Activity Homogeneity: A Measure for Comparing Time Discretization and State Quantization in ODE Simulation.

Journal Title
XX(X):1–21

DOI: 10.1177/ToBeAssigned
www.sagepub.com/
SAGE

Mariana Bergonzi[1], Rodrigo Castro[2] and Ernesto Kofman[1]

Abstract

In this work, we introduce the concept of *activity homogeneity* in the solutions of ordinary differential equations (ODEs) and characterize it through a metric called *Homogeneity Factor*. This indicator quantifies the degree of similarity in the temporal evolution of the system state variables. We show that this measure can be related to the convenience of using classic numerical integration schemes or quantization-based methods such as Quantized State System (QSS) algorithms.

The developed notions, which are also extended to systems exhibiting discontinuities, provide a theoretical argument that corroborates observations from previous works, indicating that QSS methods offer advantages when activity is heterogeneous, systems are sparse, and/or frequent discontinuities occur.

The concepts are applied to two case studies: an advection-diffusion-reaction model and a spiking neural network. Theoretical predictions are compared with empirical results obtained through simulations using different numerical integration methods, confirming that the proposed metric consistently identifies the integration strategy that is more computationally efficient in practice.

Keywords

Continuous System Simulation, Numerical Integration Algorithms, Quantized State Systems, Activity, Computational Efficiency

1 Introduction

Classic simulation methods for continuous systems described by ordinary differential equations (ODEs) are fundamentally based on temporal discretization techniques [1–3]. These algorithms advance the global system state by computing synchronous updates at discrete time points across all variables, regardless of the localized dynamics that may be present.

An alternative framework is provided by Quantized State System (QSS) methods [3–5], which operate by performing asynchronous updates on individual state variables, only when significant enough changes occur. As a result, QSS methods naturally exploit the sparsity and locality of activity in dynamical systems, offering computational advantages when state changes are confined to a limited subset of variables, i.e. when the activity [6–8] of the system exhibits a certain degree of *heterogeneity*.

In this work, we first extend the concept of *activity*, that had previously been defined to measure the total rate of change of a continuous signal [8]. In this extension, we distinguish the local activity of a particular state trajectory from the global activity of the whole state vector. We show that local activity is related to the minimum number of steps

[2]ICC-CONICET, FCEyN-UBA, Buenos Aires, ARGENTINA
[1]CIFASIS-CONICET, FCEIA-UNR, Rosario, ARGENTINA

Corresponding author:
Mariana Bergonzi, CIFASIS-CONICET, Rosario, Argentina.
Email: bergonzi@cifasis-conicet.gov.ar

required by QSS methods, while global activity is related to the minimum number of steps required by classic discrete time numerical integration algorithms.

Then, using the expressions obtained for local and global activity, we characterize the *activity homogeneity* by defining the *Homogeneity Factor*. This quantitative metric provides a criterion for identifying the most suitable simulation approach, considering the potential advantages of time slicing-based versus state quantization-based integration methods.

These concepts, initially developed for purely continuous systems, are then extended to account for the presence of discontinuities.

We show that the analysis based on the activity homogeneity constitutes the theoretical corroboration of the practical results observed in most publications comparing QSS and classic numerical algorithms, where the convenience of QSS is related to combinations of heterogeneous activity, sparsity, and the presence of frequent discontinuities.

The paper is organized as follows. Section 2 introduces a selection of previous concepts used throughout the rest of the work. Then, Section 3 presents the main results, defining the notions of instantaneous, local, and global activity, as well as the concept of Activity Homogeneity, and relates them to the computational costs of different simulation approaches. After that, Section 4 illustrates the results in two case studies and the conclusions are presented in Section 5.

2 Background

This section introduces the fundamental concepts of numerical integration algorithms based on time discretization and state quantization, along with the notion of activity.

2.1 Numerical integration methods based on Time Discretization

Let us consider a continuous-time dynamical system defined by the following set of ordinary differential equations:

$$\dot{\mathbf{x}}_a(t) = \mathbf{f}(\mathbf{x}_a(t), \mathbf{u}(t)) \tag{1}$$

where $\mathbf{x}_a(t) \in \mathbb{R}^N$ is the state vector, $\mathbf{u}(t) \in \mathbb{R}^m$ is a known input trajectory. We assume that the system of Eq.(1) has a unique solution.

Simulating such a system involves computing the state evolution starting from the initial condition $\mathbf{x}_a(t_0) = \mathbf{x}_0$. However, the analytical solution $\mathbf{x}_a(t)$ to such system is not available in most cases, and consequently, a numerical approximation $\mathbf{x}(t)$ is typically required. This approximation is obtained through numerical integration algorithms, which provide the foundational framework for continuous-time system simulation.

Classic numerical integration methods [1–3] rely on time discretization, iteratively computing approximations of the system state at discrete time instants. Some of these methods, such as the Runge-Kutta family, are classified as single-step methods that compute the next state using only information from the current step. Others, like Adams-Bashforth or Backward Differentiation Formulas (BDF), are multi-step methods that incorporate information from previous time steps to improve accuracy without increasing computational cost.

In addition, implicit methods, such as the implicit Euler or BDF schemes, require the use of future state information, typically obtained through iterative procedures, to compute the next state. These methods improve numerical stability, especially in stiff systems where explicit methods demand very small time steps to maintain stability.

In many practical applications, integration schemes with adaptive step-size control are employed. These methods dynamically adjust the time step based on local error estimates, aiming to maintain a predefined error tolerance while maximizing computational efficiency. Popular examples include adaptive Runge-Kutta methods (e.g., Dormand-Prince) and variable-step BDF methods.

2.2 Numerical integration methods based on State Quantization

QSS methods are a family of numerical integration algorithms that replace the time discretization used in classic numerical integration algorithms by the quantization of the state variables [3,9].

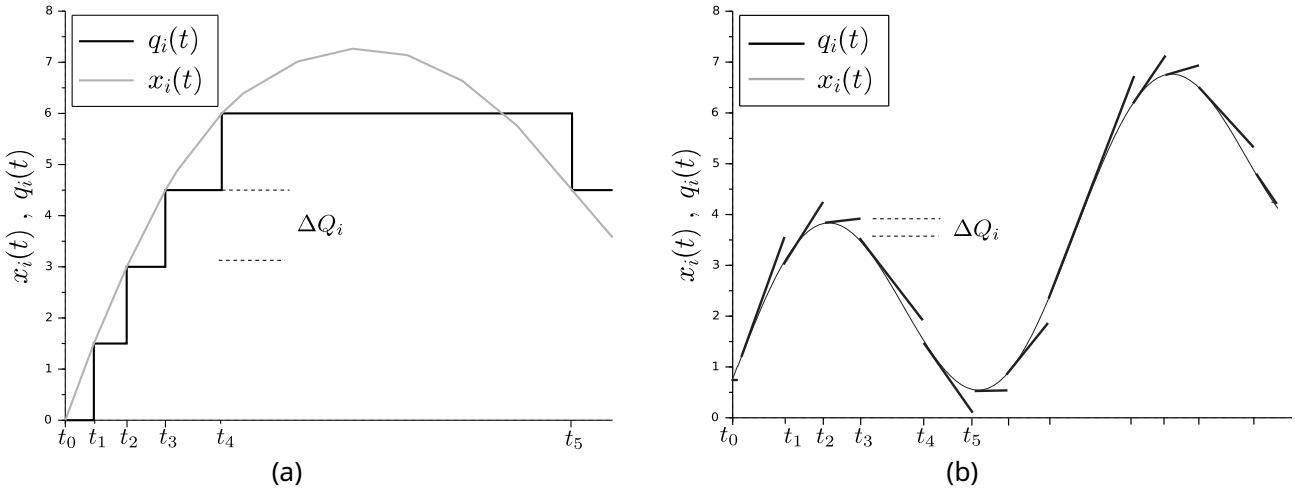


Figure 1. Typical state and quantized state trajectories: (a) QSS1, (b) QSS2

Given the system of Eq. (1), the first order Quantized State System (QSS1) method[4] approximates it by

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{q}(t), \mathbf{u}_q(t)) \tag{2}$$

where $\mathbf{q} = [q_1, q_2, \ldots, q_n]^T$ is the vector of *quantized states* and $\mathbf{u}_q = [u_1, u_2, \ldots, u_m]^T$ is a piecewise constant approximation of the input vector $\mathbf{u}(t)$. Each quantized state $q_i(t)$ is related to its corresponding state $x_i(t)$ by a *hysteretic quantization function*. We say that $x_i(t)$ and $q_i(t)$ are related by a hysteretic quantization function with *quantum* ΔQ_i if $\mathbf{q}(t_0) = \mathbf{x}(t_0)$ and they satisfy:

$$q_i(t) = \begin{cases} q_i(t_k) & \text{if } |x_i(t) - q_i(t_k)| < \Delta Q_i \\ x_i(t) & \text{otherwise} \end{cases}$$

for $t_k < t \leq t_{k+1}$, where t_{k+1} is the first time after t_k such that $|x_i(t) - q_i(t_k)| = \Delta Q_i$. Figure 1(a) illustrates the trajectories of a state and the corresponding quantized state.

Since QSS1 is only first order accurate and a good accuracy cannot be obtained without significantly increasing the number of steps, a second order accurate method called QSS2 was proposed [10].

QSS2 shares the same conceptual definition as QSS1, except that the components of $\mathbf{q}(t)$ are calculated to follow piecewise linear trajectories (rather than piecewise constant, as in QSS1), as shown in Figure 1(b). As a result, the state derivatives $\dot{\mathbf{x}}(t)$ are computed as piecewise linear trajectories, leading the states $\mathbf{x}(t)$ to follow piecewise parabolic trajectories. Formally, we say that the trayectories $x_i(t)$ and $q_i(t)$ are related by a first-order quantization function if $q_i(t_0) = x_i(t_0)$ and if they satisfy:

$$q_i(t) = \begin{cases} q_i(t_k) + m_k(t - t_k) & \text{if } |x_i(t) - q_i(t_k)| < \Delta Q_i \\ x_i(t) & \text{otherwise} \end{cases}$$

with $t_k \leq t \leq t_{k+1}$ and the sequence $t_0, ..., t_k, ...$ defined so that t_{k+1} is the minimum $t \geq t_k$ where

$$|x_i(t_k) + m_k(t - t_k) - x_i(t)| = \Delta Q_i$$

and the slopes:

$$m_0 = 0, \qquad m_k = \dot{x}_i(t_k^-), \quad k = 1, 2...$$

This idea was further extended in a similar way to develop QSS3[11], a third order accurate QSS method in which the quantized states $q_i(t)$ follow piecewise parabolic trajectories while the states follow piecewise cubic trajectories.

