Quantization-Based Simulation of Spiking Neurons: Theoretical Properties and Performance Analysis

Mariana Bergonzi[ab], Joaquín Fernández[a], Rodrigo Castro[cd], Alexandre Muzy[e], Ernesto Kofman[ab]

[a]CIFASIS-CONICET, Rosario, Argentina; [b]FCEIA-UNR, Rosario, Argentina; [c]ICC-CONICET,Buenos Aires, Argentina; [d]FCEyN-UBA, Buenos Aires, Argentina; [e]Université Côte d'Azur, CNRS, I3S, Nice, France

ARTICLE HISTORY

Compiled November 21, 2023

ABSTRACT

In this work we present an exhaustive analysis of the use of Quantized State Systems (QSS) algorithms for the discrete event simulation of Leaky Integrate and Fire models of spiking neurons. Making use of some properties of these algorithms, we first derive theoretical error bounds for the sub-threshold dynamics as well as estimates of the computational costs as a function of the accuracy settings. Then, we corroborate those results on different simulation experiments, where we also study how these algorithms scale with the size of the network and its connectivity.

The results obtained show that the QSS algorithms, without any type of optimization or specialization, obtain accurate results with low computational costs even in large networks with a high level of connectivity.

KEYWORDS

Hybrid Systems; Discontinuity Handling; Quantized State Systems; Spiking Neural Networks; Event-Driven Simulation

1. Introduction

The presence of discontinuities in systems of Ordinary Differential Equations (ODE) has been always a challenge for the numerical integration algorithms used for their simulation. Steps that integrate through discontinuities usually introduce unacceptable large errors and for this reason they must be avoided. For this purpose, it is necessary to detect the precise instant of time at which each discontinuity occurs in order to advance the simulation up to that time. Then, the integration can be restarted from the situation after the event occurrence (Cellier & Kofman, 2006). This process, known as *zero crossing detection and handling*, effectively avoids integrating through discontinuities at the price of large computational costs: the detection of the discontinuity times usually requires iterations (Park & Barton, 1996) and then restarting the simulation implies using a small step size.

Models of spiking Integrate and Fire neurons are a particular case of discontinuous ODEs. There, when some variable (usually the membrane potential) reaches a threshold, the neuron fires producing a spike that not only resets some internal variables but

CONTACT Mariana Bergonzi. Email: bergonzi@cifasis-conicet.gov.ar

that is also propagated through synaptic connections to other neurons in the network. When the size of a spiking neural network (SNN) grows, the time between successive spikes on the whole network becomes smaller, and consequently the maximum step size that can be used by the numerical algorithms is reduced. In addition, the computational cost of each step grows with the number of neurons.

Thus, simulating large SNN with standard zero crossing detection and handling algorithms becomes impractical and different solutions have been proposed to overcome that problem (Brette et al., 2007; Hansel, Mato, Meunier, & Neltner, 1998; Morrison & Diesmann, 2007; Shelley & Tao, 2001). Most of these solutions involve using a fixed small step size and whenever one or more spikes are detected within an integration step, instead of reducing the step size to the time instant of the first spike, some correction procedure is executed.

In the last two decades, a family of numerical ODE algorithms that replaces the classic time discretization by the quantization of the state variables was developed. These algorithms, called Quantized State Systems (QSS) methods (Cellier & Kofman, 2006; Kofman, 2002; Kofman & Junco, 2001; Migoni, Bortolotto, Kofman, & Cellier, 2013), approximate the ODEs in an event driven fashion where each step only involves local computations around a state variable that experiences a significant change. QSS methods, besides having strong theoretical properties regarding stability and error bounds, are particularly efficient to integrate discontinuous models (Kofman, 2004a) since zero crossing detection is straightforward and the occurrence of a discontinuity only provokes local computations similar to those of a normal step.

In this work, we first analyze different theoretical properties of QSS methods applied to the simulation of Leaky Integrate and Fire (LIF) spiking neurons that establish upper bounds for different errors (synaptic current, membrane potential, and firing times) and also estimate the expected computational costs. We also perform an exhaustive comparison of computational costs and errors for the different QSS algorithms in the simulation of a single neuron model. This analysis is then scaled up to a neural network level, studying the way the computational costs increase with the growth of the network and its connectivity.

We shall see that QSS methods offer a promising alternative for the simulation of SNNs, providing accurate results with low computational costs. Moreover, this is achieved without introducing any ad-hoc modification or specialization to the algorithms for the particular problem, as it is usually done with classic methods.

The article is organized as follows. Section 2 provides the background in which the rest of the work is based, including the description of LIF models and QSS algorithms. Then, Section 3 contains the main theoretical results and Section 4 reports and analyzes the simulation results. Finally, Section 5 presents the conclusions and the future work.

2. Background

In this section we introduce different ODE models of spiking neurons and the numerical methods used to simulate them. We then present the QSS family of methods and we continue by recalling the core concept of *activity* that allows analyzing the computational costs of QSS simulations. Finally, we introduce the software tool we have used to run the different experiments throughout this work and we discuss previous results where these methods were used to simulate spiking neurons.

2.1. Modeling and Simulation of Spiking Neurons

Spiking neurons are usually described by systems of ODEs with discontinuities associated to the firing times. Among the different models that have been proposed, we can mention the following ones:

- Hodgkin–Huxley (HH) model (Hodgkin & Huxley, 1952): this detailed model, developed in the 1950s, describes the electrical action potentials of neurons accounting for ion channel potentials. The main drawback of this model is that its simulation is computationally expensive, since each neuron is represented by four differential equations governed by 10 parameters. Thus, its usage is usually limited to networks formed by few neurons (Izhikevich, 2003b).
- Leaky Integrate and Fire (LIF) model (Lapicque, 1907; Stein, 1967): this simplified model, introduced by Lapique in 1907, is also based on an electrical representation. The circuit consists of a resistor and a capacitor in parallel, representing the leakage and capacitance of the membrane (Burkitt, 2006), respectively. The membrane capacitor is charged until it reaches a certain threshold, after which it discharges producing an action potential (spike). This behaviour is modeled by the differential equation

$$\dot{v}(t) = I(t) + a - bv(t), \qquad \text{if } v(t) \geq v_u \text{ then } v \leftarrow c,$$

  where $v(t)$ is the membrane potential, $I(t)$ is the input current and a, b, c, and v_u (the threshold) are the model parameters.

