021) 111235.

[3] J. Bhat, M. Saqib, A.H. Moon, Fuzzy extractor and chaos enhanced elliptic curve cryptography for image encryption and authentication, Int. J. Syst. Assur. Eng. Manage. (2021) 1–16.

[4] S. Bosmans, S. Mercelis, P. Hellinckx, J. Denil, Reducing computational cost of large-scale simulations using opportunistic model approximation, in: 2019 Spring Simulation Conference (SpringSim), IEEE, 2019, pp. 1–12.

[5] J.C. Butcher, N. Goodwin, Numerical Methods for Ordinary Differential Equations, vol. 2, Wiley Online Library, 2008.

[6] F. Dridi, S. El Assad, W. El Hadj Youssef, M. Machhout, R. Lozi, The design and FPGA-based implementation of a stream cipher based on a secure chaotic generator, Appl. Sci. 11 (2) (2021) 625.

[7] D. Faranda, M.F. Mestre, G. Turchetti, Analysis of round off errors with reversibility test as a dynamical indicator, Int. J. Bifurcation Chaos 22 (09) (2012) 1250215.

[8] Z. Galias, The dangers of rounding errors for simulations and analysis of nonlinear circuits and systems? And how to avoid them, IEEE Circuits Syst. Mag. 13 (3) (2013) 35–52.

[9] G. Gouesbet, J. Maquet, Construction of phenomenological models from numerical scalar time series, Physica D 58 (1–4) (1992) 202–215.

[10] S.M. Hammel, J.A. Yorke, C. Grebogi, Do numerical orbits of chaotic dynamical processes represent true orbits? J. Complex 3 (2) (1987) 136–145.

[11] M. Hanif, S. Abbas, M.A. Khan, N. Iqbal, Z.U. Rehman, M.A. Saeed, E.M. Mohamed, A novel and efficient multiple RGB images cipher based on chaotic system and circular shift operations, IEEE Access 8 (2020) 146408–146427.

[12] A. Hasan, E.C. Kerrigan, G.A. Constantinides, Control-theoretic forward error analysis of iterative numerical algorithms, IEEE Trans. Automat. Control 58 (6) (2013) 1524–1529.

[13] Institute of Electrical and Electronics Engineers IEEE, IEEE Standard for floating-point arithmetic, in: IEEE Std 754–2019 (Revision of IEEE 754–2008), 2019, pp. 1–84.

[14] T. Karimov, D. Butusov, A. Karimov, Comparison of analog and numerical chaotic system simulation, in: 2015 XVIII International Conference on Soft Computing and Measurements (SCM), IEEE, 2015, pp. 81–83.