The family of QSS methods also includes a set of algorithms called Linearly Implicit QSS (LIQSS), which are appropriate to simulate some stiff systems[5]. LIQSS methods combine the principles of QSS methods with those of classic linearly implicit solvers. There are LIQSS algorithms that perform first, second, and third order accurate approximations: LIQSS1, LIQSS2, and LIQSS3,

respectively. The main idea behind LIQSS methods is inspired by classic implicit methods that evaluate the state derivatives at future instants of time.

QSS offers some practical benefits over conventional time-discretization algorithms. A key feature is the localized nature of state updates: a state variable x_i is only updated when the deviation from its quantized value q_i exceeds the threshold ΔQ_i. These updates affect only q_i and those state derivatives $\dot{x}_j = f_j(\mathbf{q}, \mathbf{u}_q)$ for which f_j explicitly depends on q_i. Thus, computations are triggered asynchronously, only when and where significant changes occur in the system.

This local computation nature of QSS leads to practical advantages, particularly in large and sparse systems where dynamic changes are confined to a limited subset of variables. In such scenarios, like those encountered e.g. in advection-diffusion-reaction models characterized by steep traveling wavefronts, QSS methods naturally restrict computations to the regions where activity occurs, thereby reducing the overall computational cost. Moreover, QSS is especially well-suited to systems with discontinuous dynamics. Thanks to the piecewise polynomial structure of the state trajectories, discontinuities can be detected directly and handled without requiring global restarts or reinitialization procedures. These properties make QSS a flexible and efficient tool for simulating a wide variety of systems, particularly when changes in the system are both sparse and localized.

In addition, QSS methods are distinguished by having a global error bound (at least in linear systems) that depends linearly on the value of the quantum[3] $\Delta\mathbf{Q} = [\Delta Q_1, \ldots, \Delta Q_N]^T$. That way, the quantum in QSS plays an role equivalent to that of the error tolerance in variable-step classic ODE solvers.

2.3 Activity of order n

The notion of activity for continuous-time signals was first introduced to measure the rate of change of such signals[6]. The formal definition of the activity associated with a continuous signal $x_i(t)$ with integrable derivative, evaluated over a time interval $[t_0, t_f]$, is expressed as follows:

$$A_{x_i(t_0,t_f)} \triangleq \int_{t_0}^{t_f} \left| \frac{dx_i(\tau)}{d\tau} \right| \cdot d\tau \tag{3}$$

This metric captures the cumulative variation of a signal by quantifying the distance between successive maxima and minima. Notice that in a monotonic signal the expression of Eq.(3) simply computes the distance from the maximum to the minimum, while in general cases, it accumulates these distances between successive monotonic segments. In this way, the activity of a signal can be related to the number of intervals required in a piecewise constant approximation of it, such as the one depicted in Figure 1(a).

To extend this concept and enable the analysis of higher-order approximations—such as those illustrated in Figure 1(b)—the original definition was generalized in[8] through the introduction of the n-th order activity. This refined metric incorporates information not only from the signal's values but also from its time derivatives up to order n. Formally, for a given signal $x_i(t)$, the n-th order activity over the interval $[t_0, t_f]$ is defined as:

$$A^{(n)}_{x_i(t_0,t_f)} \triangleq \int_{t_0}^{t_f} \left| \frac{1}{n!} \cdot \frac{d^n x_i(\tau)}{d\tau^n} \right|^{1/n} \mathrm{d}\tau \tag{4}$$

Using this definition, it is possible to estimate the number of segments of polynomials up to order $n-1$ that are needed to approximate the signal $x_i(t)$ with an error less than ΔQ_i in the interval $[t_0, t_f]$ as:

$$\begin{aligned} k \approx \left(\frac{1}{\Delta Q_i} \right)^{1/n} \int_{t_0}^{t_f} \left| \frac{1}{n!} \cdot \frac{d^n x_i(\tau)}{d\tau^n} \right|^{1/n} \mathrm{d}\tau = \\ = \frac{A^{(n)}_{x_i(t_0,t_f)}}{(\Delta Q_i)^{1/n}} \end{aligned} \tag{5}$$

This estimate considers that each segment of the approximating polynomial starts with the same value and derivatives (up to order $n-1$) than the signal $x_i(t)$.

In the context of a QSS method of order n (which uses polynomials of order up to $n-1$ in the quantized states), this expression provides a theoretical estimate of the minimum number of integration steps required to

approximate the i-th state variable when a quantization size ΔQ_i is specified.

3 Main Results

This section presents the main results of the work. We begin with a simulation example that motivates the introduction of the Activity Homogeneity concept, which constitutes the core contribution of this study. To define it, we first formulate the notions of instantaneous, local, and global activity, explaining their significance and how they can be used to derive lower bounds on the number of steps required by both QSS and classic discrete-time numerical algorithms.

Then, the Homogeneity Factor is formally defined and its relationship with the computational costs of using classic and quantization-based approaches is analyzed. Finally, the analysis is extended to systems that exhibit discontinuities.

3.1 Motivating Example

Consider the following set of ODEs, representing a coarse spatial discretization of a one-dimensional advection-diffusion-reaction (ADR) model:

$$\dot{x}_i(t) = -A \cdot \frac{x_i(t) - x_{i-1}(t)}{\Delta x} + D \cdot \frac{x_{i+1}(t) - 2x_i(t) + x_{i-1}(t)}{\Delta x^2} + R \cdot (x_i(t)^2 - x_i(t)^3) \quad (6)$$

for $i = 2, 3, \cdots, N-1$, where A, D and R are the advection, diffusion and reaction parameters, respectively.

The first ($i = 1$) and last ($i = N$) points of the one dimensional grid respond the following equations:

$$\dot{x}_1(t) = -A \cdot \frac{x_1(t) - 1}{\Delta x} + D \cdot \frac{x_2(t) - 2x_1(t) + 1}{\Delta x^2} + R \cdot (x_1(t)^2 - x_1(t)^3) \quad (7)$$

$$\dot{x}_N(t) = -A \cdot \frac{x_N(t) - x_{N-1}(t)}{\Delta x} + D \frac{2x_{N-1}(t) - 2x_N(t)}{\Delta x^2} + R \cdot (x_N(t)^2 - x_N(t)^3) \quad (8)$$

Figure 2a shows the state trajectories obtained from simulating the system for $N = 4$, with parameters $A = 1$, $D = 1$, $R = 0.1$, length $L = 10$ and $\Delta x = L/(N-1)$, while Figure 2b shows the results using the same parameters except for R which is set to 1000. We can observe that in the first case most of the time there are at least two variables exhibiting noticeable simultaneous changes, while in the second case there are no variables exhibiting simultaneous changes. We will say that in the first case the trajectories evolve in a more *homogenous* way while in the second case the evolution is more *heterogeneous*.

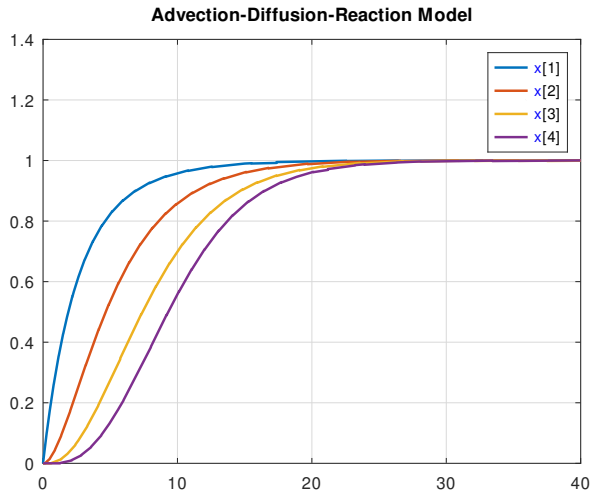
We simulated both models using two different second-order numerical integration methods: a variable step version of the classic Trapezoidal Rule (TR12), and LIQSS2, a state quantization-based integration method. Table 1 reports the number of steps performed by each method to complete the simulations using an absolute tolerance (for TR12) and a quantum size (for LIQSS2) of 10^{-3}.

Table 1. Number of steps required to simulate the ADR model.

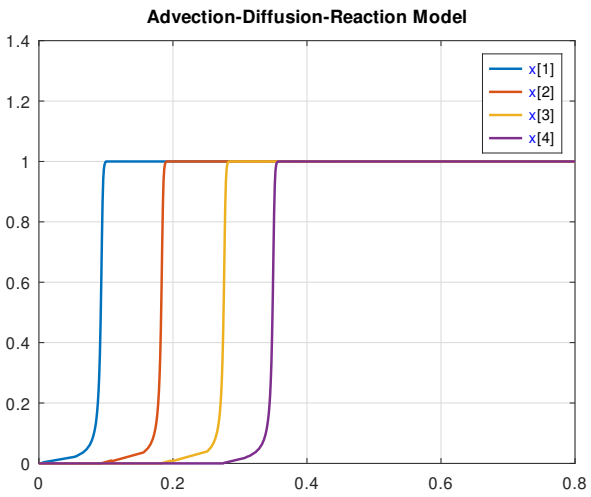
Case	Nr. Steps LIQSS2	Nr. Steps TR12
Case 1 (R=0.1)	101	78
Case 2 (R=1000)	143	272

As seen in the table, for the first case ($R = 0.1$), the TR12 completed the simulation with fewer steps than LIQSS2. However, for the second case ($R = 1000$), LIQSS2 required significantly fewer steps than TR12.

We will show that this observation leads to a general conclusion: when a system evolves *homogeneously* (i.e., when the individual state trajectories exhibit similar rates of change) it is generally more efficient to update all variables simultaneously (i.e. synchronously) as it is done in classic integration methods. Otherwise, when state trajectories show dissimilar rates of change, we say that the system evolves *heterogenously*, and updating variables



(a) ADR system with $\Delta x = 10/3$, $A = 1$, $D = 1$, $R = 0.1$



(b) ADR system with $\Delta x = 10/3$, $A = 1$, $D = 1$, $R = 1000$

Figure 2. ADR trajectories for different parameter settings.

independently (i.e. asynchronously, as in QSS methods) is usually more convenient.

In order to formalize the previous claim, we shall extend the concept of *activity*, since its current definition does not allow measuring the homogeneity of the system trajectories and, consequently, does not allow the estimation of the number of steps required by classic integration algorithms.

3.2 Instantaneous, Local and Global Activity

Considering the definition of the n-th order activity of a signal $x_i(t)$ in a given interval $[t_0, t_f]$ presented en Section 2.3, we define the *local instantaneous activity of order* n of the signal as the integrand of Equation (4) given by

$$a_{x_i}^{(n)}(t) \triangleq \left| \frac{1}{n!} \cdot \frac{\mathrm{d}^n x_i(t)}{\mathrm{d}t^n} \right|^{1/n} \tag{9}$$

and define the local activity of order n of $x_i(t)$ in the time interval $[t_0, t_f]$ as

$$A_{x_i(t_0,t_f)}^{(n)} \triangleq \int_{t_0}^{t_f} a_{x_i}^{(n)}(t)\mathrm{d}t \quad \text{with } i = 1, 2, \ldots, N \tag{10}$$

which coincides with the original definition of *activity* given in[8] and can be related to the minimum number of steps performed by QSS methods.