Several modifications have been proposed to improve the LIF model, such as the inclusion of a quadratic term on $v(t)$ (quadratic integrate and fire (Ermentrout, 1996)), or the addition of a second state variable in order to represent more complex behaviors (integrate and fire or burst (Izhikevich, 2003b)). Other variants are the exponential integrate-and-fire neuron (Fourcaud-Trocmé & Brunel, 2005; Fourcaud-Trocmé, Hansel, Van Vreeswijk, & Brunel, 2003), the generalized integrate-and-fire neuron (Brunel, Hakim, & Richardson, 2003; Richardson, Brunel, & Hakim, 2003) and the noisy LIF (Gerstner, Kistler, Naud, & Paninski, 2014). The so called Izhikevich model,described in (Izhikevich, 2003a) and (Izhikevich, 2007), that consists of a second order nonlinear system (with states representing the membrane potential and the membrane recovery), can also be regarded as an extension of the LIF model.

The above presented models can be simulated with classic numerical methods (Brette et al., 2007), although some problems often appear since every time a neuron emits or receives a spike a discontinuity occurs. This constitutes a difficulty for conventional (time discretization-based) numerical methods because they cannot integrate across discontinuities since the polynomial approximations on which these numerical schemes are based are not valid on discontinuous functions. To overcome this problem, the methods must detect each discontinuity, then advance the simulation time to that instant, and finally restart the simulation from the new situation (Cellier & Kofman, 2006).

This whole process, which usually involves iterations to find the precise instant of the discontinuity occurrence, is computationally expensive. Moreover, in a model like that of a spiking neural network, the rate of occurrence of discontinuities (*i.e.*, spikes) grows with the number of neurons and then the maximum step size is limited to a smaller value as the size of the system grows.

To overcome these problems, different strategies have been implemented. A simple solution is to use a small fixed step size that introduces a small error in the spiking times. A more precise and sophisticated approach is that of (Hanuschkin, Kunkel, Helias, Morrison, & Diesmann, 2010a) in which synapses communicate between neurons at discrete time instants (usually larger than the integration step size) making use of a correction mechanism to take into account the late arrival of incoming spikes.

A different approach is followed in (Zheng, Tonnelier, & Martinez, 2009), where the authors propose a numerical integration method for generic simulations of nonlinear spiking neuron models called *voltage-stepping*. This approach is based on the discretization of the voltage values in order to simulate with a local event-driven method and to take advantage of its properties. Then, in (Kaabi, Tonnelier, & Martinez, 2011) this approach is scaled up to a neural network level using the Discrete Event System Specification (DEVS) (Zeigler, Kim, & Praehofer, 2000).

Another approach also related to event-driven techniques is the one presented in (Mascart, Scarella, Reynaud-Bouret, & Muzy, 2022), where the authors modeled only the time of the spikes by a point processes and they developed a specific algorithm called Activity Tracking with Time Asynchrony (ATiTA).

2.2. Quantized State System Methods

QSS methods are a family of numerical integration algorithms that replace the time discretization of classic numerical integration algorithms by the quantization of the state variables.

Given a time invariant ODE in its State Equation System (SES) representation:

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

where $\mathbf{x}(t) \in \mathbb{R}^n$ is the state vector, the first order Quantized State System (QSS1) method (Kofman & Junco, 2001) solves an approximate ODE called Quantized State System:

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

Here, $\mathbf{q}(t)$ is the *quantized state* vector. Each component of the quantized state $q_i(t)$ follows a piecewise constant trajectory that only changes when its difference with the corresponding state $x_i(t)$ reaches the quantum ΔQ_i.

The quantized state trajectory is related to the corresponding state trajectory $x_i(t)$ as follows:

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

for $t_k < t \leq t_{k+1}$, where t_{k+1} is the first time after t_k at which $|x_i(t) - q_i(t_k)| = \Delta Q_i$. In addition, we consider that initially $\mathbf{q}(t_0) = \mathbf{x}(t_0)$. This defines a *hysteretic quantization function* generating trajectories like those depicted in Figure 1.

Since the quantized state trajectories $q_i(t)$ are piecewise constant, then, provided that the system is autonomous (or that $\mathbf{f}(\cdot, t)$ is piecewise constant with t), the state derivatives $\dot{x}_i(t)$ also follow piecewise constant trajectories and, consequently, the states $x_i(t)$ follow piecewise linear trajectories. In non autonomous systems Eq. (2)

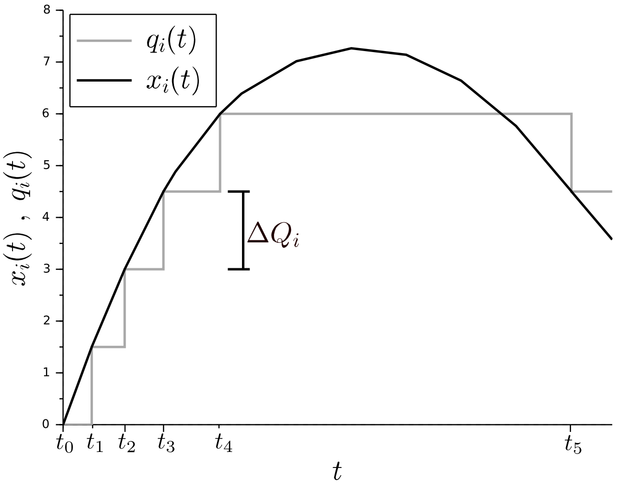

Figure 1. Typical QSS1 state and quantized state trajectories.

can be rewritten as

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{q}(t), t) = \tilde{\mathbf{f}}(\mathbf{q}(t), \mathbf{u}(t))$$

for some *input trajectories* $\mathbf{u}(t)$ that are approximated by piecewise constant trajectories $\mathbf{v}(t)$ such that the difference $v_i(t) - u_i(t)$ remains bounded by a certain quantity (given by the input quantization). That way, the QSS1 approximation actually integrates the system

$$\dot{\mathbf{x}}(t) = \tilde{\mathbf{f}}(\mathbf{q}(t), \mathbf{v}(t))$$

Due to the particular form of the trajectories, the numerical solution of Eq. (2) is straightforward and can be easily translated into a simulation algorithm.

