Chaos, Solitons and Fractals 193 (2025) 116084

Contents lists available at ScienceDirect

Chaos, Solitons and Fractals

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

Preservation of Lyapunov stability through effective discretization in Runge–Kutta method

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

[a] Graduate Program in Electrical Engineering, Federal University of São João Del-Rei, 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
[d] School of Computing and Digital Media London Metropolitan University, UK

ARTICLE INFO

Keywords:
Lyapunov stability theory
Discretization method
Computer simulation

ABSTRACT

To analyze continuous-time dynamic systems, it is often necessary to discretize them. Traditionally, this has been accomplished using various variants of the Runge–Kutta (RK) method and other available discretization schemes. However, recent advancements have revealed that effective discretization can be achieved by considering the precision of the computer. In studying the stability of such continuous systems according to Lyapunov theory, it is imperative to consider the Lyapunov function of dynamic systems described by differential equations, as well as their discrete counterparts. This study demonstrates that the discretization using the RK method and the effective discretization based on the reduced Runge–Kutta (RRK) method, wherein terms are reduced due to computational precision, preserve the Lyapunov stability across different step-size values. Despite a notable reduction in the number of terms, particularly evident in the fourth-order Runge–Kutta method, stability according to Lyapunov remains intact. Furthermore, reducing the number of terms decreases the operations required at each iteration, yielding reductions of up to 46.67%, 93.58%, and 99.91% for $RRK2$, $RRK3$, and $RRK4$, respectively, in the numerical example. This directly impacts computational cost, as illustrated in the numerical experiments.

1. Introduction

Continuous-time systems governed by differential equations are widely used to represent the behavior of real systems. An accurate mathematical model is essential to analyze the behavior of the systems and also to propose mechanisms to modify them to achieve a set of desired specifications. When the analysis requires the use of computers, a discretization scheme is often employed [1,2]. Thus, a discrete-time system is obtained to make the computational simulation possible [3–8]. One of the most celebrated discretization techniques in the literature is the fourth-order Runge–Kutta method, which is based on the expansion of the Taylor series [9–12].

In [13], it was demonstrated that a stable continuous-time system and its discretized counterpart via the Padé transformation (of any order) always share a specific polyhedral Lyapunov function, ensuring system stability. This result is particularly significant for the discretization of switched systems. Similarly, [14] proposed the Positive Definiteness Preserving Lyapunov Discretization (PDPLD) to address Lyapunov differential equations, preserving the positive definiteness of solutions. In [15], the SVIR (susceptible–vaccinated–infected–recovered) epidemic model was analyzed using the non-standard Mickens finite difference scheme. The study showed that this discretization scheme effectively preserves the global asymptotic stability of the equilibria of the corresponding continuous model. Furthermore, [16] introduced a discretization method for asymptotically stable homogeneous systems that maintains both the stability and convergence rate of the original continuous-time system.

When analyzing the discrete-time counterpart derived from a continuous-time representation, it is fundamental to check whether the numerical approximation can represent the behavior of the original

* Corresponding author.
E-mail addresses: pri12_guedes@hotmail.com (P.F.S. Guedes), emmendes@cpdee.ufmg.br (E.M.A.M. Mendes), nepomuceno@ufsj.edu.br (E. Nepomuceno), m.lacerda@londonmet.ac.uk (M.J. Lacerda).

https://doi.org/10.1016/j.chaos.2025.116084
Received 3 May 2024; Received in revised form 14 January 2025; Accepted 23 January 2025
Available online 8 February 2025

system with a high degree of fidelity. One way to analyze this correspondence is through the Lyapunov stability theory, a topic that has received great attention from the community [17–19].

In [20] an explicit nonstandard Runge–Kutta (ENRK) method was proposed. The method has a higher accuracy order and preserves the stability of autonomous dynamical systems. The method is based on the classical explicit Runge–Kutta method, where instead of the usual h in the formulas there stands an appropriately chosen function that depends on h. In [21], it was proposed dynamics, which are generalizations of the recognized ODE models formulated in [22], of two fractional-order SIQRA (Susceptible-Infected-Quarantine-Removed-Antidotal) malware propagation models and their discretizations. In this study, it was obtained step-size thresholds which guarantee the boundedness, and asymptotic stability properties of the fractional-order models are preserved correctly by the discrete-time models. In the same context, the dynamics of COVID-19 were studied in [23]. The behavior of the virus in the environment was investigated, considering human–environment–human transmission. The study was carried out using the fourth-order Runge–Kutta method and Lyapunov stability theory was used to analyze the global stability of the proposed model.

In [24] a non-standard finite differences numerical scheme was proposed. The scheme was validated by the fact that it preserves the fixed points and their stable nature and also the positivity. Furthermore, in most of the scenarios, the non-standard scheme behaves better or equivalently to a Runge–Kutta method of order 4, presenting also a lower computational complexity. In [25] it is shown that the global asymptotic stability of the equilibrium points of dynamic systems described by ordinary differential equations is established based on Lyapunov's stability theory. To study these continuous-time systems, an explicit non-standard finite difference (NSFD) method was proposed, which preserves any quadratic Lyapunov function V, i.e. they admit V as a discrete-time Lyapunov function.

In [26], a feedback control strategy is established to manage chaos created by bifurcation in a discretized predator–prey system. Additionally, in [27], stability analysis is performed for discretized systems using modified Pantakar–Runge–Kutta methods, where the stability is investigated through the eigenvalues of the Jacobian. Furthermore, in [28], a continuous-time cancer system is studied using Euler discretization methods, Taylor series expansion methods, and Runge–Kutta methods (RK), with stability analyzed via the eigenvalues of the Jacobian matrix. Lastly, in [29], a discretization approach is applied to uncertain continuous polytopic systems, with two sufficient LMI conditions proposed to design robust digital controllers. The first condition uses a constant Lyapunov function, while the second employs a parameter-dependent Lyapunov function, both ensuring the asymptotic stability of the closed-loop continuous-time system with the digital controller.

There is a theoretical concern for the continuous-time and discrete-time systems to be similar when employing discretization methods. However, simulations are carried out computationally, and due to the way computers are designed, there are some limitations in the simulation. In [30], the computational limitation was explored as a way to provide more efficient and faster discretization schemes. The method uses the computational precision of the machine to exclude some terms in the computation of the discretization. To evaluate the effectiveness of the method, the trajectory in the xy plane, the observability of the studied systems, as well as the Lyapunov exponent were investigated.

This article aims to evaluate the discretization of a continuous-time system by using tools from the Lyapunov theory and considering computational aspects of discrete-time models. The continuous-time system is discretized using the Runge–Kutta method. The discrete-time model is then reduced by using the technique developed in [30] which is able to eliminate some terms based on the computational precision. The Lyapunov function V of the continuous-time system is obtained by using the Sum of Squares (SOS) formulation [31,32]. Data from the reduced discrete-time model and from the original discrete-time model are then employed to check if the positivity of the Lyapunov function $V > 0$, and the negativity of ΔV is guaranteed. Numerical experiments using different step-sizes were employed to evaluate if the properties of the Lyapunov function are preserved in the discrete-time models. The experiments illustrate that the reduced discrete-time models and the original discretization can keep the properties of the Lyapunov function up to the same step-size. On the other hand, when Lyapunov stability cannot be guaranteed, this occurs for the same discretization step-size in both methods, RK and RRK (reduced Runge–Kutta). However, the reduced discrete-time model was able to reduce up to 99.84% of the terms in the discrete-time model when considering the fourth-order Runge–Kutta method.