In order to obtain a measure related to the number of steps required by classic discrete time methods, we extend this idea by defining the *global instantaneous activity* of order n as the maximum *local instantaneous activity* among all system states:

$$\begin{aligned} a_{\mathbf{x}}^{(n)}(t) &\triangleq \max_i \left\{ a_{x_i}^{(n)}(t) \right\} = \\ &= \max_i \left\{ \left| \frac{1}{n!} \cdot \frac{\mathrm{d}^n x_i(t)}{\mathrm{d}t^n} \right|^{1/n} \right\} \end{aligned} \tag{11}$$

Then, we define the *global activity* of order n for the state trajectories vector $\mathbf{x}(t)$ in an interval $[t_0, t_f]$ by integrating the global instantaneous activity: *

$$A_{\mathbf{x}(t_0,t_f)}^{(n)} \triangleq \int_{t_0}^{t_f} a_{\mathbf{x}}^{(n)}(t)\mathrm{d}t \tag{12}$$

Table 2. Summary of Activity definitions.

Activity type	Symbol	Expression
local instantaneous	$a_{x_i}^{(n)}(t)$	$\left\|\frac{1}{n!} \cdot \frac{\mathrm{d}^n x_i(t)}{\mathrm{d}t^n}\right\|^{1/n}$
local	$A_{x_i(t_0,t_f)}^{(n)}$	$\int_{t_0}^{t_f} a_{x_i}^{(n)}(t)\mathrm{d}t$
global instantaneous	$a_{\mathbf{x}}^{(n)}(t)$	$\max_i \left\{ a_{x_i}^{(n)}(t) \right\}$
global	$A_{\mathbf{x}(t_0,t_f)}^{(n)}$	$\int_{t_0}^{t_f} a_{\mathbf{x}}^{(n)}(t)\mathrm{d}t$

*For notational simplicity, the explicit dependency on the time interval (t_0, t_f) will occasionally be omitted.

3.3 Homogeneity Factor

When all the states evolve with a similar rate of change (i.e., in an *homogeneous* way), it can be easily noticed that the local instantaneous activity $a_{x_i}^{(n)}(t)$ of all state variables will be close to the maximum local instantaneous activity, which coincides with the global instantaneous activity $a_{\mathbf{x}}^{(n)}(t)$. Extending this analysis to a given interval, we conclude that the local activity $A_{x_i(t_0,t_f)}^{(n)}$ of every state is close to the global activity $A_{\mathbf{x}(t_0,t_f)}^{(n)}$.

On the contrary, when the system trajectories are *heterogeneous*, the global activity will be significantly larger than most local activities. Thus, it makes sense to define the n-th order *Homogeneity Factor* (HF) as:

$$H_{\mathbf{x}(t_0,t_f)}^{(n)} \triangleq \frac{\sum_{i=1}^{N} A_{x_i(t_0,t_f)}^{(n)}}{N \cdot A_{\mathbf{x}(t_0,t_f)}^{(n)}} \tag{13}$$

Notice that:

- $H_{\mathbf{x}}^{(n)}$ only takes values in the interval $[\frac{1}{N}, 1]$.
- One extreme case occurs when all state variables evolve with identical activity. In that scenario, we have $A_{x_i}^{(n)} = A_{\mathbf{x}}^{(n)}$ for all i and, from Eq.(13) it results $H_{\mathbf{x}}^{(n)} = 1$. In that case we have the maximum possible homogeneity.
- The other extreme case occurs when only one state variable exhibits activity. In this situation, we have $A_{x_i}^{(n)} = A_{\mathbf{x}}^{(n)}$ for that variable and $A_{j(t_0,t_f)}^{(n)} = 0$ for the remaining states. Then, from Eq.(13) it results $H_{\mathbf{x}}^{(n)} = 1/N$. In that case we have the minimum possible homogeneity.
- A similar situation arises when different states exhibit activity but not simultaneously (i.e., the instantaneous activity of all but one state is always null). In that case, it results also $H_{\mathbf{x}}^{(n)} = 1/N$.
- In general, a HF close to 1 indicates homogeneous activity, while a value close to $1/N$ corresponds to heterogeneous activity.

3.4 Relation between the activity and the number of steps

The computational cost of an ODE system simulation is intrinsically linked to the number of steps and calculations required by a numerical integration method to achieve a given level of accuracy (typically a requisite demanded by the user) in the approximation of the state variables.

The following theorem, whose proof can be found in A.2, will be useful for relating the activity of a system to the minimum number of steps required for its simulation:

Theorem 1. *Given a signal $x_i(t)$ expressed as a polynomial of degree n, there exists a polynomial $q_i(t)$ of degree less than or equal to $n-1$ satisfying the condition $|q_i(t) - x_i(t)| \leq \Delta Q_i$ in an interval $[t_j, t_j + \Delta t]$ if and only if*

$$\Delta t \leq \frac{2^{\frac{2n-1}{n}} \Delta Q_i^{\frac{1}{n}}}{a_{x_i}^{(n)}(t_j)} \tag{14}$$

where $a_{x_i}^{(n)}(t_j)$ is the local instantaneous activity of order n of the signal $x_i(t)$ in $t = t_j$.

3.4.1 Computational Costs in Quantization-Based Methods .

Equation (14), which holds for any time interval $[t_j, t_j + \Delta t]$, can be rewritten as follows

$$\frac{a_{x_i}^{(n)}(t_j) \cdot \Delta t}{2^{\frac{2n-1}{n}} \Delta Q_i^{\frac{1}{n}}} \leq 1 \tag{15}$$

Consider that a n-th order QSS simulation of Eq.(2) from time t_0 to t_f updates the quantized state $q_i(t)$ at times $t_0, t_1, \ldots, t_{k-1} \leq t_f$. In each interval of the form $[t_j, t_{j+1})$, the quantized state $q_i(t)$ follows a polynomial of degree $n-1$, while the state $x_i(t)$ follows a polynomial of degree n. Since the difference between x_i and q_i is bounded by ΔQ_i, then Eq.(15) holds for all $0 \leq j \leq k-1$. Summing both sides of that inequality from $j = 0$ to $j = k-1$ we obtain:

$$\sum_{j=0}^{k-1} \frac{a_{x_i}^{(n)}(t_j) \cdot (t_{j+1} - t_j)}{2^{\frac{2n-1}{n}} \Delta Q_i^{\frac{1}{n}}} \leq \sum_{j=0}^{k-1} 1 = k \tag{16}$$

Since $a_{x_i}^{(n)}$ is constant in every interval $[t_j, t_{j+1}]$, then the above sum is equal to an integral, and we have

$$\frac{1}{2^{\frac{2n-1}{n}}\Delta Q_i^{\frac{1}{n}}}\int_{t_0}^{t_f} a_{x_i}^{(n)}(t)\mathrm{d}t \leq k \tag{17}$$

and replacing the integral by the local activity definition of Eq.(10) it results

$$\frac{A_{x_i}^{(n)}}{2^{\frac{2n-1}{n}}\Delta Q_i^{\frac{1}{n}}} \leq k \tag{18}$$

Thus, the minimum number of segments of $q_i(t)$ used by a QSS method of order n in the time interval $[t_0, t_f]$ can be approximated by

$$k_{x_i(t_0,t_f)}^{(n)}(\Delta Q_i) \triangleq \frac{A_{x_i}^{(n)}}{2^{\frac{2n-1}{n}} \cdot (\Delta Q_i)^{\frac{1}{n}}} \tag{19}$$

Since QSS methods update each state independently, a lower bound for the number of steps required to simulate the whole system can be approximated by

$$\begin{aligned} k_{\mathrm{QS}(t_0,t_f)}^{(n)}(\Delta \mathbf{Q}) &\triangleq \sum_{i=1}^{N} k_{x_i(t_0,t_f)}^{(n)}(\Delta Q_i) = \\ &= \sum_{i=1}^{N} \frac{A_{x_i}^{(n)}}{2^{\frac{2n-1}{n}} \cdot (\Delta Q_i)^{\frac{1}{n}}} \end{aligned} \tag{20}$$

where $\Delta \mathbf{Q} = [\Delta Q_1, \ldots, \Delta Q_N]^T$.

During each step, a QSS method performs calculations not only in the state x_i that changes its quantized value, but also in the states x_j whose derivatives depend on x_i. Then, denoting r_i as the number of states whose derivatives depend on x_i, the total number of updates performed by a QSS method can be lower bounded by

$$\begin{aligned} u_{\mathrm{QS}(t_0,t_f)}^{(n)}(\Delta \mathbf{Q}) &\triangleq \sum_{i=1}^{N} k_{x_i(t_0,t_f)}^{(n)}(\Delta Q_i) \cdot r_i = \\ &= \frac{1}{2^{\frac{2n-1}{n}} \cdot (\Delta Q)^{\frac{1}{n}}} \sum_{i=1}^{N} A_{x_i}^{(n)} \cdot r_i \end{aligned} \tag{21}$$

where the last step is only valid when $\Delta Q_i = \Delta Q$ for $i = 1, \ldots, N$.

It is worth mentioning that Eq.(19) provides a smaller bound than Eq.(5). The reason is that the expression of Eq.(5) was only valid for the original QSS methods where each segment of q_i started equal to x_i, while the new expression of Eq.(19) covers the general case. In LIQSS algorithms, for instance, the segments of q_i start at a distance ΔQ_i from x_i which allow to perform larger steps until the distance to x_i becomes again equal to ΔQ_i.

3.4.2 Computational Costs in Time Discretization-Based Methods

Classic step size control algorithms estimate the local error by subtracting a solution given by a method of order $n-1$ from a solution given by a method of order n. If this error estimate is larger than the error tolerance, then the step is repeated with a smaller step size. Otherwise, it is accepted (and the step size is increased).