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 (Kofman, 2002).

QSS2 has 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 2. Then, the state derivatives $\dot{\mathbf{x}}(t)$ are computed as piecewise linear trajectories so that the states $\mathbf{x}(t)$ follow piecewise parabolic trajectories.

The idea was also extended in a similar way to obtain a third order accurate QSS method called QSS3, in which the quantized states $q_i(t)$ follow piecewise parabolic trajectories while the states follow piecewise cubic trajectories.

An important advantage of the QSS methods is that they handle discontinuities in a straightforward and very efficient manner (Kofman, 2004b). According to the order of the method, the quantized state variables follow piecewise constant, linear or parabolic trajectories. Then, detecting zero crossings is straightforward, as it involves solving a quadratic equation in the worst case (QSS3). Once a discontinuity is detected, the algorithm handles it as an ordinary step, in which only the state derivatives that are affected by that discontinuity are recomputed.

The efficiency in the discontinuity handling and the fact that the computations

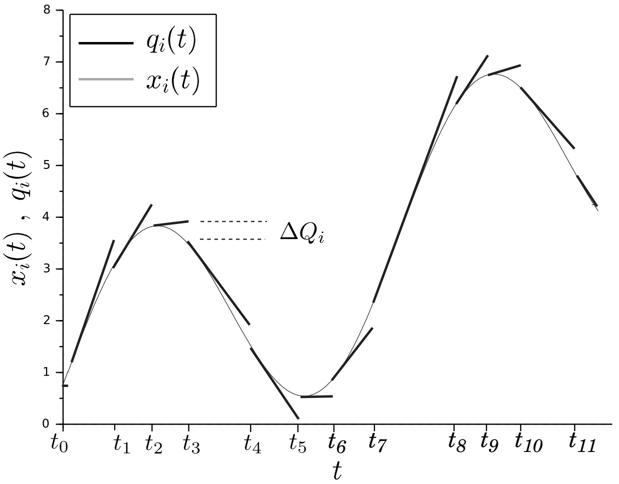

Figure 2. Typical QSS2 state and quantized state trajectories.

at each step are confined to the states that experience changes imply that these algorithms are very efficient to simulate large systems with heterogeneous activity involving discontinuities. The family of QSS methods also includes a set of algorithms called Linearly Implicit QSS (LIQSS), which are appropriate to simulate some stiff systems (Migoni et al., 2013). 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.

2.3. Theoretical Properties of QSS Methods

The fact that the difference between the states x_i and the corresponding quantized states q_i is bounded by the quantum ΔQ_i allows to rewrite Eq. (2) as

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t) + \mathbf{\Delta x}(t), t) \tag{3}$$

where $\mathbf{\Delta x}(t) \triangleq \mathbf{q}(t) - \mathbf{x}(t)$ is a perturbation term bounded by the quantum.

In consequence, the use of a QSS algorithm is equivalent to the addition of a bounded perturbation to the original system and several properties regarding convergence, stability, and global error bounds can be easily derived (Cellier & Kofman, 2006; Kofman & Junco, 2001) for linear and non-linear systems. One of those properties establishes that the use of QSS in stable linear time invariant systems of the form $\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + \mathbf{Bu}(t)$ produces a global error that can be bounded by the formula [1]

$$|\mathbf{x}(t) - \mathbf{x_a}(t)| \leq |\mathbf{V}| \cdot |\mathbb{R}e\{\mathbf{\Lambda}\}^{-1} \cdot \mathbf{\Lambda}| \cdot |\mathbf{V}^{-1}| \cdot \mathbf{\Delta Q} \tag{4}$$

where $\mathbf{x}$ and $\mathbf{x_a}$ are the QSS and the analytical solutions, and $\mathbf{\Lambda} = \mathbf{V}^{-1}\mathbf{AV}$ is the Jordan decomposition of matrix $\mathbf{A}$. That way, there is a linear dependence between

[1] The expression $|\mathbf{M}|$ denotes the component-wise absolute value of a vector or matrix $\mathbf{M}$. Also, $\mathbf{a} \preceq \mathbf{b}$ expresses a set of component-wise inequalities $a_i \leq b_i$ on all the components of $\mathbf{a} \in \mathbb{R}^n$ and $\mathbf{b} \in \mathbb{R}^n$.

the quantum and the global error bound.

For these reasons, the quantum plays an equivalent role to that of the tolerance in variable step size algorithms.

2.4. Activity and QSS Methods

The concept of activity associated to continuous signals was introduced in (Jammalamadaka, 2003; Muzy, Jammalamadaka, Zeigler, & Nutaro, 2011) in order to measure the rate of change of the signal. The formal definition of the activity metrics for a continuous signal $x_i(t)$ between an initial time t_0 and a final time t_f is given by:

$$A_{x_i(t_0,t_f)} \triangleq \int_{t_0}^{t_f} \left| \frac{dx_i(\tau)}{d\tau} \right| \cdot d\tau \tag{5}$$

This definition measures the distance between successive maxima and minima of a signal and results proportional to the number of segments used by a piecewise constant approximation like that of Figure 1.