[15] Q. Lai, S. Chen, Generating multiple chaotic attractors from Sprott B system, Int. J. Bifurcation Chaos 26 (11) (2016) 1650177.
[16] Q. Lai, G. Xu, H. Pei, Analysis and control of multiple attractors in Sprott B system, Chaos Solitons Fractals 123 (2019) 192–200.
[17] J.V. Lambers, A.C. Sumner, Explorations in Numerical Analysis, The University of Southern Mississippi, Hattiesburg, 2016.
[18] C. Letellier, L.A. Aguirre, Investigating nonlinear dynamics from time series: the influence of symmetries and the choice of observables, Chaos 12 (3) (2002) 549–558.
[19] C. Letellier, L.A. Aguirre, Interplay between synchronization, observability, and dynamics, Phys. Rev. E 82 (1) (2010) 016204.
[20] C. Letellier, L.A. Aguirre, J. Maquet, Relation between observability and differential embeddings for nonlinear dynamics, Phys. Rev. E 71 (6) (2005) 066213.
[21] C. Letellier, J. Maquet, L. Le Sceller, G. Gouesbet, L. Aguirre, On the non-equivalence of observables in phase-space reconstructions from recorded time series, J. Phys. A 31 (39) (1998) 7913.
[22] C. Letellier, E.M. Mendes, Robust discretizations versus increase of the time step for the Lorenz system, Chaos 15 (1) (2005) 013110.
[23] S.J. Liao, P.F. Wang, On the mathematically reliable long-term simulation of chaotic solutions of Lorenz equation in the interval [0,10000], Sci. China Phys. Mech. Astron. 57 (2) (2014) 330–335.
[24] B. Liu, L. Wang, Y.-H. Jin, F. Tang, D.-X. Huang, Improved particle swarm optimization combined with chaos, Chaos Solitons Fractals 25 (5) (2005) 1261–1271.
[25] E.N. Lorenz, Deterministic nonperiodic flow, J. Atmos. Sci. 20 (2) (1963) 130–141.
[26] R. Lozi, Can we trust in numerical computations of chaotic solutions of dynamical systems? in: Topology and dynamics of Chaos: in Celebration of Robert Gilmore's 70th birthday, World Scientific, 2013, pp. 63–98.
[27] R.E. Mickens, Nonstandard finite difference schemes for differential equations, J. Differ. Eqs. Appl. 8 (9) (2002) 823–847.
[28] S. Monaco, D. Normand-Cyrot, A combinatorial approach of the nonlinear sampling problem, in: Analysis and Optimization of Systems, Springer Berlin Heidelberg, Berlin, Heidelberg, 1990, pp. 788–797.
[29] R.E. Moore, R.B. Kearfott, M.J. Cloud, Introduction to Interval Analysis, vol. 110, Siam, 2009.
[30] E. Nepomuceno, S. Martins, G. Amaral, R. Riveret, On the lower bound error for discrete maps using associative property, Syst. Sci. Control Eng. 5 (1) (2017) 462–473, doi:10.1080/21642583.2017.1387874.
[31] E.G. Nepomuceno, Convergence of recursive functions on computers, J. Eng. 327 (5) (2014) 2–4.
[32] E.G. Nepomuceno, P.F. Guedes, A.M. Barbosa, M. Perc, R. Repnik, Soft computing simulations of chaotic systems, Int. J. Bifurcation Chaos 29 (08) (2019) 1950112.
[33] E.G. Nepomuceno, S.A. Martins, B.C. Silva, G.F. Amaral, M. Perc, Detecting unreliable computer simulations of recursive functions with interval extensions, Appl. Math. Comput. 329 (2018) 408–419.
[34] E.G. Nepomuceno, E.M. Mendes, On the analysis of pseudo-orbits of continuous chaotic nonlinear systems simulated using discretization schemes in a digital computer, Chaos Solitons Fractals 95 (2017) 21–32.
[35] E.G. Nepomuceno, L.G. Nardo, J. Arias-Garcia, D.N. Butusov, A. Tutueva, Image encryption based on the pseudo-orbits from 1D chaotic map, Chaos 29 (6) (2019) 061101.
[36] M.L. Overton, Numerical Computing with IEEE Floating Point Arithmetic, Society for Industrial and Applied Mathematics, Philadelphia, 2001.
[37] M. Perc, Visualizing the attraction of strange attractors, Eur. J. Phys. 26 (4) (2005) 579.
[38] A. Quarteroni, R. Sacco, F. Saleri, Numerical Mathematics, vol. 37, Springer Science & Business Media, 2010.
[39] A. Quarteroni, F. Saleri, Scientific computing with Matlab and Octave, Comput. Sci. Eng. (2006).
[40] O.E. Rössler, An equation for continuous chaos, Phys. Lett. A 57 (5) (1976) 397–398.
[41] T. Sauer, C. Grebogi, J.A. Yorke, How long do numerical chaotic solutions remain valid? Phys. Rev. Lett. 79 (1) (1997) 59.
[42] P.H.O. Silva, L.G. Nardo, S.A.M. Martins, E.G. Nepomuceno, M. Perc, Graphical interface as a teaching aid for nonlinear dynamical systems, Eur. J. Phys. 39 (6) (2018) 065105.
[43] J.C. Sprott, Some simple chaotic flows, Phys. Rev. E 50 (2) (1994) R647.
[44] E. Süli, D.F. Mayers, An Introduction to Numerical Analysis, Cambridge University Press, 2003.
[45] W. Tang, A note on continuous-stage Runge–Kutta methods, Appl. Math. Comput. 339 (2018) 231–241, doi:10.1016/j.amc.2018.07.044.
[46] A.V. Tutueva, T.I. Karimov, L. Moysis, E.G. Nepomuceno, C. Volos, D.N. Butusov, Improving chaos-based pseudo-random generators in finite-precision arithmetic, Nonlinear Dyn. 104 (1) (2021) 727–737.
[47] A. Wolf, J.B. Swift, H.L. Swinney, J.A. Vastano, Determining Lyapunov exponents from a time series, Physica D 16 (3) (1985) 285–317.
[48] L.-S. Yao, Computed chaos or numerical errors, arXiv preprint nlin/0506045 (2005).
[49] F. Yu, L. Li, B. He, L. Liu, S. Qian, Z. Zhang, H. Shen, S. Cai, Y. Li, Pseudorandom number generator based on a 5D hyperchaotic four-wing memristive system and its FPGA implementation, Eur. Phys. J. Spec. Top. 123 (2021).
[50] X. Zhuang, Q. Wang, J. Wen, Numerical dynamics of nonstandard finite difference method for nonlinear delay differential equation, Int. J. Bifurcation Chaos 28 (11) (2018) 1850133.