After each step, the difference between the solutions of the methods of order $n-1$ and n can be approximated as the difference between two polynomials of the corresponding orders[3]. If the error tolerance is the same for all states and it is denoted as ΔQ_i, then the step size Δt will verify Eq.(14) for $i = 1, \ldots, N$. Then, the step size will be upper bounded by the component i that gives the smaller bound for Δt, i.e.,

$$\Delta t \leq \frac{2^{\frac{2n-1}{n}}\Delta Q_i^{\frac{1}{n}}}{\max_i(a_{x_i}^{(n)}(t_j))} = \frac{2^{\frac{2n-1}{n}}\Delta Q_i^{\frac{1}{n}}}{a_{\mathbf{x}}^{(n)}(t_j)} \tag{22}$$

From this equation, by repeating the procedure followed for QSS methods, we obtain the following expression that approximates the lower bound for the number of steps performed by a classic method

$$k_{\mathrm{DT}(t_0,t_f)}^{(n)}(\Delta Q) \triangleq \frac{A_{\mathbf{x}}^{(n)}}{2^{\frac{2n-1}{n}} \cdot (\Delta Q)^{\frac{1}{n}}} \tag{23}$$

Since discrete time algorithms update all N states in each step, the total number of state updates can be lower bounded by

$$\begin{aligned} u_{\mathrm{DT}(t_0,t_f)}^{(n)}(\Delta Q) &\triangleq N \cdot k_{\mathrm{DT}(t_0,t_f)}^{(n)}(\Delta Q) = \\ &= N \cdot \frac{A_{\mathbf{x}}^{(n)}}{2^{\frac{2n-1}{n}} \cdot (\Delta Q)^{1/n}} \end{aligned} \tag{24}$$

3.5 Computational Costs and Homogeinity Factor

Let us consider a system where the number of state derivatives that depend on every state are the same, i.e., $r_i = r$ for $i = 1, \dots, N$ (note that r denotes the degree of interconnection between states). Suppose also that we use the same quantization $\Delta Q_i = \Delta Q$ for all states in a QSS simulation and that it coincides with the tolerance of a classic discrete time algorithm. Then, using Equations (13), (21) and (24) we can relate the minimum number of variable updates of both algorithms with the Homogeneity Factor as

$$\frac{u^{(n)}_{\mathrm{QS}}}{u^{(n)}_{\mathrm{DT}}} = H^{(n)}_{\mathbf{x}} \cdot r \tag{25}$$

From this expression, we can draw the following conclusions:

- If the product $H^{(n)}_{\mathbf{x}} \cdot r > 1$, then the lower bound for the number of state updates of classic methods is smaller than that of QSS methods. Thus, we can expect classic methods to be more efficient than quantization based approaches.
- Conversely, whenever $H^{(n)}_{\mathbf{x}} \cdot r < 1$ we can expect QSS methods to be more efficient.
- If the system trajectories are homogeneous ($H^{(n)}_{\mathbf{x}} \approx 1$) it will result in $H^{(n)}_{\mathbf{x}} \cdot r > 1$ (unless the system is totally disconnected with $r = 1$). Thus, classic discrete time algorithms are expected to be more efficient.
- If the system trajectories are heterogeneous ($H^{(n)}_{\mathbf{x}} \approx 1/N$) it will result in $H^{(n)}_{\mathbf{x}} \cdot r < 1$ (unless the system is fully connected with $r = N$). Thus, quantization-based algorithms are expected to be more efficient.
- When the system is very sparse (i.e., $r \approx 1$) it will result in $H^{(n)}_{\mathbf{x}} \cdot r < 1$ (unless the trajectories are very homogeneous and $H^{(n)}_{\mathbf{x}} \approx 1$). Then, QSS methods are expected to be convenient.
- When the system is very connected (i.e., $r \approx N$) it will result in $H^{(n)}_{\mathbf{x}} \cdot r > 1$ (unless the trajectories are very heterogeneous and $H^{(n)}_{\mathbf{x}} \approx 1/N$). Then, classic algorithms are expected to be convenient.

These conclusions, which are a direct consequence of Eq.(25), corroborate what was observed in practice and mentioned in various previous works regarding the convenience of using QSS methods in sparse systems with heterogeneous activity. The analysis, however, has certain limitations:

- We considered that every state variable appears exactly r times on the right hand side of Eq.(1). Anyway, we can take r as a weighted average value of r_i. Notice that, from Eq.(21), it results

$$u^{(n)}_{\mathrm{QS}(t_0,t_f)}(\Delta\mathbf{Q}) = \sum_{i=1}^{N} k^{(n)}_{x_i(t_0,t_f)}(\Delta Q_i) \cdot r^{(n)}$$

  where we defined

$$r^{(n)} \triangleq \frac{\sum_{i=1}^{N} k^{(n)}_{x_i(t_0,t_f)}(\Delta Q_i) \cdot r_i}{\sum_{i=1}^{N} k^{(n)}_{x_i(t_0,t_f)}(\Delta Q_i)}$$

  i.e.

$$r^{(n)} = \sum_{i=1}^{N} \frac{k^{(n)}_{x_i(t_0,t_f)}(\Delta Q_i)}{k^{(n)}_{\mathrm{QS}(t_0,t_f)}(\Delta\mathbf{Q})} \cdot r_i = \sum_{i=1}^{N} c_i r_i \tag{26}$$

  where the weighting constants c_i are the ratio of steps performed by each state variable. In case $\Delta Q_i = \Delta Q$ for all i, the expression of Eq.(26) can be rewritten as

$$r^{(n)} = \sum_{i=1}^{N} \frac{A^{(n)}_{x_i(t_0,t_f)}}{\sum_{i=1}^{N} A^{(n)}_{x_i(t_0,t_f)}} \cdot r_i = \sum_{i=1}^{N} c_i r_i \tag{27}$$

  which gives an expression for c_i that does not depend on the quantum.
- We arrived at Eq.(25) by assuming that the quantum was set as $\Delta Q_i = \Delta Q$ for every state in QSS methods[†]. In addition, we considered that ΔQ is constant. In practice, logarithmic quantization is mostly used where ΔQ_i increases with the signal $x_i(t)$. Equivalently, in classic discrete time methods

[†]Notice that the use of different values for ΔQ_i can be avoided by introducing appropriate change of variables of the form $z_i = x_i/\Delta Q_i$.

we only considered the absolute tolerance. Thus, if relative tolerances are taken into account with signals having large variations in magnitude, the expression of Eq.(25) may not be valid.

- In any case, the expression of Eq.(25) relates the lower bounds for the number of state updates. The actual number of steps can be much larger when the systems are stiff, for instance, and the algorithms are not appropriate for that.

- In practice, the computation of the Homogeneity Factor requires the computation of the local and global activities, which in turn requires the knowledge of the system trajectories[‡]. These trajectories are only known after simulating the system, thus we would not be able to use the concept in advance to decide which approach is better. However, we can compute the value of $H_{\mathbf{x}}^{(n)}$ for a certain particular system and then use it as an approximate value for systems that exhibit similar behavior. It is important to remark that most conclusions listed above do not depend on the exact knowledge of the HF.

- A possible way to obtain an approximate value for $H_{\mathbf{x}}^{(n)}$ and decide about the most appropriate algorithm is to perform a quick *coarse* simulation using a QSSn algorithm with a large quantum exploiting the fact that these algorithms preserve stability irrespective of the quantum size.

- The HF could be also computed during the simulation (starting with either a QSS or a Discrete Time approach) and then used to switch the algorithm according to its value. This idea could lead to something similar to the automatic algorithm selection already used in classic ODE solvers [12].

3.6 Systems with Discontinuities

The analysis performed above assumes that the state variables follow differentiable trajectories. We will now extend these notions to take into account the presence of discontinuities.

For that goal, we will consider a hybrid system of the form

$$\dot{\mathbf{x}}_a(t) = \mathbf{f}(\mathbf{x}_a(t), \mathbf{z}(t), \mathbf{u}(t)) \tag{28}$$

where the new variable $\mathbf{z}(t) \in \mathbb{R}^p$ represents the discrete state. We will say that a discontinuity of type i occurs, with $i \in \{1, \ldots, D\}$, whenever the condition

$$g_i(\mathbf{x}_a(t), \mathbf{z}(t), \mathbf{u}(t)) = 0 \tag{29}$$

is met. Here, $g_i : \mathbb{R}^{N+p+m} \to \mathbb{R}$ are the *zero-crossing functions*. After the i–th zero crossing condition is met, the states are recomputed as

$$[\mathbf{x}_a(t^+), \mathbf{z}(t^+)] = H_i(\mathbf{x}_a(t), \mathbf{z}(t), \mathbf{u}(t)) \tag{30}$$

where $H_i : \mathbb{R}^{N+p+m} \to \mathbb{R}^{N+p}$ is the i–th *event handler*.

A solution of this hybrid system $\mathbf{x}_a(t)$ verifies Eq.(28) almost everywhere, except at event times, i.e., when a zero crossing condition of Eq.(29) is verified.

We will suppose that the discontinuity of type i provokes instantaneous changes in s_i state derivatives (due to the event handler H_i and the structure of function $\mathbf{f}$). Consider also that the discontinuity of type i occurs d_i times in the interval $[t_0, t_f]$, at event times $t_1^i, t_2^i, \ldots, t_{d_i}^i$. Thus, the total number of discontinuities in that interval is $d = \sum_{i=1}^{D} d_i$.

In order to compute the number of steps performed by quantization-based and discrete time algorithms in this case, we will consider the way they treat discontinuities. At every event time t_j^i, a QSS method will update the s_i state derivatives affected by the i–th type discontinuity. Similarly, a classic discrete time method will perform a step at the event time and restart the simulation computing the entire state. Thus, in both cases, the presence of a discontinuity triggers computations similar to those of a regular step.

Therefore, in order to compute the total number of steps, we must add those corresponding to the continuous integration (related to the activity as in the continuous case) with those provoked by the presence of discontinuities. Then, from Eqs.(20) and (23), the minimum total number of

[‡] In practice, global and local activities can be directly computed from the numerical results provided that the ODE solvers produce dense output.

steps required by each type of method can be approximated by the following quantities:

- State Quantization:

$$k_{\mathrm{QS}}^{(n)}(\Delta\mathbf{Q}) \triangleq \sum_{i=1}^{N} \frac{A_{x_i}^{(n)}}{2^{\frac{2n-1}{n}} \cdot (\Delta Q_i)^{\frac{1}{n}}} + d \qquad (31)$$

- Time Discretization

$$k_{\mathrm{DT}}^{(n)}(\Delta Q) \triangleq \frac{A_{\mathbf{x}}^{(n)}}{2^{\frac{2n-1}{n}} \cdot (\Delta Q)^{\frac{1}{n}}} + d \qquad (32)$$

Regarding the total number of updates, taking into account that discontinuities trigger regular steps in both cases, we can reformulate Eqs. (21) and (24) as:

- State Quantization:

$$u_{\mathrm{QS}}^{(n)}(\Delta\mathbf{Q}) \triangleq \sum_{i=1}^{N} k_{x_i}^{(n)}(\Delta Q_i) \cdot r_i + \sum_{i=1}^{D} d_i \cdot s_i \qquad (33)$$

- Time Discretization

$$u_{\mathrm{DT}}^{(n)}(\Delta Q) \triangleq N \cdot k_{\mathrm{DT}}^{(n)}(\Delta Q) + d \cdot N \qquad (34)$$

Assuming that $r_i = r$ for $i = 1, \dots N$ (or taking r as the weighted average of r_i) and $s_i = s$ (or taking s as the average of s_i), we can approximate the relation between the minimum number of updates required by each type of method as:

$$\frac{u_{\mathrm{QS}}^{(n)}}{u_{\mathrm{DT}}^{(n)}} = \frac{\sum_{i=1}^{N} \left(k_{x_i}^{(n)} \cdot r\right) + \sum_{i=1}^{D} (d_i \cdot s)}{N \cdot k_{\mathrm{DT}}^{(n)} + d \cdot N} = \\ = \frac{r \cdot k_{\mathrm{QS}}^{(n)} + d \cdot s}{N \cdot k_{\mathrm{DT}}^{(n)} + d \cdot N} \qquad (35)$$

From this expression, we can draw the following conclusions:

- If the number of discontinuities is very small the terms $d \cdot s$ and $d \cdot N$ can be neglected in Eq.(35) resulting

$$\frac{u_{\mathrm{QS}}^{(n)}}{u_{\mathrm{DT}}^{(n)}} \approx H_{\mathbf{x}}^{(n)} \cdot r$$

and the convenience of QSS or classic discrete time approaches depends almost exclusively on the continuous dynamics.