With the goal of considering higher order approximations like that of Figure 2, this notion was extended in (Castro & Kofman, 2015), defining the concept of n-th order activity that takes into account not only the values but also the derivatives of the trajectory. Formally, given a signal $x_i(t)$, its n-th order activity in the interval $[t_0, t_f]$ is defined as

$$A^{(n)}_{x_i(t_0,t_f)} \triangleq \int_{t_0}^{t_f} \left| \frac{\frac{d^n x_i(\tau)}{d\tau^n}}{n!} \right|^{1/n} \cdot d\tau \tag{6}$$

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

$$k^{(n)}_{x_i(t_0,t_f)}(\Delta Q_i) \approx \left(\frac{1}{\Delta Q_i} \right)^{1/n} \int_{t_0}^{t_f} \left| \frac{\frac{d^n x_i(\tau)}{d\tau^n}}{n!} \right|^{1/n} \cdot d\tau = \frac{A^{(n)}_{x_i(t_0,t_f)}}{(\Delta Q_i)^{1/n}} \tag{7}$$

In QSS methods, using n equal to the order of the method (that uses polynomials of order up to $n-1$ in the quantized states) this formula allows estimating the number of steps performed in the i-th state variable when a quantum size ΔQ_i is selected.

A simple consequence of Eq. (7) is that the number of steps grows linearly with the accuracy (i.e., with the inverse of the quantum size) in the first order accurate QSS1 and LIQSS1 methods. It also grows with the square root of the accuracy in QSS2 and LIQSS2 and with the cubic root of the accuracy in the third order schemes.

2.5. The Stand Alone QSS Solver

While some DEVS-based simulation tools have implementations of different QSS algorithms (D'Abreu & Wainer, 2005; Quesnel, Duboz, Ramat, & Traoré, 2007; Zeigler, Muzy, & Kofman, 2018), the most efficient and complete tool for the family of QSS solvers is the Stand-Alone QSS solver (Fernández & Kofman, 2014).

The models in this solver are described using a subset of the Modelica language (Mattsson, Elmqvist, & Otter, 1998) called μ-Modelica. The tool automatically translates the models into a C language piece of code, containing the set of ODEs with the corresponding zero crossing functions and event handlers for discontinuous cases. The tool also extracts structure information (incidence matrices) and produces the code for the symbolic evaluation of the Jacobian matrix. The C code produced is then linked to the different QSS algorithms (QSS and LIQSS of order one to three) or to classic ODE solvers like DOPRI (Dormand & Prince, 1980), DASSL (Petzold, 1982), CVODE (Cohen, Hindmarsh, & Dubois, 1996) and IDA (Hindmarsh et al., 2005).

The fact that the tool provides all the information about the structure of the model and the code for the symbolic evaluation of the sparse Jacobian matrices implies that the results obtained by this tool using classic ODE solvers are noticeably faster than the results obtained by other interfaces (Kofman, Fernández, & Marzorati, 2021). In addition, the fact that the solver uses the same piece of code for computing the ODE right hand side functions in the different algorithms (QSS and classic ODE integrators), allows fair performance comparisons among them.

2.6. QSS and SNN Simulation

There exist some previous works applying state discretization to simulate spiking neural network models. Among them, the already mentioned voltage-stepping methods of (Kaabi et al., 2011; Zheng et al., 2009) are not formally QSS methods but use similar ideas also leading to event-driven simulations.

In the voltage-stepping approach, the variable representing the membrane potential is limited to take values on a discrete set. Then, steps are only performed when there is a discrete change in the voltage value. For that goal, the analytical solution of a differential equation is obtained at each step and, from that solution, the time of the next crossing for a new voltage value is computed. The differential equation that is solved is obtained from a piecewise linear approximation of the expression that computes the voltage derivative.

The authors showed that the accuracy of the approximation is of the order of Δv^2 where Δv is the voltage step, i.e., the difference between successive voltage values. Thus, in principle, the method would have the same accuracy order as both QSS2 and LIQSS2. However, the method lacks error bound formulas such as that of Eq.(4).

A problem of this approach is that it does not constitute a general numerical integration scheme like those of classic algorithms (Euler, Runge-Kutta) or QSS methods. Its definition requires analytical integration and its usage beyond first order systems becomes very complex. In addition, the simulations reported are limited to networks of 100 neurons which does not allow to infer the behavior of the algorithm in large networks, and comparisons are only done against a modified second order accurate Runge Kutta (RK) algorithm.

The first use of QSS algorithms for simulation of spiking neurons was presented in (Grinblat, Ahumada, & Kofman, 2012), where the authors explored the use of these algorithms in the simulation of spiking neuron networks based on Izhikevich's model (Izhikevich, 2003a, 2007). The study included the simulation of a single isolated neuron, then a set of several disconnected neurons and finally a network of up to 4000 neurons with 80 input connections. The results showed that, as the size of the network grows, the QSS3 algorithm scaled better than variable step RK algorithms equipped with discontinuity handling routines.

However, the work did not include any type of theoretical analysis and the only errors analyzed via simulation were those corresponding to the voltage variable for the model of a single isolated neuron. In addition, the results were obtained with the QSS3 algorithm using the PowerDEVS toolkit, which is less efficient than the Stand-Alone QSS solver for this type of models. Consequently, the results were not significantly faster than those obtained using classic algorithms like Runge-Kutta-Fehlberg.

In a more recent work (Fernandez, Kofman, & Bergero, 2017) QSS simulations of a SNN model adapted from (Vogels & Abbott, 2005) are presented showing that QSS algorithms are not only efficient but they can be also efficiently parallelized. In this work, the model represents the effect of the incoming spikes in a neuron as a modification of the excitatory and inhibitory conductance, so that each neuron is modeled as a nonlinear system with three state variables (membrane potential and both conductances).

The simulations presented a model with $300,000$ neurons and 300 synaptic connections per neuron with a particular structure (connections were only allowed between neurons that are relatively close to each other). It was shown that the parallel version of the Stand Alone QSS Solver was able to accelerate up to 25 times the simulation using 62 cores. However, the work did not provide any type of theoretical analysis or even any type of error analysis.