By excluding negligible monomials based on computational precision, the effective discretization approach significantly reduces computational costs, particularly for the discrete-time Lyapunov function. This reduction is crucial for real-time applications, such as digital control systems and field-programmable gate array (FPGA) based implementations, where high sampling rates and rapid computation are essential. Specifically, when implementing discretization schemes on FPGAs, the length of the algorithm becomes a critical consideration. Reducing the computational complexity of discrete models directly translates into shorter and more efficient algorithms, which are ideal for FPGA deployment. These shorter algorithms not only optimize resource usage but also improve execution speed, enhancing the appeal of this approach in hardware-based systems [33,34]. Furthermore, employing the continuous Lyapunov function directly as a discrete Lyapunov function ensures that stability properties are retained without the need to redefine stability criteria, bridging theoretical guarantees with practical feasibility. This study demonstrates that this approach effectively balances computational efficiency and system stability, enabling scalable solutions for diverse real-time applications.

The rest of the paper is organized as follows. In Section 2, a background for the manuscript is presented. A brief description of numerical computing, the Runge–Kutta discretization method, and the sum of squares are reviewed. The method based on the exclusion of terms due to finite computer precision and stability analysis is presented in Section 3. Numerical Experiments are given in Section 4. Section 5 presents the conclusions.

2. Background

In this section, basic concepts of numerical computing, the Runge–Kutta discretization method and the sum of squares (SOS) decomposition are briefly described.

2.1. Numerical computing

The IEEE 754-2019 floating-point standard [35] aims to establish norms for representing and manipulating real numbers on computers. This standard delineates two fundamental formats: single format and double format. The analysis conducted herein focuses on the double format, the binary layout of which is illustrated in Table 1.

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

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

The precision of a floating-point system is determined by the number of bits allocated to the format, which includes the bits of the mantissa, including the hidden bit [36]. A floating-point system with precision ρ can be represented as:

$$x = \pm(1.b_1 b_2 \ldots b_{\rho-2} b_{\rho-1})_2 \times 2^E. \quad (1)$$

Using this representation, the following definition of 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. Runge–Kutta methods

The Runge–Kutta methods constitute a significant group of iterative techniques utilized for approximating solutions to ordinary differential equations. Consider the initial value problem determined by:

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

where x is the state variable and x_0 is the initial value of the state variable at time t_0. The estimated solution of (2) employ the Runge–Kutta numerical technique:

$$x_{k+1} = g(x_k, h), \tag{3}$$

where h is the step-size.

2.2.1. Second-order Runge–Kutta (RK2)

Let step-size $h > 0$, then RK2 can be expressed by [10]:

$$x_{k+1} = x_k + \frac{h}{2}\left(K_1 + K_2\right), \tag{4}$$

where

$$K_1 = f(x_k),$$
$$K_2 = f(x_k + hK_1), \tag{5}$$

and $f(x_k)$ is the differential equation and (5) is used for systems that do not depend on time explicitly.

2.2.2. Third-order Runge–Kutta (RK3)

Let step-size $h > 0$, then RK3 can be expressed by [10]:

$$x_{k+1} = x_k + \frac{h}{6}\left(K_1 + 4K_2 + K_3\right), \tag{6}$$

where

$$K_1 = f(x_k),$$
$$K_2 = f(x_k + \frac{h}{2}K_1),$$
$$K_3 = f(x_k + 2hK_2 - hK_1), \tag{7}$$

and $f(x_k)$ is the differential equation and (7) is used for systems that do not depend on time explicitly.

2.2.3. Fourth-order Runge–Kutta (RK4)

One of the most widely employed discretization methods is the fourth-order Runge–Kutta method [11,37,38]. Let step-size $h > 0$, then RK4 can be expressed by [10]:

$$x_{k+1} = x_k + \frac{h}{6}\left(K_1 + 2K_2 + 2K_3 + K_4\right), \tag{8}$$

where

$$K_1 = f(x_k),$$
$$K_2 = f(x_k + \frac{h}{2}K_1),$$
$$K_3 = f(x_k + \frac{h}{2}K_2),$$
$$K_4 = f(x_k + hK_3), \tag{9}$$

and $f(x_k)$ is the differential equation and (9) is used for systems that do not depend on time explicitly. The following Example is introduced to clarify the implementation of the Runge–Kutta methods:

Example 1. Consider the following continuous-time system:

$$\begin{cases} \dot{x} = y^2, \\ \dot{y} = x + y. \end{cases} \Longrightarrow \begin{cases} \dot{x} = f(x, y) = y^2, \\ \dot{y} = g(x, y) = x + y. \end{cases}$$

For K_1 we have:

$$K_{1_x} = f(x_k, y_k) = y_k^2,$$
$$K_{1_y} = g(x_k, y_k) = x_k + y_k.$$

Likewise, for K_2 we have:

$$\begin{aligned} K_{2_x} &= f(x_k + hK_{1_x}, y_k + hK_{1_y}), \\ &= f(x_k + hy_k^2, y_k + h(x_k + y_k)), \\ &= (y_k + h(x_k + y_k))^2, \\ K_{2_y} &= g(x_k + hK_{1_x}, y_k + hK_{1_y}), \\ &= g(x_k + hy_k^2, y_k + h(x_k + y_k)), \\ &= x_k + hy_k^2 + y_k + h(x_k + y_k). \end{aligned}$$

Therefore, the discretization of the continuous-time system using the RK2 method can be given by using (4)

$$\begin{aligned} x_{k+1} &= x_k + \frac{h}{2}(K_{1_x} + K_{2_x}) = x_k + hy_k^2 + \frac{1}{2}h^3(x_k^2 + y_k^2) \\ &\quad + h^2(x_k y_k + y_k^2) + h^3 x_k y_k, \\ y_{k+1} &= y_k + \frac{h}{2}(K_{1_y} + K_{2_y}) = y_k + h(x_k + y_k) + \frac{1}{2}h^2(x_k + y_k^2 + y_k). \end{aligned}$$

Remark 1. It is important to observe that discretizing the continuous-time polynomial system provided in (10) using RK2 also yields a state polynomial system in discrete-time. Additionally, the discretization step-size h is present in the system equations.

Remark 2. When applying RK3 and RK4 discretization to the system illustrated in Example 1, the resulting discrete-time system would contain a higher quantity of terms compared to RK2.

2.3. Sum of Squares (SOS)

Obtaining a stability certificate for a nonlinear system by using the Lyapunov theory requires finding a positive Lyapunov function that has its derivative negative along the trajectories of the system. This can be a very difficult task for general nonlinear systems. To obtain convex formulations, in this work we will consider the class of polynomial systems. The nonnegativity constraints will be ensured by using the sum of squares (SOS) formulation [31].