- If the number of discontinuities is very large compared with the steps corresponding to the continuous dynamics, we have

$$d \to \infty \implies \frac{u_{\mathrm{QS}}^{(n)}}{u_{\mathrm{DT}}^{(n)}} \to \frac{s}{N}$$

and quantization based integration algorithms are expected to perform better.

- If QSS methods are expected to perform better for the continuous dynamics, then

$$\frac{r \cdot k_{\mathrm{QS}}^{(n)}}{N \cdot k_{\mathrm{DT}}^{(n)}} < 1 \implies \frac{r \cdot k_{\mathrm{QS}}^{(n)} + d \cdot s}{N \cdot k_{\mathrm{DT}}^{(n)} + d \cdot N} < 1$$

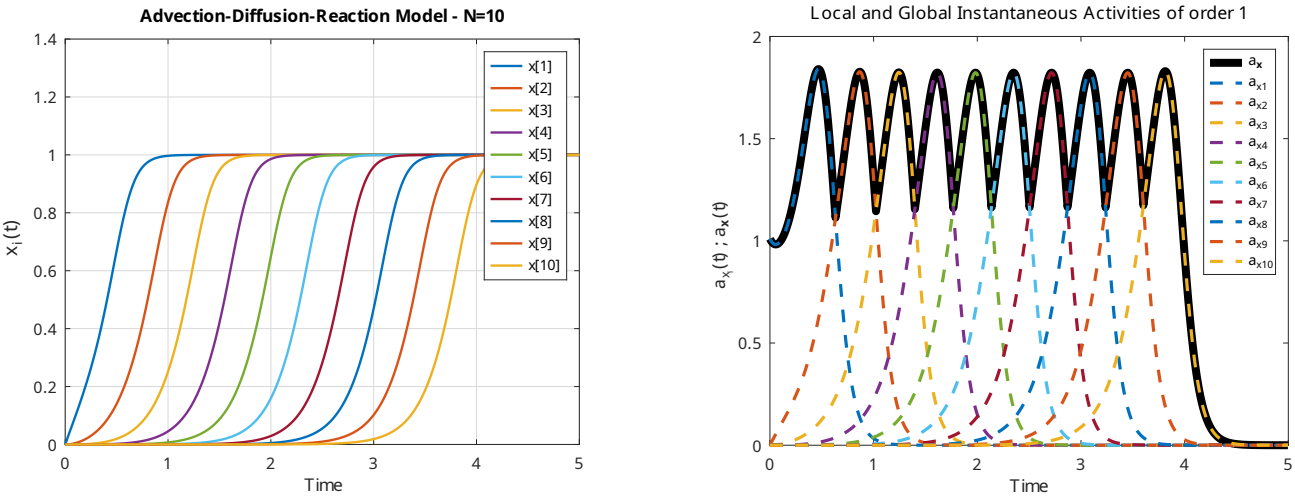
and thus QSS methods will be expected to perform better also in presence of discontinuities.

These new conclusions, that are a direct consequence of Eq.(35), corroborate the practical observation about the convenience of using QSS methods in presence of discontinuities.

It is important to mention that at the event times the instantaneous activity of the states affected by the event given by Eq.(9) is not well defined since the time derivatives of the corresponding states may not exist. This can be solved by setting any value for the instantaneous activity (0, for instance) at the event times, as it will not affect the computation of the local and global activities, which are computed by integrating the instantaneous activities.

4 Examples and Simulation Results

In this section we present two application examples that corroborate the correspondence between the theory developed in the previous section and the simulation results. The first example corresponds to a purely continuous advection-diffusion-reaction (ADR) model while the second one is a spiking neural network model that contains frequent discontinuities.



(a) Numerical solution of ADR system with N=10

(b) Local and Global Instantaneous Activities of order 1

Figure 3. State trajectories and instantaneous activity (local and global) of the ADR model

The simulation results reported below were obtained using the Stand Alone QSS Solver[13] for QSS and CVODE methods. The results of classic algorithms were obtained using Octave. The corresponding algorithms and models can be downloaded from https://github.com/ernestokofman/HomActivity/blob/master/Models_Files.zip in order to reproduce the reported results.

4.1 Advection-Diffusion-Reaction Model

We consider here the ADR system of Eqs. (6)– (8) with the following parameters: $N = 10; A = 1; D = 0.01; R = 10$ and $dx = 10/N = 1$. The system will be simulated from time $t_0 = 0$ to time $t_f = 5$.

4.1.1 Local Activities, Global Activities and Homogeneity Factor .

Since the state trajectories of the system cannot be computed analytically, their local and global activities were calculated from the numerical solution obtained using the LIQSS2 algorithm[5] with a very low tolerance ($\Delta Q = 10^{-12}$) since this method is known to guarantee global error bounds in these types of problems[14].

Figure 3a displays the state trajectories, while Figure 3b shows the corresponding local instantaneous activities of order 1, $a_{x_i}^{(1)}$, computed using Eq.(9). Figure 3b also presents the global instantaneous activity of order 1, $a_{\mathbf{x}}^{(1)}(t)$, which clearly corresponds to the maximum local instantaneous activity at each instant of time.

Integrating $a_{x_i}^{(n)}$ over time according to Eq.(10), for orders $n = 1$, $n = 2$, and $n = 3$, we also obtained the values for the local activity of each state variable. These values are reported in Table 3[§].

Similarly, integrating the global instantaneous activity $a_{\mathbf{x}}^{(n)}(t)$ according to Eq.(12), we computed the global activities of order 1 to 3 obtaining the following values:

$$A_{\mathbf{x}}^{(1)} = 6.2226;\ A_{\mathbf{x}}^{(2)} = 6.2144;\ A_{\mathbf{x}}^{(3)} = 7.5833$$

Using these results, we then computed the Homogeneity Factor $H_{\mathbf{x}}^{(n)}$ of orders 1 to 3 according to Eq.(13) and reported them in Table 4. There, we also computed the value $H_{\mathbf{x}}^{(n)} \cdot r^{(n)}$ using the following values of $r^{(n)}$ obtained according to Eq.(27).

- $r^{(1)} = 2.8$,
- $r^{(2)} = 2.82$,
- $r^{(3)} = 2.82$

It can be seen that the HF is relatively small (i.e., the system trajectories have heterogeneous activity) and also

[§]Notice that the local activity of order 1 of each state is equal to 1 since all trajectories grow monotonically from 0 to 1 experiencing an absolute change of 1.

Table 3. Local activities of order 1 to 3 for the states of the ADR model

n	$A_1^{(n)}$	$A_2^{(n)}$	$A_3^{(n)}$	$A_4^{(n)}$	$A_5^{(n)}$	$A_6^{(n)}$	$A_7^{(n)}$	$A_8^{(n)}$	$A_9^{(n)}$	$A_{10}^{(n)}$
1	1	1	1	1	1	1	1	1	1	1
2	1.150	1.579	1.650	1.678	1.692	1.699	1.704	1.706	1.708	1.705
3	1.712	1.899	2.162	2.247	2.291	2.317	2.335	2.347	2.341	2.328

Table 4. Homogeneity Factors.

n	$H_{\mathbf{x}}^{(n)}$	$H_{\mathbf{x}}^{(n)} \cdot r^{(n)}$
1	0.1606	0.4497
2	0.2618	0.7395
3	0.2898	0.8161

$H_{\mathbf{x}}^{(n)} \cdot r^{(n)} < 1$, suggesting that QSS algorithms can be expected to be more efficient.

4.1.2 *Number of Steps and Updates* .

Based on the local and global activities, we estimated the lower bounds for the number of steps required by QSS and classic algorithms according to Eqs.(20) and (23), using different quantum and accuracy settings. We also simulated the model using LIQSS algorithms of order 1 to 3 (linearly implicit algorithms were used due of the system stiffness) and using classic implicit variable step algorithms of order 1 to 3. In particular, we used Backward Euler (BE), the Trapezoidal Rule (TR), and a third order Backward Runge-Kutta (BRK3). In all cases, we controlled the step-size estimating the error with a method of one less order of approximation giving place to the methods of BE01, TR12 and BRK23.

The lower bound estimates and the actual number of steps performed by the corresponding algorithms are reported in Table 5.

Table 5. Lower bound and actual number of steps using different numerical methods.

ΔQ	LIQSS1		LIQSS2		LIQSS3	
	$k_{\text{QS}}^{(1)}$	Real	$k_{\text{QS}}^{(2)}$	Real	$k_{\text{QS}}^{(3)}$	Real
10^{-2}	500	980	58	103	33	138
10^{-3}	5000	9981	182	243	70	261
10^{-4}	50000	99981	576	729	150	482
ΔQ	BE01		TR12		BRK23	
	$k_{\text{DT}}^{(1)}$	Real	$k_{\text{DT}}^{(2)}$	Real	$k_{\text{DT}}^{(3)}$	Real
10^{-2}	312	657	22	75	12	50
10^{-3}	3113	6556	70	216	24	96
10^{-4}	31130	65535	220	657	52	195

In all cases, the number of steps performed are between 2 and 5 times larger than the lower bounds computed, except for LIQSS2 where the actual number of steps is closer to the theoretical lower bound. In all cases, the methods perform more steps than the lower bound because the algorithm do not follow the *optimal* trajectories of Theorem 1, where the difference between the n–th and the $n-1$–th order trajectories follow a Chebyshev polynomial, as shown in Lemma 1.

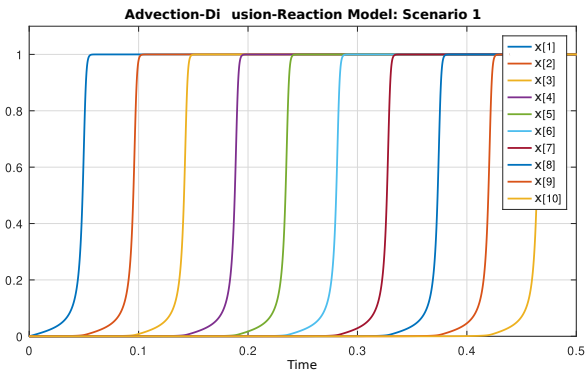
4.1.3 *Homogeneity Factor and state updates* .