In the present work, we shall work with linear LIF models, performing a theoretical analysis that provides upper bounds for the errors introduced by QSS methods in the different variables (membrane voltages and synaptic currents) and also upper bounds for the errors in the firing times and firing rates. In addition, the theoretical analysis will also establish estimates on the number of computations required by the different QSS methods as a function of the accuracy.

We shall also study the performance of QSS methods in terms of CPU time and errors when the size of the network and/or its connectivity grows and we shall also compare it with the performance of the variable step DOPRI5 algorithm using the same pieces of code for computing the state derivatives and the zero crossing functions. To the best of our knowledge, this type of study, which includes theoretical results that are then verified by simulations that clearly exhibit the advantages of QSS algorithms over classic approaches, had been never carried out before.

3. Main Theoretical Results

In this section we describe first the Leaky Integrate and Fire (LIF) model used along this work, taken from (Schmidt et al., 2018). We use two equivalent descriptions of that model, with one and two state variables. Then we derive theoretical properties of their QSS approximations including error bounds and computational complexity.

As expected, we will see that all the error bounds depend linearly on the quantum size and that the computational costs estimations depend on the $1/n$-th power of the quantum where n is the order of the algorithm.

3.1. Model Description

The LIF model used in this work corresponds to that of (Schmidt et al., 2018) representing the synapses with instantaneous jumps and exponential decay. In this case the state variable is the membrane potential $V(t)$ and the corresponding sub-threshold dynamics is described by the following differential equation:

$$\frac{\mathrm{d}V}{\mathrm{d}t} = -\frac{V(t) - E_L}{\tau_m} + \frac{I_s(t)}{C_m} \tag{8}$$

where E_L is the leak potential, τ_m is the membrane time constant, and C_m is the membrane capacity and

$$I_s(t) = I_s(t_l) \cdot e^{-(t-t_l)/\tau_s} \tag{9}$$

is the synaptic current which is represented as an exponentially decaying signal that was restarted after receiving a spike. Here, t_l is the instant of time in which the neuron received the last incoming spike and τ_s is the postsynaptic current time constant.

When the neuron receives a spike, the synaptic current is updated according to the following law

$$I_s(t^+) \leftarrow I_s(t) + \Delta I \triangleq I_s(t) + J \tag{10}$$

where J is the excitatory synaptic strength.

Whenever the membrane potential reaches the firing threshold θ, a spike is emitted by the neuron and the potential is reset to the resting potential V_r.

In addition, the neuron enters a refractory period of duration τ_r in which the membrane potential keeps the value $V(t) = V_r$. After that period, the neuron comes back to the sub-threshold dynamics described by Eq. (8).

We shall consider two equivalent representations for the synaptic current. In the first one we directly use Eq. (9) to compute its evolution so that the entire neuron has a single state variable $V(t)$. We shall refer to this model as the *single-state* model.

In the second representation we shall compute $I_s(t)$ from an additional differential equation given by

$$\frac{\mathrm{d}I_s}{\mathrm{d}t} = -\frac{I_s(t)}{\tau_s} \tag{11}$$

such that the neuron model has now two states ($V(t)$ and $I_s(t)$).

Although both representations are equivalent, their numerical solutions will differ. The single-state model is a time-varying ODE and the accuracy of the approximation given by the QSS methods will depend not only on the quantum, but also on the order of the method and on the rate at which incoming spikes are received. The subthreshold dynamics of the two-state model is linear time-invariant and we shall see that stronger properties and simpler error bounds can be established. For this reason, more accurate results can be expected.

We analyze next the different theoretical properties of the QSS approximation of each representation.

3.2. QSS Sub-Threshold Error Bounds

3.2.1. Two-State Model

QSS algorithms have a global error bound formula for linear time invariant systems depending on the quantum size ΔQ_i used in each variable, given by Eq. (4).

If we apply this formula to a system where the evolution matrix has a triangular structure,

$$\begin{aligned} \frac{\mathrm{d}z_1}{\mathrm{d}t}(t) &= -az_1(t) + bz_2(t) + u_1(t) \\ \frac{\mathrm{d}z_2}{\mathrm{d}t}(t) &= -cz_2(t) + u_2(t) \end{aligned} \tag{12}$$

with $a > 0$, $b > 0$, $c > 0$, as it is the case of the two-state LIF model used in this work, we obtain that the error is bounded as:

$$\begin{aligned} |e_1(t)| &\leq \Delta Q_1 + \left|\frac{2b}{c-a}\right| \Delta Q_2 \\ |e_2(t)| &\leq \Delta Q_2 \end{aligned} \tag{13}$$

A problem with this expression is that the error bound in the first state $e_1(t)$ goes to infinite as c goes to a. In order to obtain a less conservative bound, we shall specialize that formula for this particular triangular case. In addition, we shall extend the existing results to take into account the presence of impulsive input dynamics.

Theorem 1. *Error Bound.*

Consider an LTI system of the form of Eq.(12) with $a > 0$, $b > 0$, $c > 0$, and consider its QSS approximation

$$\begin{aligned} \frac{\mathrm{d}x_1}{\mathrm{d}t}(t) &= -aq_1(t) + bq_2(t) + u_1(t) \\ \frac{\mathrm{d}x_2}{\mathrm{d}t}(t) &= -cq_2(t) + u_2(t) \end{aligned} \tag{14}$$

If a quantum ΔQ_i is used for variable x_i, then the maximum difference between the solutions of both systems is given by

$$|x_1(t) - z_1(t)| \leq \Delta_1 \triangleq \Delta Q_1 + \frac{2b}{max(a,c)}\Delta Q_2 \tag{15}$$

$$|x_2(t) - z_2(t)| \leq \Delta Q_2 \tag{16}$$

provided that the initial state verifies Eqs.(15)-(16).