A multivariable polynomial $F(x_1, x_2, \dots, x_n)$ of degree $2d$ is SOS, if it can be written according to

$$F(x_1, x_2, \dots, x_n) = \sum_{i=1}^{m} f_i^2(x_1, x_2, \dots, x_n), \tag{10}$$

where each polynomial $f_i(x_1, x_2, \dots, x_n)$ has degree lower or equal to d. Eq. (10) is nonnegative and can be written as

$$F(x) = z^T Q z, \tag{11}$$

where z is a vector containing monomials of degree up to d of $(x_1, x_2, \dots, x_n)$.

2.4. Effective computational discretization scheme

According to Guedes et al. [30], in the discretization of continuous-time systems, some terms may be neglected due to computer precision. That is, considering the values of the initial conditions, parameters, and step-size, some terms resulting from the discretization of the continuous-time system under study can be excluded.

Theorem 3 ([30]). *Let γ be the set of monomials, that is, $\gamma = \{\alpha_1 10^{\beta_1}, \alpha_2 10^{\beta_2}, \dots, \alpha_n 10^{\beta_n}\}$, with $1 \le \alpha_n \le 9$ and with $\beta_n > \beta_{n-1} > \beta_{n-2} > \cdots > \beta_1$. Let Ω be the set of difference between β_n and other exponents, that is, $\Omega = \{\Omega_1, \Omega_2, \dots, \Omega_{n-1}\} = \{(\beta_n - \beta_1), (\beta_n - \beta_2), \dots, (\beta_n - \beta_{n-1})\}$. If $\Omega_i > \rho$, then the monomial γ_i may be excluded in the implementation of the discretization scheme.*

Example 2. Consider the following equation

$$X = 0.01 + 2.6731 \times 10^{-22} - 7.8424 \times 10^{-8}, \\ = 1.0 \times 10^{-2} + 2.6731 \times 10^{-22} - 7.8424 \times 10^{-8}.$$

According to Theorem 3, the set of monomials are as follows:

$$\gamma_1 = \alpha_1 10^{\beta_1} = 2.6731 \times 10^{-22},$$
$$\gamma_2 = \alpha_2 10^{\beta_2} = -7.8424 \times 10^{-8},$$
$$\gamma_3 = \alpha_3 10^{\beta_3} = 1.0 \times 10^{-2}.$$

The set Ω is given by

$$\Omega_1 = \beta_3 - \beta_1 = -2 - (-22) = 20,$$
$$\Omega_2 = \beta_3 - \beta_2 = -2 - (-8) = 6.$$

Since $\Omega_1 > 16$ (for double precision), γ_1 may be excluded without loss of accuracy.

$$X = 1 \times 10^{-2} + 0.0000000000000000 \times 10^{-2}, \\ -0.0000078424 \times 10^{-2} \\ = 0.9999921576 \times 10^{-2}. \quad (12)$$

This example executed in Matlab is presented below, which confirms the exclusion of the term $\gamma_1 = 2.6731 \times 10^{-22}$ does not change the final result.

```
>> format long
>> 0.01 + 2.6731e-22 - 7.8424e-8
ans =
  0.009999921576000
>> 0.01 - 7.8424e-8
ans =
  0.009999921576000
```

3. Main results

3.1. Stability analysis

The stability of a system can be verified by finding a Lyapunov function $V(x)$. According to Lyapunov's stability criterion, to ensure the stability of the system, $V(x)$ must be a positive-definite function that decreases along the system's trajectories; in other words, its derivative must be negative. This requirement is formally addressed in the following theorem.

Theorem 4 ([2]). *Consider the system $\dot{x} = f(x)$, and let $\mathcal{D} \subseteq \mathbb{R}^n$ be a neighborhood of the origin. If there is a continuously differentiable function $V : \mathcal{D} \to \mathbb{R}_+$ such that the following conditions hold:*

1. *$V(x) > 0$ for all $x \in \mathcal{D} \setminus \{0\}$ and $V(0) = 0$.*
2. *$\dot{V}(x) \le 0$ for all $x \in \mathcal{D}$.*

then, the origin is a stable equilibrium. If the second condition is negative definite in $\mathcal{D}$ then the origin is asymptotically stable. If $\mathcal{D} = \mathbb{R}^n$ and $V(x)$ is radially unbounded, i.e., $V(x) \to \infty$ as $\|x\| \to \infty$, then the result holds globally.

Let us assume for now that $f(x)$ is a polynomial vector field, and we are looking for a Lyapunov function $V(x)$ that is also a polynomial in x. In this case, the two conditions stated in Theorem 4 transform into requirements for the nonnegativity of certain polynomials.

Proposition 1 ([32]). *Suppose that for the polynomial system $\dot{x} = f(x)$, $x \in \mathbb{R}^n$, there exists a polynomial $V(x)$ of degree $2d$ such that $V(0) = 0$, and*

$$V(x) - \varphi_1(x) \quad \text{is a SOS}, \quad (13)$$

$$-\frac{\partial V}{\partial x} f(x) - \varphi_2(x) \quad \text{is a SOS}, \quad (14)$$

where $\varphi_1(x)$ and $\varphi_2(x)$ are positive polynomials. Then, $f(x)$ is globally asymptotically stable.

The classical conditions (Theorem 4) rely on finding a Lyapunov function and verifying its properties through direct inequalities, Proposition 1 uses SOS polynomials to provide a computationally tractable method to verify global asymptotic stability for systems where $f(x)$ is a polynomial vector field.

The main novelty of this paper is the application of the Lyapunov function structure obtained for continuous-time systems to discretized systems, especially for effective discretization. Let us consider that the stability of the system (2) has been verified using a polynomial Lyapunov function $V(x)$. A fundamental condition for the discrete-time model (3) to provide a fair representation of its continuous-time counterpart, is its capability to maintain the properties of the Lyapunov function V, obtained for the original continuous-time system, for a given step-size h. In other words, (3) accommodates the function V as a discrete Lyapunov function. As a result, drawing from the Lyapunov stability theory for discrete-time dynamical systems [39], we conclude that the difference equation model (3) upholds the global asymptotic stability of (2) irrespective of the chosen step-size.

The following steps summarize the proposed study.

1. The first step is to employ Proposition 1 to obtain a polynomial Lyapunov function of the original continuous-time systems using the sum of squares methods. The continuous-time polynomial Lyapunov function will be used to test the quality of the discretization method. In this sense, the function $V(x)$ will be replaced by $V(x_k)$.
2. The next step consists of obtaining the discretization using the Runge–Kutta method (RK), as described in Section 2.2. In this case, the RK method uses the original function $f(x)$ and the step-size h to obtain the following discrete-time system

   $$x_{k+1} = g(x_k, h). \quad (15)$$

   Note that, if the original function $f(x)$ is polynomial, then $g(x_k, h)$ is also a polynomial.
3. The third step consists of obtaining the effective discretization (RRK), that is, the reduction of terms according to Theorem 3.