The Homogeneity Factor reported in Table 4, and its product with the degree of connectivity $r^{(n)}$, suggest that QSS methods are expected to perform fewer scalar state updates than classic algorithms. In order to corroborate it, we computed the state updates of the different algorithms by multiplying the number of steps by the connectivity $r^{(n)}$ for QSS and by the size of the state vector $N = 10$ for the classic algorithms. The corresponding results are reported in Table 6, where the advantage of QSS methods results greater in practice than what the product $H_{\mathbf{x}}^{(n)} \cdot r^{(n)}$ predicted.

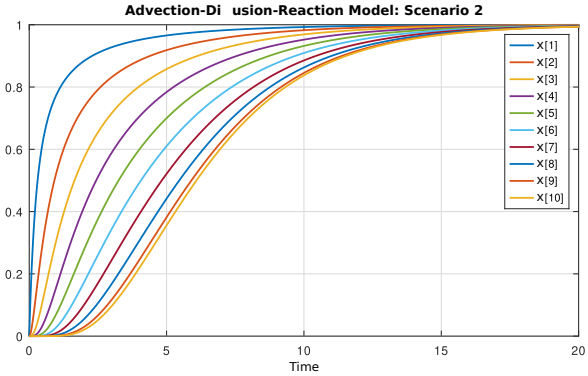
Table 6. Relation between state updates of LIQSS and classic numerical integration algorithms.

	Theoretical results	Simulation results	
n	$H_{\mathbf{x}}^{(n)} \cdot r^{(n)}$	ΔQ	$\frac{u_{QSS}}{u_{DT}}$
1	0.4497	10^{-2}	0.447
		10^{-3}	0.457
		10^{-4}	0.458
2	0.7395	10^{-2}	0.396
		10^{-3}	0.277
		10^{-4}	0.259
3	0.8161	10^{-2}	0.828
		10^{-3}	0.815
		10^{-4}	0.741

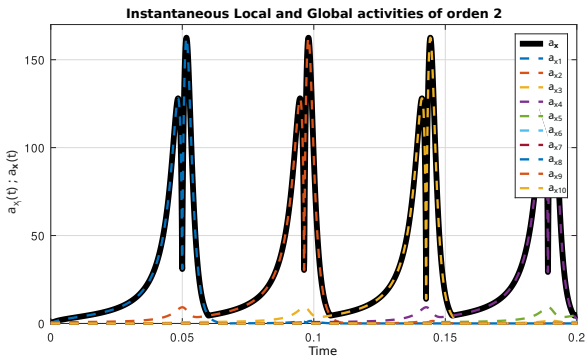
It is important to emphasize that the relationship $H_{\mathbf{x}}^{(n)} \cdot r^{(n)} < 1$ does not imply that a quantization-based method will always result in fewer updates than a discrete-time method. The correct interpretation is that the theoretical



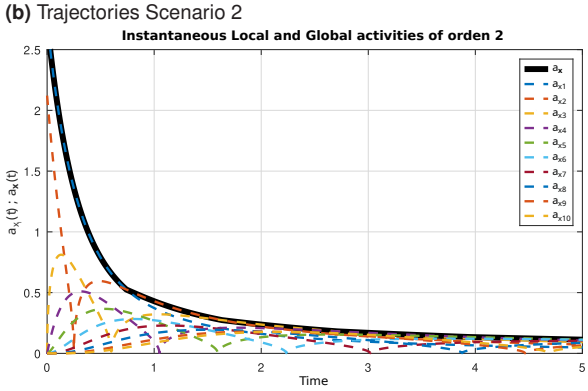
(a) Trajectories Scenario 1



(b) Trajectories Scenario 2



(c) Activity Scenario 1 (detail)



(d) Activity Scenario 2 (detail)

Figure 4. Trajectories and activities for different parameters

lower bound for the number of updates in quantization based methods is less than the theoretical lower bound in discrete-time methods. A reciprocal reasoning applies for the case $H_{\mathbf{x}}^{(n)} \cdot r^{(n)} > 1$.

4.1.4 *Parameter Variation* .

We now consider two *extreme* scenarios for the same ADR model of Eqs.(6)–(8) but using two different parameter sets:

- Scenario 1: $N = 10$, $R = 1000$, $D = 0.1$, $A = 1$ and $\Delta x = 10/N = 1$. Period of simulation $(t_0 = 0, t_f = 0.6)$.
- Scenario 2: $N = 10$, $R = 0.1$, $D = 2$, $A = 1$ and $\Delta x = 10/N = 1$. Period of simulation $(t_0 = 0, t_f = 25)$.

Figures 4a and 4b illustrate the system trajectories for both scenarios, while Figures 4c and 4d show their corresponding instantaneous and global activities of order 2.

In the first case, the trajectories have heterogeneous activity, which can be clearly observed both in the state trajectories (Fig.4a) and in the instantaneous activities (Fig.4c). Conversely, in the second case the trajectories have homogeneous activities.

We then computed the local and global activities of order 2 according to Eqs.(10) and (12). The resulting Homogeneity Factors, calculated from Eq.(13) and the values $H_{\mathbf{x}}^{(2)} \cdot r^{(2)}$ (with $r^{(2)} = 2.8$ for both scenarios) are reported in Table 7

Table 7. Homogeneity Factors.

Scenario	$H_{\mathbf{x}}^{(2)}$	$H_{\mathbf{x}}^{(2)} \cdot r^{(2)}$
1	0.109	0.3052
2	0.567	1.5876

These results indicate that QSS methods are expected to perform better in the first scenario (heterogeneous trajectories), while classic methods are expected to perform better in the second scenario (homogeneous trajectories).

Table 8. Simulation steps and updates and their theoretical lower bounds.

Sc.	Abs.Tol.	LIQSS2				TR12			
		Steps		Updates		Steps		Updates	
	(ΔQ)	$k_{\text{QS}}^{(2)}$	Real	$u_{\text{QS}}^{(2)}$	Real	$k_{\text{DT}}^{(2)}$	Real	$u_{\text{DT}}^{(2)}$	Real
1	10^{-2}	68	157	191	468	62	225	620	2250
	10^{-3}	215	354	602	1021	196	628	1960	6280
	10^{-4}	678	945	1899	2672	620	1870	6200	18700
2	10^{-2}	56	408	157	1181	10	37	100	370
	10^{-3}	177	619	496	1766	32	102	320	1020
	10^{-4}	560	1084	1568	3063	100	305	1000	3050

Finally, we simulated both scenarios for different tolerance values ΔQ to compare the number of steps and updates performed by a quantization based method (LIQSS2) and a time discretization algorithm (TR12). Table 8 provides the simulation results obtained along with the theoretical lower bounds for the number of steps and updates.

These results corroborate our theoretical expectations: the LIQSS2 method performs better in the first scenario, while the TR12 has the best performance in the second scenario.

An interesting observation is that in the first scenario, the number of state updates observed in LIQSS2 is less than the lower bound for the discrete time algorithm $u_{\text{DT}}^{(2)}$. This implies that it is not possible to construct a second order accurate discrete time algorithm that performs fewer state updates than those produced by LIQSS2 in this case.

Another observation is that heterogeneous activity is linked with large values of the reaction parameter R (which causes each state to experience an abrupt change from 0 to 1). On the contrary, the homogeneous behavior is linked with large values of the diffusion term D. Thus, it is not actually necessary to compute the HF for every set of parameters, as the convenience of QSS or classic approaches can be extrapolated from the previous observation.

4.2 Spiking Neural Network Model

We consider a spiking neural network (SNN) where the behavior of each neuron is represented by a Leaky Integrate and Fire (LIF) model, based on the one described in [15,16]. The state variables associated to the i-th neuron are the membrane potential $V_i(t)$ and the synaptic current $I_i^s(t)$, whose continuous dynamics are described by the following differential equations:

$$\begin{aligned} \frac{\mathrm{d}I_i^s(t)}{\mathrm{d}t} &= -\frac{I_i^s(t)}{\tau_s} + \frac{I_e}{\tau_s} \\ \frac{\mathrm{d}V_i(t)}{\mathrm{d}t} &= -\frac{V_i(t) - E_L}{\tau_m} + \frac{I_i^s(t)}{C_m} \end{aligned} \tag{36}$$

for $i = 1, \ldots, N_e$, where N_e is the number of neurons in the network.

Here, τ_s, E_L, τ_m, and C_m are parameters whose values and definitions can be found in Table 9 and I_e is an additional parameter representing a constant input current.

Table 9. SNN Model Parameters.

Model parameters	
Value	Description
$\tau_m = 10$ ms	membrane time constant
$\tau_r = 2$ ms	absolute refractory period
$\tau_s = 0.5$ ms	postsynaptic current time constant
$C_m = 250$pF	membrane capacity
$V_r = -65$mV	reset potential
$\theta = -50$ mV	fixed firing threshold
$E_L = -65$ mV	leak potential

This model also exhibits discontinuities: whenever the membrane potential V_i reaches a value θ, the i–th neuron generates a *spike* and the membrane potential is reset to the value V_r. In addition, the spike is transmitted (synapses) to the neurons connected to the firing neuron. These neurons update their synaptic current according to

$$I_j^s(t^+) \leftarrow I_j^s(t) + J_i^{\text{exc}} \tag{37}$$

if the firing neuron is of excitatory type, while it is updated as

$$I_j^s(t^+) \leftarrow I_j^s(t) - J_i^{\text{inh}} \tag{38}$$

when the neuron that fires is of inhibitory type. The parameters J_i^{exc} and J_i^{inh} are the excitatory and inhibitory synaptic strengths, whose values are normally distributed as explained in [16].

Notice that Eq.(36) defines the continuous dynamics like Eq.(28), the conditions $V_i - \theta = 0$ correspond to the zero crossing functions of Eq.(29) and the Equations (37)–(38) correspond to the event handler of Eq.(30).

We consider a network formed by $N_e = 1,000$ interconnected neurons, of which 800 are of excitatory type, and the remaining 200 are of inhibitory type. Following the model of [16], each neuron receives spikes from $m = 10$ randomly selected synaptic connections, with 80% of these connections originating from excitatory neurons and the remaining 20% from inhibitory neurons.

In order to analyze how discontinuities impact the resulting number of state updates, we simulated the model provoking different rates of discontinuities by changing the parameter I_e. Notice that a larger value of this parameter increases the speed at which V_s reaches the threshold value, which in turn increases the rate at which discontinuities occur.

We performed simulations using the algorithms LIQSS2 and CVODE (with maximum order limited to 2) and different tolerance settings.

Figures 5a and 5b illustrate a subset of the state trajectories resulting from a simulation using a constant external current $I_e = 375.5$ nA (this current is close to the minimum value for which neurons can reach the threshold and fire).