This theorem, whose proof can be found in A.1, provides a bound for the error in both states of the model of Eq. (8) and Eq. (11) taking $z_1(t) = V(t)$ and $z_2(t) = I_s(t)$.

A problem with this theorem is that it does not take into account the eventual presence of the impulsive inputs corresponding to the arrival of input spikes that are represented by instantaneous changes in the synaptic current. However, the error bounds are still valid in that case taking into account the way in which QSS algorithms treat impulsive inputs during the simulation.

When a neuron receives an input spike, the QSS method instantaneously changes the value of the synaptic current adding the value ΔI that corresponds to the same quantity in which is modified the analytical solution. That way, if at the time of the spike arrival the numerical solution has certain error, immediately after the spike the error will remain unmodified since both solutions (numerical and analytical) will experience an identical change.

More formally, consider a system of the form of Eq.(12) and its QSS approximation given by Eq.(14) and consider an instant of time t_1 in which the system receives an input impulse. If from t_0 to t_1 it is verified that $|\mathbf{x}(t) - \mathbf{z}(t)| \preceq \mathbf{e}_{\max}$, when the input event arrives, both the analytical solution $\mathbf{z}(t)$ and the numerical solution $\mathbf{x}(t)$ are increased or decreased by the same quantity. Thus $|\mathbf{x}(t_1^+) - \mathbf{z}(t_1^+)| = |\mathbf{x}(t) - \mathbf{z}(t)| \preceq \mathbf{e}_{\max}$. This implies that the error bound holds beyond the arrival of an input impulse. Extending this reasoning to the time of the following spike arrival and then using induction, it results that the error bound obtained in the Theorem 1 is still valid in presence of arbitrary sequences of impulsive inputs.

This result ensures that the error in the QSS approximation of the synaptic current $I_s(t) = z_2(t)$ computed from Eq. (11) is bounded according to Eq. (16) even in presence of a sequence of input spikes. Taking into account that the firing of a neuron does not have any effect on its own synaptic current I_s, this error bound is also valid beyond the sub-threshold dynamics provided that the sequence of input spikes received is not changed.

3.2.2. Single State Model

The single state model of Eqs.(8) contains a term $I_s(t)$ given by Eq.(9) that can be treated as an input signal or as a time-varying parameter (in most numerical simulation literature input signals are assumed to be known and are treated as time-varying parameters so that the ODEs are represented by Eq.(1) without the need of an input term). In either case, QSS algorithms will approximate the time-varying terms by piecewise polynomial signals up to one order less than the order of the method. For instance, given a linear system of the form

$$\frac{\mathrm{d}z_1}{\mathrm{d}t}(t) = -az_1(t) + u(t) \tag{17}$$

a QSSn method will approximate it by

$$\frac{\mathrm{d}x_1}{\mathrm{d}t}(t) = -aq_1(t) + v(t) \tag{18}$$

where $q_1(t)$ is the quantized state trajectory (piecewise polynomial of order $n-1$) and $v(t)$ is a piecewise polynomial trajectory of order $n-1$ that approximates the signal $u(t)$. That way, the state derivative $\frac{\mathrm{d}x_1}{\mathrm{d}t}(t)$ follows a piecewise polynomial trajectory of order $n-1$ and the state trajectory $x_1(t)$ follows a piecewise polynomial trajectory of order n.

The difference between a piecewise continuous signal $u(t)$ and its piecewise polynomial approximation of order $n-1$, $v(t)$, can be upper bounded using the remainder of the Taylor expansion as follows:

$$|v(t) - u(t)| \leq \gamma(t - t_k)^n \tag{19}$$

for certain constant γ, where t_k was the time of the last step at which the value and derivatives (up to order $n-1$) of $v(t)$ were made equal to those of $u(t)$. This property holds beyond QSS algorithms, assuming that $u(t)$ is n times differentiable in between the time steps t_k at which the piecewise polynomial approximation $v(t)$ is updated.

Then, using this upper bound for the approximation error of the input trajectory, the following result can be derived to establish an upper bound on the numerical error introduced by a QSS algorithm in the simulation of a system like that of the single-state neuron:

Theorem 2. *Error bound in a non autonomous linear system*

Consider a first order ODE of Eq.(17) with $a > 0$ and $u(t)$ being a piecewise continuous trajectory. Let Eq.(18) be a QSS approximation of order n of that system such that $v(t)$ and $u(t)$ verify Eq.(19) for certain constant γ. Then, assuming that $x_1(t_0) = z_1(t_0)$, it results that

$$|z_1(t) - x_1(t)| \leq \Delta Q_1 + \frac{\gamma}{a}\Delta t_{\max}^n \tag{20}$$

where ΔQ_1 is the quantum and $\Delta t_{\max}$ is the maximum interval between successive steps.

This result, whose proof can be found in A.2, establishes that the error bound depends not only on the quantum but also in the order n of the approximation and in the maximum time $\Delta t_{\max}$ between successive steps.

In the context of the single-state model of the neuron given by Eqs.(8) and (9), the condition of Eq. (19) is verified since a QSS method of order n will approximate the synaptic current $I_s(t)$ using polynomial segments of order $n-1$ and the error term will result proportional to Δt^n where Δt is the time elapsed since the last update of the approximated signal.

In this case, we can expect the error to decrease not only with the quantum size and method order, but also as the input spike rate grows. Anyway, as the quantum becomes smaller, the time between successive changes in $V(t)$ results smaller than the time between input spikes. In that case, $\Delta t_{\max}$ is inversely proportional to the total number of changes in the quantized state. From the activity formula of Eq. (7) the number of steps is inversely proportional to $\Delta Q^{1/n}$. Then, $\Delta t_{\max}$ is proportional to $\Delta Q^{1/n}$ and $(\Delta t_{\max})^n$ is proportional to ΔQ. This implies that the error in Eq. (20) becomes proportional to the quantum size.