   (a) x_n receives the discretized system.
   (b) Each monomial of x_n will be placed in a vector in which each element of the vector corresponds to a term (t_i) of the discretized system.
   (c) The reduced discretized system ($x_{R_{k+1}}$) is found by eliminating the terms that do not respect the computational precision, $\rho_{10} = 16$. In this case, a 64-bit operating system was employed.

   At the end of this step, the following discrete-time polynomial system is obtained

   $$x_{R_{k+1}} = \hat{g}(x_{R_k}, h, x_0). \quad (16)$$

   The polynomial function $\hat{g}$ in (16) is guaranteed to have fewer monomials than the polynomial function g in (15) which will impact the computational cost required to certify the stability of the discretized model.

4. From the discretized (15), the reduced discretized (16) system and using the Lyapunov function found in step 1, we carried out simulations to check the stability of the system. Aiming to verify whether the discretization and reduced discretization are capable of maintaining the properties of the Lyapunov function obtained for the original continuous-time system. Loop to verify the stability of the system using the Lyapunov conditions. At each iteration:
   (a) Check whether the Lyapunov conditions hold: $V(x_k) > 0$ and $V(x_{k+1}) - V(x_k) < 0$.
   (b) If either Lyapunov condition is violated, terminate the algorithm and return the current values of x_k and h.
   (c) If the conditions are satisfied, update x_{k+1} using the RK method: $x_{k+1} = \mathrm{RK}(f, x_k, h)$. The maximum step-size that guarantees stability was determined by testing various step-sizes, with increments of 0.1. Optionally adjust the step size h based on stability analysis.
   (d) Check the convergence criteria: Verify if the states (x_k and y_k) are sufficiently close to zero, i.e., if $|x_k| \leq 10^{-16}$ and $|y_k| \leq 10^{-16}$. This step confirms that the system is converging to the origin as $t \to \infty$.
   (e) If the convergence criteria are met, terminate the algorithm, as the system has stabilized.
5. The loop continues until convergence is achieved, i.e., the system remains stable, or the step size h and state x_k are adjusted based on the stability feedback.

Remark 5. In this work, the Runge–Kutta discretization was expanded so that high-order polynomials could be excluded. However, there are implementations of the Runge–Kutta method that keep the functions implicit making it difficult to exclude monomials from the RK model.

The pseudocode of the proposed method is described in Algorithm 1.

4. Numerical experiments

To illustrate the proposed method numerical experiments are considered. The routines were implemented in Maple 18 and in Matlab R2014a using the SOSTOOLS [40]. Consider the following system borrowed from [25]:

$$\begin{cases} \dot{x} = -Ax^3 + By, \\ \dot{y} = -Cx - Dy^3, \end{cases} \tag{17}$$

where A, B, C and D are positive constants. The system (17) admits the following quadratic Lyapunov function with the parameters $(A, B, C, D) = (0.16, 1.0, 1.0, 0.1)$.

$$V(x, y) = 1.209x^2 + 1.209y^2, \tag{18}$$

as a Lyapunov function. To evaluate the obtained RK and the reduced RRK the discrete-time counterpart will be used

$$V(x_k, y_k) = 1.209x_k^2 + 1.209y_k^2. \tag{19}$$

To observe the behavior of the system (17), the fourth-order Runge–Kutta method with $\Delta t = 10^{-3}$, the set of the parameters $(A, B, C, D) = (0.16, 1.0, 1.0, 0.1)$ and initial conditions $(x(0), y(0)) = (0.5, 0.01)$ was employed. The reference solution is shown in Fig. 1, which was obtained using ODE4 in Matlab software. The system is stable, therefore when $t \to \infty$, $x(t)$ and $y(t)$ converge to zero. In other words, the phase plane forms a spiral into the origin.

Subsequently, the system (17) was discretized by the second, third, and fourth-order Runge–Kutta methods, the number of terms in this

Algorithm 1

$V(x) \leftarrow$ LYAPUNOV(f)

$V(x_k) \leftarrow V(x)$
$x_{k+1} \leftarrow RK(f, x, h)$

procedure RRK(x_{k+1}, h, x_0)
    $x_n \leftarrow x_{k+1}$
    $x_n \leftarrow [t_1 \quad t_2 \quad t_3 \ldots]$
    $x_{R_{k+1}} \leftarrow find(x_n \quad that \quad \Omega < \rho_{10})$
    return x_R
end procedure

while not converged **do**
    Check if Lyapunov conditions hold: $V(x_k) > 0$ and $V(x_{k+1}) - V(x_k) < 0$
    if $V(x_k) \leq 0$ **or** $V(x_{k+1}) - V(x_k) \geq 0$ **then**
        Terminate the algorithm
        return x_k, h
    else
        Update $x_{k+1} \leftarrow \mathrm{RK}(f, x_k, h)$
        Optionally update h based on stability analysis
        $k \leftarrow k + 1$
    end if

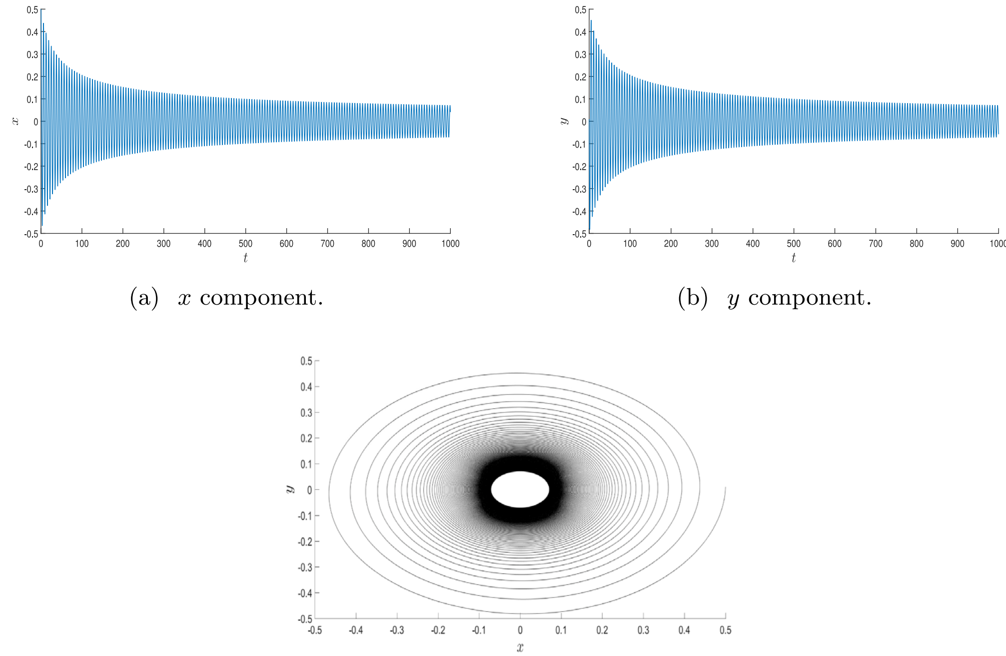
    Check convergence criteria:
    if $|x_k| \leq 10^{-16}$ **and** $|y_k| \leq 10^{-16}$ **then**
        Convergence achieved. Terminate the algorithm.
        return x_k, h
    end if
end while
return x_k

Table 2
Number of monomials for each of the discretized equations for the systems (17). The comparison is made between second-order Runge–Kutta (RK2), Reduced RK2 (RRK2), third-order Runge–Kutta (RK3), Reduced RK3 (RRK3), fourth-order Runge–Kutta (RK4) and Reduced RK4 (RRK4). Initial conditions $(x_0, y_0) = (0.5, 0.01)$, parameters $(A, B, C, D) = (0.16, 1, 1, 0.1)$ and step-size $h = 10^{-3}$.