From the simulation results using high-accuracy settings, we computed the activities of order 2 for the different scenarios according to the value of I_e. Subsequently, we calculated the theoretical lower bounds for the number of steps required by each method to complete the simulation, following Eqs.(31) and (32), which account for the presence of discontinuities. These results are presented in Table 10, alongside the actual number of simulation steps performed by each numerical method.

In addition, we calculated the Homogeneity Factor $H_{\mathbf{x}}^{(2)}$, the product $H_{\mathbf{x}}^{(2)} \cdot r^{(2)}$ and the lower bounds for the number of updates required for each approach following Eqs. (33) and (34).

To compute the lower bounds for $u_{\text{QS}}^{(2)}$, we calculated the value of $r^{(2)}$ for each scenario, as the connectivity is not constant (the state $V_i(t)$ appears on the right hand side of a single state equation while $I_i^s(t)$ appears in two state equations). We obtain the following values:

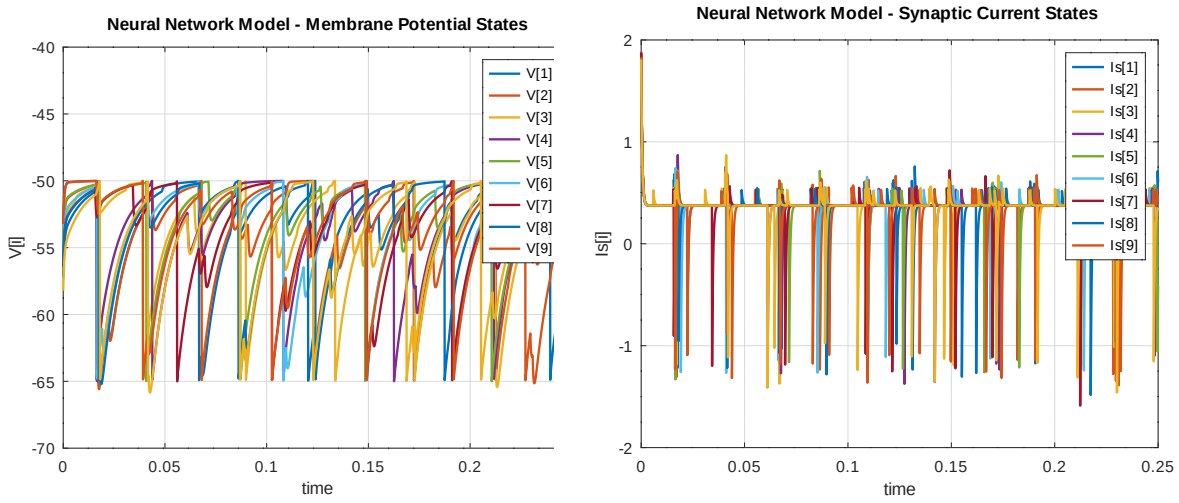
- Scenario 1 ($I_e/\tau_s = 751$): $r^{(2)} = 1.16$
- Scenario 2 ($I_e/\tau_s = 800$): $r^{(2)} = 1.22$
- Scenario 3 ($I_e/\tau_s = 1000$): $r^{(2)} = 1.19$

We also set $s = 11$ since each spike affects 11 state derivatives (corresponding to $m = 10$ neurons that receive the event and the one that fires).

Table 11 reports the Homogeneity Factor, the ratio between the lower bounds for the updates $u_{\text{QS}}^{(2)}/u_{\text{DT}}^{(2)}$ and the actual observed ratio between the updates in the simulation. In addition, the table reports the ratio between the simulation CPU times.

From the results of Tables 10 and 11, we can draw the following observations:

- The model has $N = 2,000$ state variables (two states per neuron); thus, the theoretical range for $H_{\mathbf{x}}^{(n)}$ is $[\frac{1}{2000}, 1]$. In all three scenarios, the Homogeneity Factor is significantly below 1, which is consistent with the heterogeneous activity of the trajectories.
- CVODE2 performs fewer steps than LIQSS2. However, every step in CVODE2 involves computations in the entire state vector (of size $N = 2,000$), while LIQSS2 steps only involve computations in at most $r = 2$ states (continuous steps) or $s = 11$ states (when it corresponds to a discontinuity).
- In agreement with the previous observation, the number of updates performed by LIQSS2 results between 10 and 70 times lower than that of CVODE2.
- The number of steps performed by both algorithms is close to their corresponding theoretical lower bounds. Consequently, the ratio between the state



(a) Membrane potentials

(b) Synaptic currents

Figure 5. A subset of the state trajectories for the SNN model.

Table 10. Number of discontinuities (spikes) and number of steps (lower bounds and observed) in different scenarios.

I_e/τ_s	Spikes	Abs Tol.	LIQSS2		CVODE$_2$	
			$k_{\text{QS}}^{(2)}$	Real	$k_{\text{DT}}^{(2)}$	Real
751	4,741	10^{-1}	$1.15 \cdot 10^5$	$1.60 \cdot 10^5$	$5.47 \cdot 10^3$	$9.57 \cdot 10^3$
		10^{-2}	$3.55 \cdot 10^5$	$4.51 \cdot 10^5$	$7.04 \cdot 10^3$	$1.08 \cdot 10^4$
		10^{-3}	$1.11 \cdot 10^6$	$1.41 \cdot 10^6$	$1.20 \cdot 10^4$	$1.28 \cdot 10^4$
800	7,701	10^{-1}	$2.10 \cdot 10^5$	$2.09 \cdot 10^5$	$8.98 \cdot 10^3$	$1.52 \cdot 10^4$
		10^{-2}	$6.49 \cdot 10^5$	$7.63 \cdot 10^5$	$1.17 \cdot 10^4$	$1.69 \cdot 10^4$
		10^{-3}	$2.03 \cdot 10^6$	$2.32 \cdot 10^6$	$2.05 \cdot 10^4$	$2.08 \cdot 10^4$
1000	14,531	10^{-1}	$3.05 \cdot 10^5$	$3.05 \cdot 10^5$	$1.60 \cdot 10^4$	$2.85 \cdot 10^4$
		10^{-2}	$9.33 \cdot 10^5$	$1.15 \cdot 10^6$	$1,92 \cdot 10^4$	$3.00 \cdot 10^4$
		10^{-3}	$2.92 \cdot 10^6$	$3.49 \cdot 10^6$	$2.92 \cdot 10^4$	$3.55 \cdot 10^4$

Table 11. Theoretical vs. simulation relation between computational costs of a state quantization algorithm and a discrete time algorithm

I_e/τ_s	Abs. Tol. ΔQ	Theoretical Results			Simulation Results	
		$H_{\mathbf{x}}^{(2)}$	$H_{\mathbf{x}}^{(n)} \cdot r^{(2)}$	$\frac{u_{\text{QS}}^{(2)}}{u_{\text{DT}}^{(2)}}$	$\frac{\text{upd LIQSS2}}{\text{upd CVODE2}}$	$\frac{\text{CPU LIQSS2}}{\text{CPU CVODE2}}$
751	10^{-1}	0.076	0.088	0.017	0.019	0.0015
	10^{-2}			0.033	0.037	0.0029
	10^{-3}			0.056	0.079	0.0069
800	10^{-1}	0.079	0.096	0.018	0.017	0.0013
	10^{-2}			0.037	0.038	0.0029
	10^{-3}			0.062	0.084	0.0069
1000	10^{-1}	0.099	0.118	0.016	0.015	0.0012
	10^{-2}			0.033	0.034	0.0026
	10^{-3}			0.062	0.078	0.0059

update lower bounds $u_{\text{QS}}^{(2)}/u_{\text{DT}}^{(2)}$ is close to what was observed in the simulation.

- The fact that $H_{\mathbf{x}}^{(n)} \cdot r^{(2)} < 1$ implies that, even without discontinuities, QSS methods are expected to be more efficient in this model. The presence

of discontinuities makes QSS even more efficient, as can be deduced from the fact that $u_{\mathrm{QS}}^{(2)}/u_{\mathrm{DT}}^{(2)} \ll H_{\mathbf{x}}^{(n)} \cdot r^{(2)}$.

- The relationship between CPU times shows even greater advantages for QSS algorithms. While the number of updates is between 10 to 70 times smaller in QSS, the simulations are between 140 and 800 times faster. This is mainly related to the fact that CVODE is an implicit algorithm and needs to perform iterations within each single time step. Although the iterations are on a linear system and use a sparse Jacobian representation, it implies a larger computational cost compared to LIQSS2 (which is also implicit, but only needs to solve scalar linear implicit equations).

5 Conclusions and Future Work

We extended the concept of activity for continuous time signal by defining the notions of instantaneous activity, local activity and global activity. While the original definition of activity was tied to the minimum number of steps required by a QSS method to simulate a system, these new metrics can also be related to the minimum number of steps needed by classic numerical integration algorithms to complete a simulation. In that sense, we also developed expressions linking local and global activity to the lower bounds on the number of steps performed by both QSS and classic algorithms.

Based on these developments, we formally introduced the concept of activity homogeneity for the system trajectories and characterized this concept by defining a Homogeneity Factor. This measure provides a numerical value close to 1 when trajectories are homogeneous and a value close to $1/N$ when they are heterogeneous (where N is the dimension of the state vector).

Using this metric, we provided a formal proof for the claim that QSS methods are expected to perform better than classic methods when the activity is heterogeneous and the systems are sparse.

Furthermore, we extended these results to systems exhibiting discontinuities, providing lower bounds for the number of steps and state updates in both quantization-based and classic discrete time simulations. This analysis formally supports the advantage of using QSS algorithms in systems with frequent discontinuities.

Finally, we demonstrated the application of these proposed metrics in two examples under different parameter settings, validating the theoretical insights through simulation results.

We are currently working on extending the definitions to incorporate the use of relative tolerance, that in QSS methods is implemented using a quantum size ΔQ_i that changes with the time so that it is proportional to the signal $x_i(t)$[17]. In addition, we are formulating new QSS algorithms designed to approximate as closely as possible the lower bound in the number of steps established by Eq.(19).

Another future line of work is to develop an automatic solver selection algorithm that computes the homogeneity factor as a simulation progresses and uses its value to decide which numerical integration approach is expected to perform better.

6 Statements and Declarations

Funding

This work was partially funded by grant PICT–2021 00826 (ANPCYT).

References

1. Hairer E, Nørsett SP and Wanner G. *Solving ordinary differential equations. I: Nonstiff problems*. Springer, 1993.
2. Hairer E and Wanner G. *Solving ordinary differential equations. II: Stiff and differential-algebraic problems*. Springer-Verlag, 1996.
3. Cellier F and Kofman E. *Continuous System Simulation*. New York: Springer, 2006.
4. Kofman E and Junco S. Quantized State Systems. A DEVS Approach for Continuous System Simulation. *Transactions of SCS* 2001; 18(3): 123–132.
5. Migoni G, Bortolotto M, Kofman E et al. Linearly implicit quantization-based integration methods for stiff ordinary

differential equations. *Simulation Modelling Practice and Theory* 2013; 35: 118–136.