3.3. QSS Firing Time and Firing Rate Error Bounds

The most important feature of a spiking neuron is the time at which the output spikes are produced. Thus, it is crucial to analyze the error that the QSS approximations introduce in the firing times. The following theorem provides an upper bound for this error

Theorem 3. *Firing Time Error Bound.*

Consider the LTI system of Eq. (12) and its QSS approximation of Eq. (14).

Let t_θ be the instant of time at which the analytical solution $z_1(t)$ crosses the threshold θ. Let $\tilde{t}_\theta$ be the instant of time at which $x_1(t)$ crosses θ.

Suppose also that $x_1(0) = z_1(0) = V_r$, $u_1(t) = \bar{u}_1$, and assume that $z_2(t)$ is constant *and it verifies*

$$z_2(t) = \bar{z}_2 > \frac{a(\theta + \Delta_1) - u_1}{b} \tag{21}$$

for all $t \in [0, t_\theta]$ with Δ_1 defined in Eq. (15). Then, the maximum difference between the firing times of both systems is given by

$$|t_\theta - \tilde{t}_\theta| < \left| \frac{1}{a} \log \left(1 + \frac{\Delta Q_1 + \dfrac{2b}{\max(a,c)} \Delta Q_2}{\theta - \dfrac{b\bar{z}_2 + u_1}{a}} \right) \right| \tag{22}$$

This theorem, whose proof can be found in A.3, establishes an upper bound on the output spike timing. While precise spike timing is crucial in several applications (where it encodes information), there are occasions in which the information is encoded by the firing rate. For those cases, a straightforward analysis concludes that the difference between the analytical firing period of a two-state neuron with constant synaptic current and that of its QSS approximation is bounded by the error bound in the firing time. This result is formalized by the following corollary:

Corollary 1. *Firing Period Error Bound*

Consider an LTI system of Eq. (12) and its QSS approximation of Eq. (14) with $u_1(t) = \bar{u}_1$, and assume that $z_2(t)$ is constant and it verifies Eq. (21). Suppose also that $z_1(t)$ and $x_1(t)$ are reset to the value V_r whenever they reach θ.

Let T be the time period between two consecutive resets of $z_1(t)$, and let $\tilde{T}$ be the time period between two consecutive resets of $x_1(t)$.

Then, the difference between both periods is bounded by

$$|\Delta T| = |T - \tilde{T}| \leq \Delta T_{\max} \triangleq \left| \frac{1}{a} \log \left(1 + \frac{\Delta Q_1 + \dfrac{2b}{\max(a,c)} \Delta Q_2}{\theta - \dfrac{b\bar{z}_2 + u_1}{a}} \right) \right| \tag{23}$$

The bound given by Eq. (23) can be simplified if we compute the relative error in the period assuming that the synaptic current $z_2(t)$ is large. The following proposition states this result:

Proposition 1. *Relative Error of the Firing Period*

Consider an LTI system of Eq. (12) and its QSS approximation of Eq. (14) under the same assumptions of Corollary 1. Assume also that $z_2(t)$ verifies

$$|b\bar{z}_2 + u_1| >> a(\theta - V_r) \tag{24}$$

Then, the relative error between the firing periods of both systems is bounded by

$$\left| \frac{\Delta T}{T} \right| \leq \frac{\Delta T_{\max}}{T} \simeq \left| \frac{\Delta_1}{\theta - V_r} \right| \tag{25}$$

The proof of this proposition can be found in A.4.

This last result implies that when the synaptic current of the two-state neuron is sufficiently large, the relative error in the period is bounded by a magnitude that becomes independent on that current. Moreover, that error depends linearly on the quantum used in both variables.

Regarding the single-state neuron, very similar results can be easily derived based on the fact that the error in the membrane potential results proportional to the quantum size as analyzed after Theorem 2.

3.4. Activity and Computational Complexity

As introduced in Section 2.4, the activity of order n of a signal allows estimating the number of steps performed by a QSS algorithm in order to approximate that signal. In the context of the QSS approximation of a two-state neuron model, we shall exploit this idea to have a theoretical estimation of the computational costs associated to the simulation with QSS algorithms of different orders and different quantum sizes. For that goal, we shall compute the n-order activity of the synaptic current on a synapse s, $I_s(t)$, since this is the state that experiences more sudden changes. The reason is that a neuron usually receives many more spikes than it produces.

We consider then a neuron receiving a train of input spikes with a constant rate $1/T_s$. In order to compute the activity of the signal I_s between two consecutive input spikes, we can express the analytical solution of $z_2(t)$ in Eq. (12) as

$$z_2(t) = (I_0 + \Delta I)e^{-ct} = I_m e^{-ct}$$

where

$$I_m = I_0 + \Delta I \tag{26}$$

is the synaptic current $z_2(t = 0)$ at the beginning of each period, $\Delta I = J$ is the increment in the synaptic current after receiving a spike, and I_0 is the synaptic current at the end of the period, *i.e.*,

$$z_2(T_s) = (I_0 + \Delta I)e^{-cT_s} = I_0 \tag{27}$$

Thus, we can express I_0 depending on ΔI and T_s as

$$I_0 = \frac{\Delta I \cdot e^{-cT_s}}{1 - e^{-cT_s}} \tag{28}$$

The activity of order n of $z_2(t)$ in a period can be then expressed as

$$
\begin{aligned}
A^{(n)}_{0,T_s} &= \int_0^{T_s} \left| \frac{\dfrac{d^n(I_m \cdot e^{-ct})}{dt^n}}{n!} \right|^{\dfrac{1}{n}} dt = \int_0^{T_s} \left| \frac{I_m \cdot (-c)^n \cdot e^{-ct}}{n!} \right|^{\dfrac{1}{n}} dt \\
&= \int_0^{T_s} \left(\frac{I_m \cdot c^n \cdot e^{-ct}}{n!} \right)^{\dfrac{1}{n}} dt = \left(\frac{I_m}{n!} \right)^{\dfrac{1}{n}} c \cdot \int_0^{T_s} e^{-\dfrac{c}{n} \cdot t} dt \\
&= \left(\frac{I_m}{n!} \right)^{\dfrac{1}{n}} c \left(\left. \frac{e^{-\dfrac{c}{n} \cdot t}}{-\dfrac{c}{n}} \right|_0^{T_s} \right) = \left(\frac{I_m}{n!} \right)^{\dfrac{1}{n}} n \left(1 - e^{-\dfrac{c}{n} \cdot T_s} \right)
\end{aligned}
$$