Equations	RK2	RRK2	Reduction
x_{k+1}	14	10	28.57%
y_{k+1}	14	8	42.86%
	RK3	RRK3	Reduction
x_{k+1}	199	13	93.47%
y_{k+1}	199	11	94.47%
	RK4	RRK4	Reduction
x_{k+1}	9279	15	99.84%
y_{k+1}	6968	12	99.83%

discretization is presented in Table 2. It was found that as the order of the Runge–Kutta method increases, the number of discretization terms also increases. Furthermore, there was a reduction in the terms of this discretization based on Guedes et al. [30]. It was possible to verify that the reduction in the number of terms was significant, especially for the fourth-order Runge–Kutta. This reduction was possible due to the finite precision of the computer considered in Theorem 3.

For the study of the reduction in the number of terms, considering the variable x, it is possible to verify that for the second-order Runge–Kutta, there was a reduction of approximately 28.57% in the number of terms, whereas for the third-order Runge–Kutta a reduction of 93.47%, while for the fourth-order Runge–Kutta discretization there was a significant reduction of 99.84%. Likewise, for the variable y there was a reduction in the number of terms of approximately 42.86%, 94.47%, and 99.83% for the second, third, and fourth-order methods, respectively. Due to the reduction in the number of terms, there is a reduction in the number of operations performed at each iteration,



(a) x component. (b) y component.

(c) The phase plane.

Fig. 1. The reference solution generated by employing the RK4 method using ODE4 with $h = 10^{-3}$.

consequently contributing to a reduction in computational time.

Table 3 summarizes the basic operations used in the calculations. The system presents a total of 210 operations per iteration using RK2 whereas, using RRK2, the number of operations was 112, representing a reduction of approximately 46.67% of operations performed per iteration. There were 2444 and 157 mathematical operations per iteration when discretized using RK3 and RRK3, respectively. The reduction in the number of mathematical operations was 93.58%. Likewise, there were a total of 214155 operations when discretized using RK4 and 184 operations using RRK4. Therefore, there was a reduction of approximately 99.91% in the number of mathematical operations. Despite this substantial reduction, especially with RRK4, the system's characteristics and simulation quality were maintained.

For the discretized and reduced system for the different Runge–Kutta methods, according to the data presented in Table 2, it was possible to verify that by carrying out the numerical simulation, the system maintains stability as verified by applying ODE4. To illustrate, Fig. 2 presents the result for the reduced third-order Runge–Kutta method. It is possible to affirm that when $t \to \infty$, $x(t)$ and $y(t)$ converge to zero. In the same way, as observed in Fig. 1, ensuring the stability of the system (17). Moreover, from Fig. 3 it is possible to see that the characteristics of the Lyapunov function in (18) are also preserved, that is, $V > 0, \forall x \neq 0$ and $\Delta V < 0, \forall x \neq 0$.

The equivalence presented by the Runge–Kutta and reduced Runge–Kutta methods is related to the computational precision and order of convergence of the discretized systems. According to [10], the $RK2$, $RK3$, and $RK4$ methods have orders of convergence 2, 3, and 4, respectively. The order of convergence p of a numerical method measures how quickly the overall error decreases as the step-size h decreases. In this study, the root mean square error (RMSE) will be considered. The RMSE is calculated as follows

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{approx} - x_{ref})^2},$$

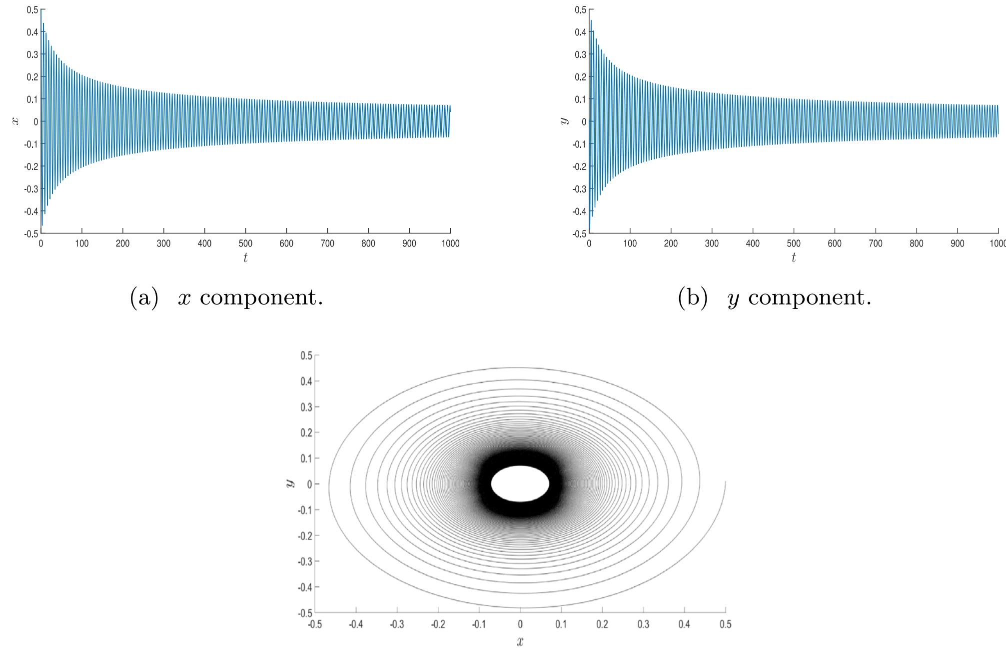
where n is the number of samples, x_{approx} are the values obtained by the Runge–Kutta methods and x_{ref} is the reference value obtained using Matlab's ode4.

Table 3
Summary of computational complexity. The basic operations used were analyzed, that is, Sum/Subtraction, Multiplication/Division and Power. For each method, all the operators for variables x, y are added and the reduction was calculated.