6. Jammalamadaka R. Activity characterization of spatial models: Application to the discrete event solution of partial differential equations. *Master's thesis, University of Arizona, Tucson, Arizona, USA* 2003; .
7. Muzy A, Jammalamadaka R, Zeigler BP et al. The activity-tracking paradigm in discrete-event modeling and simulation: The case of spatially continuous distributed systems. *Simulation* 2011; 87(5): 449–464.
8. Castro R and Kofman E. Activity of order n in continuous systems. *Simulation* 2015; 91(4): 337–348.
9. Castro R, Bergonzi M, Marcosig EP et al. Discrete-event simulation of continuous-time systems: evolution and state of the art of quantized state system methods. *Simulation* 2024; 100(6): 613–638.
10. Kofman E. A second-order approximation for devs simulation of continuous systems. *Simulation* 2002; 78(2): 76–89.
11. Kofman E. A third order discrete event method for continuous system simulation. *Latin American applied research* 2006; 36(2): 101–108.
12. Petzold L. Automatic selection of methods for solving stiff and nonstiff systems of ordinary differential equations. *SIAM journal on scientific and statistical computing* 1983; 4(1): 136–148.
13. Fernández J and Kofman E. A stand-alone quantized state system solver for continuous system simulation. *Simulation* 2014; 90(7): 782–799.
14. Bergero F, Fernández J, Kofman E et al. Time discretization versus state quantization in the simulation of a one-dimensional advection–diffusion–reaction equation. *Simulation* 2016; 92(1): 47–61.
15. Schmidt M, Bakker R, Shen K et al. A multi-scale layer-resolved spiking network model of resting-state dynamics in macaque visual cortical areas. *PLOS Computational Biology* 2018; 14(10): e1006359.
16. Bergonzi M, Fernández J, Castro R et al. Quantization-based simulation of spiking neurons: theoretical properties and performance analysis. *Journal of Simulation* 2024; 18(5): 789–812.
17. Kofman E. Relative error control in quantization based integration. *Latin American applied research* 2009; 39(3): 231–237.
18. Süli E and Mayers DF. *An introduction to numerical analysis*. Cambridge university press, 2003.

A Proof of Theorem 1

The proof of Theorem 1 requires some auxiliary results that are given next.

A.1 Auxiliary Results

Denoting T_n to the n-th degree Chebyshev polynomial, the following property is a corollary of Theorem 8.6 of [18].

Corollary 8.1 of [18]. *Suppose that $n \geq 0$. Among all monic[¶] polynomials of degree $n+1$ the polynomials $2^{-n}T_{n+1}$ and $-2^{-n}T_{n+1}$ have the smallest ∞-norm on the interval $[-1, 1]$.*

Then, using that result, the following lemma will be used in the proof of the main theorem.

Lemma 1. *Let $\mathcal{P}_n$ be the family of monic polynomials of the form $p(t) = t^n + a_{n-1}t^{n-1} + \ldots + a_0$ with $a_i \in \mathbb{R}$. Define the exit time of $p(t) \in \mathcal{P}_n$ as*

$$\mathrm{t}_{\mathrm{out}}(p) \triangleq \inf\{t : t \geq 0 \wedge |p(t)| > 1\} \tag{39}$$

and define

$$\mathrm{t}_{\max}(n) \triangleq 2^{\frac{2n-1}{n}} \tag{40}$$

Then

(i) There exists the following maximum

$$t_m(n) \triangleq \max_{p \in \mathcal{P}_n} \mathrm{t}_{\mathrm{out}}(p) \tag{41}$$

which is called the maximum time of permanence.

(ii) For all $n \geq 1$, it results

$$t_m(n) = \mathrm{t}_{\max}(n) \tag{42}$$

and it corresponds to the polynomial

$$p_n^*(t) = \arg\max_{p \in \mathcal{P}_n}(\mathrm{t}_{\mathrm{out}}(p)) = T_n\left(\frac{2t - \mathrm{t}_{\max}(n)}{\mathrm{t}_{\max}(n)}\right) \tag{43}$$

[¶] A monic polynomial is a non-zero polynomial where the coefficient of its highest degree term (the leading coefficient) is exactly 1.

Proof. Notice first that $p_n^*(t) \in \mathcal{P}_n$ since the leading coefficient of $T_n(\tau)$ is 2^{n-1} and then, the leading coefficient of $p_n^*(t)$ can be computed as

$$\lim_{t\to\infty} \frac{p_n^*(t)}{t^n} = \lim_{t\to\infty} \frac{T_n(\frac{2t-\mathrm{t}_{\max}(n)}{\mathrm{t}_{\max}(n)})}{t^n}$$
$$= \frac{2^n}{\mathrm{t}_{\max}(n)^n} \lim_{t\to\infty} \frac{T_n(t)}{t^n} = \frac{2^n}{2^{2n-1}} 2^{n-1} = 1$$

Notice also that $\mathrm{t}_{\mathrm{out}}(p_n^*(t)) = \mathrm{t}_{\max}(n)$ since

$$t \in [0, \mathrm{t}_{\max}(n)] \iff \frac{2t - \mathrm{t}_{\max}(n)}{\mathrm{t}_{\max}(n)} \in [-1, 1]$$

and also

$$T_n(\tau) \in [-1, 1] \iff \tau \in [-1, 1]$$

thus, taking $\tau = \frac{2t-\mathrm{t}_{\max}(n)}{\mathrm{t}_{\max}(n)}$ and combining the last two relations, it results

$$p_n^*(t) \in [-1, 1] \iff t \in [0, \mathrm{t}_{\max}(n)].$$

In order to complete the proof, suppose for a contradiction that there exists a polynomial $\hat{p}_n(t) \in \mathcal{P}_n$ with $\mathrm{t}_{\mathrm{out}}(\hat{p}_n) > \mathrm{t}_{\max}(n)$. Then, define the following polynomial

$$\tilde{p}_n(\tau) = \hat{p}_n\left(\frac{\mathrm{t}_{\mathrm{out}}(\hat{p}_n)(\tau+1)}{2}\right)\left(\frac{2}{\mathrm{t}_{\mathrm{out}}(\hat{p}_n)}\right)^n$$

Notice that when $\tau \in [-1, 1]$ the polynomial $\hat{p}_n$ is evaluated in $[0, \mathrm{t}_{\mathrm{out}}(\hat{p}_n)]$ verifying $|\hat{p}_n| \leq 1$ and then, the polynomial $\tilde{p}_n(\tau)$ verifies

$$|\tilde{p}_n(\tau)| \leq \left(\frac{2}{\mathrm{t}_{\mathrm{out}}(\hat{p}_n)}\right)^n < \left(\frac{2}{\mathrm{t}_{\max}(n)}\right)^n$$

for $\tau \in [-1, 1]$. Defining the ∞ norm in the interval $[-1, 1]$, and replacing the value of $\mathrm{t}_{\max}(n)$, we can rewrite

$$\|\tilde{p}_n(\tau)\|_\infty < \left(\frac{2}{2^{\frac{2n-1}{n}}}\right)^n = 2^{-(n-1)} \tag{44}$$

Notice also that the infinite norm of the monic polynomial with minimal infinite norm, according to Corollary 8.1 of [18], is

$$\|2^{-(n-1)} T_n(\tau)\|_\infty = 2^{-(n-1)} \tag{45}$$

Then, observing that $\tilde{p}_n(\tau)$ is a monic polynomial (its leading term is 1), the expressions of Eqs.(44) and (45) contradict the result of the mentioned corollary.

That way the proof is complete since $\mathrm{t}_{\mathrm{out}}(p_n^*(t)) = \mathrm{t}_{\max}(n)$ and there is no polynomial in $\mathcal{P}_n$ with a larger value of $\mathrm{t}_{\mathrm{out}}$.

A.2 Proof of Theorem 1

Taking into account the results obtained in Lemma 1, we present the proof of the Theorem 1:

Proof. Let a_n be the leading coefficient of $x_i(t)$. Then, we define the polynomial:

$$p(t) \triangleq \mathrm{sign}(a_n)(x_i(t) - q_i(t)) \tag{46}$$

Note that $p(t)$ is a polynomial of order n, whose leading coefficient is $|a_n|$. From $p(t)$, we then define the following polynomial:

$$\hat{p}(t) \triangleq \frac{1}{\Delta Q_i} \cdot p\left((\frac{\Delta Q_i}{|a_n|})^{\frac{1}{n}} \cdot t + t_j\right) \tag{47}$$

Observe that:

- The leading coefficient of $\hat{p}(t)$ is 1. Consequently, $\hat{p}(t) \in \mathcal{P}_n$ (where the set $\mathcal{P}_n$ is the one defined in Lemma 1).
- By appropriately choosing $q_i(t)$ in Eq.(46), any polynomial $\hat{p}(t)$ from the set $\mathcal{P}_n$ can be constructed.
- Therefore, according to Lemma 1, there exists a polynomial $q_i(t)$ for which the time of permanence satisfying the condition $|\hat{p}(t)| \leq 1$ is $t_{max}(n)$, defined in Eq.(41). Moreover, there is no polynomial $q_i(t)$ for which the permanence time is longer.

Note also that $|p(t)| \leq \Delta Q_i$ in $[t_j, t_j + \Delta t] \Leftrightarrow |\hat{p}(t)| \leq 1$ in $[0, (\frac{|a_n|}{\Delta Q_i})^{\frac{1}{n}} \Delta t]$.

Therefore, there will exist a polynomial $q_i(t)$ that verifies $|p(t)| \leq \Delta Q_i$ in $[t_j, t_j + \Delta t]$ if and only if

$$(\frac{|a_n|}{\Delta Q_i})^{\frac{1}{n}} \Delta t \leq \mathrm{t}_{\max} = 2^{\frac{2n-1}{n}} \tag{48}$$

Or, equivalently:

$$\Delta t \leq 2^{\frac{2n-1}{n}} (\frac{\Delta Q_i}{|a_n|})^{\frac{1}{n}} \tag{49}$$

On the other hand, the local instantaneous activity of $x_i(t)$ in t_j is

$$a_{x_i}^{(n)}(t_j) = \left| \frac{\frac{\mathrm{d}^n x_i(\tau)}{\mathrm{d}\tau^n}}{n!} \right|^{1/n} = (\frac{|a_n| \cdot n!}{n!})^{\frac{1}{n}} = |a_n|^{\frac{1}{n}} \tag{50}$$

and replacing this expression in the Eq. (49) we obtain

$$\Delta t \leq 2^{\frac{2n-1}{n}} \cdot \frac{\Delta Q_i^{\frac{1}{n}}}{a_{x_i}^{(n)}(t_j)} \tag{51}$$

completing the proof.

Author Biographies