Operations	RK2		RRK2		Reduction
	x_{k+1}	y_{k+1}	x_{k+1}	y_{k+1}	
Sum/Subtraction	13	13	9	7	
Multiplication/Division	60	60	37	28	
Power	32	32	18	12	
Summation of operators	105	105	65	47	
Total	210		112		46.67%
Operations	RK3		RRK3		Reduction
	x_{k+1}	y_{k+1}	x_{k+1}	y_{k+1}	
Sum/Subtraction	205	223	12	10	
Multiplication/Division	515	533	52	40	
Power	475	493	24	19	
Summation of operators	1195	1249	88	69	
Total	2444		157		93.58%
Operations	RK4		RRK4		Reduction
	x_{k+1}	y_{k+1}	x_{k+1}	y_{k+1}	
Sum/Subtraction	9278	6967	14	11	
Multiplication/Division	59 447	44 165	62	45	
Power	54 136	40 162	30	22	
Summation of operators	122 861	91 294	106	78	
Total	214 155		184		99.91%

Table 4 shows this result, noting that the error is similar for both RK and RRK. From this analysis, we can see that for the considered example, the RRK method presents the same order as the RK. In addition, when the order of the RK and RRK methods increase, the error decreases. Furthermore, when the step-size h is divided by two, the RMSE (Root Mean Squared Error) decreases approximately in a ratio of 4, 8, and 16 for $RK2$, $RK3$ and $RK4$, respectively, due to the order of convergence. The same behavior is noted for the RRK.

Table 5 shows the maximum value for the step-size at which stability is guaranteed according to Lyapunov. The results for the step-size

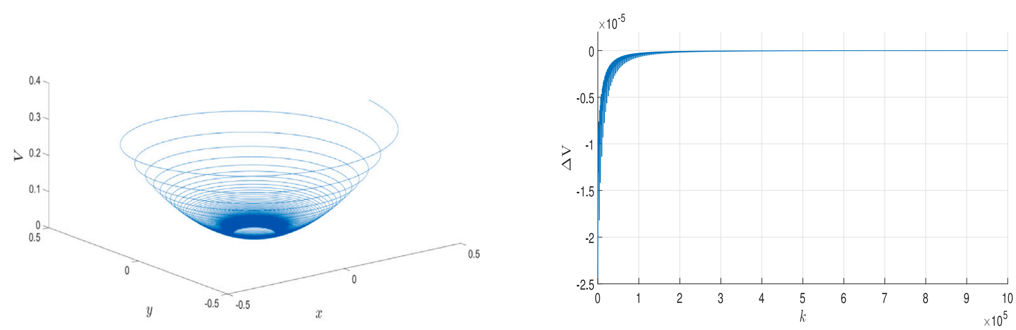


(a) x component.

(b) y component.

(c) The phase plane.

Fig. 2. The effective solution generated by employing the RRK3 method with $h = 10^{-3}$.



(a) V for Reduced RK3

(b) ΔV for Reduced RK3

Fig. 3. The discrete Lyapunov function by employing the RRK3 method with $h = 10^{-3}$.

Table 4
Root Mean Squared Error for the Runge–Kutta methods for $(x_0, y_0,) = (0.5, 0.01)$, $(A, B, C, D) = (0.16, 1.0, 1.0, 0.1)$, and step-size h of 10^{-3}.

Step-size	RK2		RRK2	
	$RMSE_x$	$RMSE_y$	$RMSE_x$	$RMSE_y$
$h = 1 \times 10^{-3}$	1.63×10^{-6}	1.6143×10^{-6}	1.6824×10^{-6}	1.6755×10^{-6}
$h = 5 \times 10^{-4}$	4.0843×10^{-7}	4.0159×10^{-7}	3.9642×10^{-7}	3.9411×10^{-7}
	RK3		RRK3	
	$RMSE_x$	$RMSE_y$	$RMSE_x$	$RMSE_y$
$h = 1 \times 10^{-3}$	2.5359×10^{-8}	2.5478×10^{-8}	2.5723×10^{-8}	2.4746×10^{-8}
$h = 5 \times 10^{-4}$	2.9699×10^{-9}	2.9848×10^{-9}	2.9904×10^{-9}	3.0476×10^{-9}
	RK4		RRK4	
	$RMSE_x$	$RMSE_y$	$RMSE_x$	$RMSE_y$
$h = 1 \times 10^{-3}$	2.5216×10^{-9}	2.5329×10^{-9}	2.1367×10^{-9}	2.1386×10^{-9}
$h = 5 \times 10^{-4}$	1.3733×10^{-10}	1.4831×10^{-10}	1.4344×10^{-10}	1.2317×10^{-10}

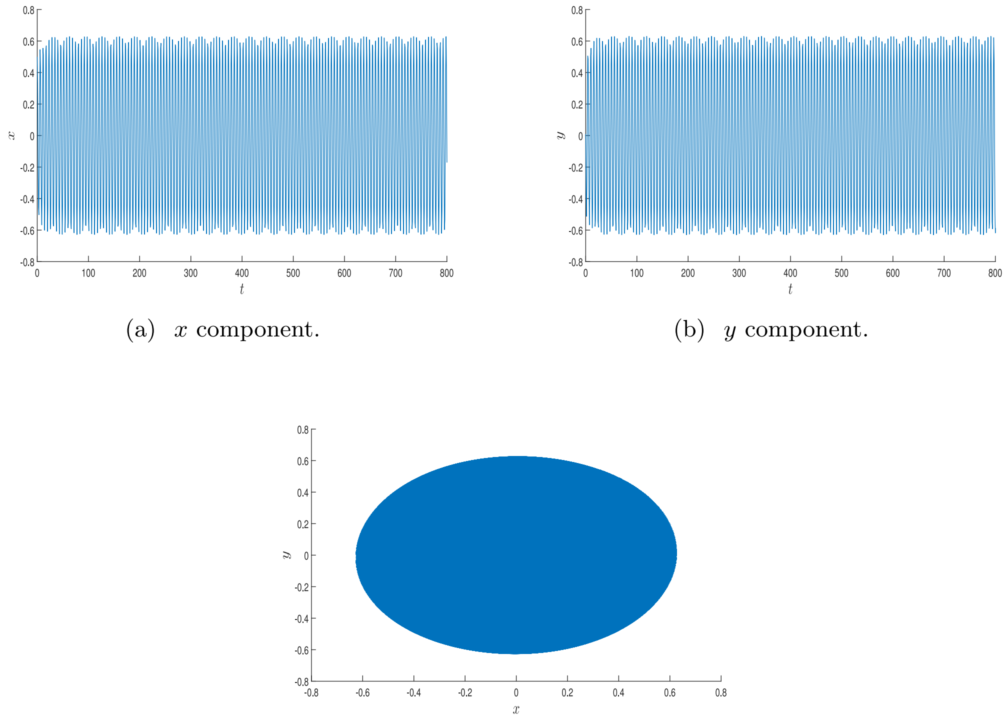
for the original Runge–Kutta and the reduced Runge–Kutta are the same, once again affirming Theorem 3 and its applicability. Based on this limit, for the discretization considering $h = 0.8$ for $RK2$, the stability of system (17) is not guaranteed, as observed in Figs. 4 and 5. It is observed that the components $x(t)$ and $y(t)$ remain unstable as they did not converge to 0 (Fig. 4), that is when $t \to \infty$ these components continue to oscillate. Furthermore, the properties of the Lyapunov function in (18) are not preserved (Fig. 5), ΔV assumes both positive and negative values along the trajectory of the system.

Table 5
Maximum step-size limit for the system discretized by Runge–Kutta methods in which stability is guaranteed according to Lyapunov.

Method	RK2	RRK2	RK3	RRK3	RK4	RRK4
	$h = 0.1$	$h = 0.1$	$h = 0.3$	$h = 0.3$	$h = 1.2$	$h = 1.2$

The reduction of terms as performed in [30] takes into account the computational precision of the software. Furthermore, this reduction guarantees the observability of the system, the attractors, and the Lyapunov exponent as shown in [30]. Making a random reduction of terms, that is, not following a pattern to neglect some terms, can contribute to erroneous results.

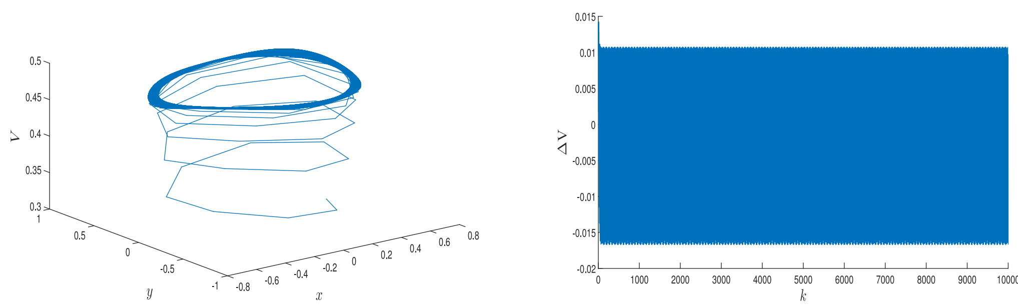
To illustrate this scenario, Table 6 presents the number of terms for discretization of system (17) by the second-order Runge–Kutta method,



(a) x component. (b) y component.

(c) The phase plane.

Fig. 4. The solution generated by employing the RK2 method with $h = 0.8$.



(a) V for RK2 (b) ΔV for RK2

Fig. 5. The discrete Lyapunov function by employing the RK2 method with $h = 0.8$.

Table 6
Number of monomials for each of the discretized equations for the systems (17). The comparison is made between second-order Runge–Kutta (RK2), Reduced RK2 (RRK2), and New Reduced RK2 (NRRK2). Initial conditions $(x_0, y_0) = (0.5, 0.01)$, parameters $(A, B, C, D) = (0.16, 1.0, 1.0, 0.1)$ and step-size $h = 10^{-3}$.

Equations	RK2	RRK2	NRRK2
x_{k+1}	14	10	8
y_{k+1}	14	8	7

where $RK2$ is the original discretization, $RRK2$ is the reduced discretization according to Theorem 3 and $NRRK2$ represents a random discretization not following any pattern.

For $NRRK2$ it is not possible to guarantee that it has the same characteristics as the system (17). Fig. 6 shows that the phase plane does not form a spiral into the origin, as in Fig. 1(c). It is possible to verify that due to this reduction in the number of terms, the characteristics of the system are not preserved, in the same way, that the characteristics of stability according to Lyapunov were also not guaranteed, that is, ΔV is not less than zero.

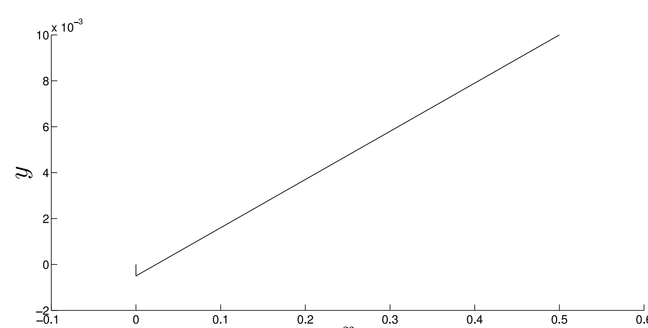


Fig. 6. The phase plane. Solution generated by employing the $NRRK2$ with $h = 10^{-3}$.

To confirm that ΔV is not negative, the value of the first five iterations is presented below, showing that ΔV is zero and remains so. In other words, this random discretization (NRRK2) does not preserve the properties of the Lyapunov function.

```
>> format long
>> deltaV(1:5)'
ans =
  -0.302370597260196
  -0.000000302739803
  -0.000000000000001
   0.000000000000000
   0.000000000000000
```

5. Conclusion

In this study, the stability analysis is conducted based on the Lyapunov function of a system discretized using an effective scheme derived from the Runge–Kutta method. Given the reduction in the number of terms resulting from computational precision in the discretized model obtained through this scheme, it was important to evaluate whether Lyapunov stability would be impacted.

Utilizing the Lyapunov function V of the continuous-time system in both the discretized system derived from the standard Runge–Kutta method and the reduced discretized system stemming from the effective scheme, it is demonstrated that the properties of the Lyapunov function are maintained, specifically, $V > 0, \forall x \neq 0$ and $\Delta V < 0, \forall x \neq 0$. This finding is of considerable significance as it underscores that, despite the reduction in the number of terms – especially noticeable in the fourth-order Runge–Kutta method – the computational cost is significantly diminished while still preserving the properties of the Lyapunov function. Moreover, both the RK and RRK methods lose the ability to guarantee Lyapunov stability at the same step-size h. In other words, when the step-size becomes too large to maintain stability, both methods fail to preserve the Lyapunov properties at the same critical step-size. As future research, the authors are investigating the properties of the Lyapunov function of a discretized and reduced system under denial of service (DoS) attacks.

CRediT authorship contribution statement

Priscila F.S. Guedes: Writing – review & editing, Writing – original draft, Software, Methodology, Investigation, Conceptualization. **Eduardo M.A.M. Mendes:** Writing – review & editing, Validation, Supervision, Methodology, Funding acquisition. **Erivelton Nepomuceno:** Writing – review & editing, Validation, Supervision, Methodology, Funding acquisition. **Marcio J. Lacerda:** Writing – review & editing, Writing – original draft, Supervision, Project administration, Methodology, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Priscila F. S. Guedes was supported by Brazilian agency CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior). Eduardo M. A. M. Mendes acknowledges financial support from CNPq (Grants No. 310788/2021-8 and No. 405419/2021-0). Erivelton Nepomuceno is grateful for the financial support of Science Foundation Ireland under Grant number 21/FFP-P/10065 and Brazilian Research Agencies: CNPq/INERGE (Grant No. 465704/2014-0), CNPq (Grant No. 425509/2018-4 and Grant No. 311321/2020-8) and FAPEMIG (Grant No. APQ-00870-17). Marcio J. Lacerda was supported by CNPq (Grant No. 315538/2021-0).

Data availability

No data was used for the research described in the article.

References

[1] Ascher UM, Petzold LR. Computer methods for ordinary differential equations and differential–algebraic equations, vol. 61, Siam; 1998.
[2] Khalil HK. Nonlinear systems. 3rd ed.. Upper Saddle River, NJ: Prentice Hall; 2002.
[3] Ardourel V, Jebeile J. On the presumed superiority of analytical solutions over numerical methods. Eur J Philos Sci 2017;7(2):201–20.
[4] Hammel SM, Yorke JA, Grebogi C. Do numerical orbits of chaotic dynamical processes represent true orbits? J Complexity 1987;3(2):136–45.
[5] Sauer T, Grebogi C, Yorke JA. How long do numerical chaotic solutions remain valid? Phys Rev Lett 1997;79(1):59.
[6] Awrejcewicz J, Krysko V, Papkova I, Krysko A. Routes to chaos in continuous mechanical systems. part 1: Mathematical models and solution methods. Chaos Solitons Fractals 2012;45(6):687–708.
[7] Lozi R. 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, p. 63–98.
[8] Zhuang X, Wang Q, Wen J. Numerical dynamics of nonstandard finite difference method for nonlinear delay differential equation. Int J Bifurc Chaos 2018;28(11):1850133.
[9] Cartwright JH, Piro O. The dynamics of Runge–Kutta methods. Int J Bifurc Chaos 1992;2(03):427–49.
[10] Butcher JC, Goodwin N. Numerical methods for ordinary differential equations, vol. 2, Wiley Online Library; 2008.
[11] Quarteroni A, Sacco R, Saleri F. Numerical mathematics. 2nd ed.. vol. 37, Springer Science & Business Media; 2010.
[12] Noorani M, Hashim I, Ahmad R, Bakar S, Ismail E, Zakaria A. Comparing numerical methods for the solutions of the Chen system. Chaos Solitons Fractals 2007;32(4):1296–304.
[13] Rossi F, Colaneri P, Shorten R. Padé discretization for linear systems with polyhedral Lyapunov functions. IEEE Trans Autom Control 2011;56(11):2717–22.
[14] Gillis J, Diehl M. A positive definiteness preserving discretization method for Lyapunov differential equations. In: 52nd IEEE conference on decision and control. IEEE; 2013, p. 7759–64.
[15] Geng Y, Xu J. Stability preserving nsfd scheme for a multi-group svir epidemic model. Math Methods Appl Sci 2017;40(13):4917–27.
[16] Sanchez T, Polyakov A, Efimov D. Lyapunov-based consistent discretization of stable homogeneous systems. Internat J Robust Nonlinear Control 2021;31(9):3587–605.
[17] Duarte-Mermoud MA, Aguila-Camacho N, Gallegos JA, Castro-Linares R. Using general quadratic Lyapunov functions to prove Lyapunov uniform stability for fractional order systems. Commun Nonlinear Sci Numer Simul 2015;22(1–3):650–9.
[18] Cangiotti N, Capolli M, Sensi M, Sottile S. A survey on Lyapunov functions for epidemic compartmental models. Bollettino dell'Unione Matematica Italiana; 2023, p. 1–17.
[19] Vargas-De-León C. Lyapunov functions for two-species cooperative systems. Appl Math Comput 2012;219(5):2493–7.
[20] Dang QA, Hoang MT. Positive and elementary stable explicit nonstandard Runge-Kutta methods for a class of autonomous dynamical systems. Int J Comput Math 2020;97(10):2036–54.
[21] Hoang MT. Dynamical analysis of two fractional-order siqra malware propagation models and their discretizations. Rend Circ Mat Palermo Ser 2 2023a;72(1):751–71.
[22] Piqueira JRC, Batistela CM. Considering quarantine in the sira malware propagation model. Math Probl Eng 2019;2019:1–8.
[23] Asamoah JKK, Owusu MA, Jin Z, Oduro F, Abidemi A, Gyasi EO. Global stability and cost-effectiveness analysis of covid-19 considering the impact of the environment: using data from ghana. Chaos Solitons Fractals 2020;140:110103.
[24] Cresson J, Pierret F. Non standard finite difference scheme preserving dynamical properties. J Comput Appl Math 2016;303:15–30.
[25] Hoang MT. Nonstandard finite difference methods preserving general quadratic Lyapunov functions. 2023, arXiv preprint arXiv:2312.01471.
[26] Tassaddiq A, Shabbir MS, Din Q, Naaz H. Discretization, bifurcation, and control for a class of predator–prey interactions. Fract Fract 2022;6(1):31.
[27] Izgin T, Kopecz S, Meister A. On Lyapunov stability of positive and conservative time integrators and application to second order modified Patankar–Runge–Kutta schemes. ESAIM Math Model Numer Anal 2022;56(3):1053–80.
[28] Saeed T, Djeddi K, Guirao JL, Alsulami HH, Alhodaly MS. A discrete dynamics approach to a tumor system. Math 2022;10(10):1774.
[29] Keles NA, Frezzatto L, Mendes EMAM, da Silva Campos VC. Discretization and state feedback control for uncertain linear systems—a new approach considering linear multistep method theory. J Franklin Inst 2024;361(1):489–500.

[30] Guedes PF, Mendes EM, Nepomuceno E. Effective computational discretization scheme for nonlinear dynamical systems. Appl Math Comput 2022;428:127207.

[31] Papachristodoulou A, Prajna S. On the construction of Lyapunov functions using the sum of squares decomposition. In: Proceedings of the 41st IEEE conference on decision and control, 2002. vol. 3, IEEE; 2002, p. 3482–7.

[32] Papachristodoulou A, Prajna S. A tutorial on sum of squares techniques for systems analysis. In: Proceedings of the 2005, American control conference, 2005. IEEE; 2005, p. 2686–700.

[33] El-Banna H, El-Fattah AA, Fakhr W. An efficient implementation of the 1d dct using fpga technology. In: Proceedings of the 12th IEEE international conference on fuzzy systems (cat. no. 03CH37442). IEEE; 2003, p. 278–81.

[34] Valtierra-Rodriguez M, Contreras-Hernandez J-L, Granados-Lieberman D, Rivera-Guillen JR, Amezquita-Sanchez JP, Camarena-Martinez D. Field-programmable gate array architecture for the discrete orthonormal stockwell transform (dost) hardware implementation. J Low Power Electron Appl 2024;14(3).

[35] Institute of Electrical and Electronics Engineers (IEEE). IEEE standard for floating-point arithmetic. IEEE Std 754-2019 (Revision of IEEE 754-2008; 2019, p. 1–84.

[36] Overton ML. numerical computing with IEEE floating point arithmetic. Philadelphia: Society for Industrial and Applied Mathematics; 2001.

[37] Quarteroni A, Saleri F. Scientific computing with Matlab and Octave. Comput Sci Eng 2006.

[38] Tang W. A note on continuous-stage Runge–Kutta methods. Appl Math Comput 2018;339:231–41.

[39] Stuart A, Humphries AR. Dynamical systems and numerical analysis, vol. 2, Cambridge University Press; 1998.

[40] Papachristodoulou A, Anderson J, Valmorbida G, Prajna S, Seiler P, Parrilo PA. SOSTOOLS: Sum of squares optimization toolbox for MATLAB. 2013, http://arxiv.org/abs/1310.4716, Available from http://www.eng.ox.ac.uk/control/sostools.