From Eq. (27) we have

$$
e^{-cT} = \frac{I_0}{I_0 + \Delta I} = \frac{I_0}{I_m} \tag{29}
$$

and then

$$
A^{(n)}_{0,T_s} = \left(\frac{I_m}{n!} \right)^{\dfrac{1}{n}} n \left(1 - (\frac{I_0}{I_m})^{\dfrac{1}{n}} \right) = n \cdot \frac{I_m^{1/n} - I_0^{1/n}}{n!^{1/n}} \tag{30}
$$

where I_m and I_0 can be computed from the system parameters using Eqs.(26) and (28).

Then, from Eq. (30) and using the estimate of the number of steps of Eq. (7), we can estimate the number of steps required by a QSS method of order n to approximate one period of the synaptic current as:

$$
k^{(n)}_{x_i(t_0 \cdot t_f)}(\Delta Q_i) \approx \frac{A^{(n)}_{0,T_s}}{(\Delta Q_i)^{1/n}} = n \frac{I_m^{1/n} - I_0^{1/n}}{(n! \Delta Q_i)^{1/n}} \tag{31}
$$

Equation (31) shows that the number of steps per period in I_s is inversely proportional to the $1/n$-th power of the quantum in the current ΔQ_i.

Regarding the number of steps in the membrane potential $V(t)$, we can take into account that its variation is mainly driven by the synaptic current. During a QSS simulation, each step in the synaptic current will change the derivative of the membrane potential, which will possibly shorten the time of its next change. Thus, unless a very large or a very small quantum is used in V, we can expect that the number of steps in the membrane potential is similar to that on the synaptic current.

Regarding the single-state model of Eq. (8) and Eq. (9) it results more involved to compute the n-th order activity since the analytical solution for $V(t)$ has a more complex expression. Anyway, it is still valid that the number of steps will be inversely proportional to the $1/n$-th power of the quantum size as established by Eq. (7).

4. Simulation Results and Performance Analysis

In this section we compare the performance of different numerical algorithms under different accuracy settings in the simulation of a single neuron. Then, we extend the analysis to populations of neurons with and without synaptic connections between them.

In all the simulations we used the *Stand Alone QSS Solver* tool (Fernández & Kofman, 2014) running on an Intel Core i5-9400 CPU @ 2.90GHz Intel i5 desktop computer under Ubuntu 20.04 OS. We considered both, the single and the two-state representations of the neuron model as presented in Section 3.1 with the set of parameters described in Table 1.

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
$EL = -65$ mV	leak potential
$J = 87.8$ pA	excitatory synaptic strength
$\nu_{bg} = 8$ spikes/s	average external spike rate
$k_{ext} = 940$	external inputs per population

Table 1. Model parameters taken from (Potjans & Diesmann, 2014).

The models used in this section can be downloaded from `https://fceia.unr.edu.ar/~kofman/files/SNNmodels.zip` and the results reported can be reproduced using the Stand Alone QSS Solver, available at `https://github.com/CIFASIS/qss-solver`.

4.1. Single Neuron Model

We simulated the models presented in Section 3.1 with different QSS methods (QSS, QSS2, QSS3)[2] changing the quantum value for the state variables ($V(t)$, and $I_s(t)$ in the two-state model). The quantum values were chosen from around one order of magnitude below the variation of the corresponding signals up to four orders of magnitude below, that is, we changed ΔQ_{I_s} from 10^{-1}nA to 10^{-4}nA and ΔQ_V from 10^{-1}mV to 10^{-4}mV.

In order to measure the errors, we also computed a reference solution using the classic DASSL solver with a very low tolerance (five orders of magnitude below the minimum tolerance settings used for the QSS algorithms). In addition, we simulated the models using DOPRI5 algorithm under different accuracy settings in order to compare the performance of QSS methods with that of a classic ODE solver. Among the

[2] We did not include Linearly Implicit QSS Methods because the models are not stiff.

different classic solvers that can be selected in the Stand Alone QSS Solver (DOPRI5, DASSL, CVODE, and IDA), DOPRI5 was the one exhibiting the best performance.

Besides using the parameters of Table 1, we adopted the following initial conditions: $V(0) = -65$ mV and $I_s(0) = 0.4$ nA in all the experiments. The input spike train of the neuron is a Poisson process with stationary rate $\nu_{\mathrm{ext}} = k_{ext}\nu_{bg}$ generated from a pseudo-random number generator[3] with fixed seed such that the different simulation runs can be compared under identical conditions.

The final simulation time was set to 1 second, after which the neuron received a total of $7,513$ input spikes and emitted 10 output spikes.

Based on the results obtained, we analyze next the different errors and computational costs associated to each simulation.

4.1.1. Error Analysis

We computed errors in two different ways associated to each simulation:

- Maximum absolute error e_{max}: is the largest absolute value of the difference between the reference and the approximate solution.
- Mean absolute error e_{mean}: is the mean value of the absolute difference between the reference and the approximate solution.

For the membrane potential V we measured these errors up to the first emitted spike (sub-threshold dynamics) and only the mean absolute error for the whole simulation. We did not report the maximum membrane potential error because it will be always equal to the voltage jump between the threshold and the reset potential (even if the difference in the firing times is infinitely small).

For the current I_s we measured the maximum and the mean absolute errors during the whole simulation, only in the model with two state variables. In the single-state model, we also report the number of emitted spikes in each simulation (we do not report it in the two-state model because it was always correct).

Regarding the firing times, we measured the error between the time of the i-th firing in the reference solution and the time of the i-th firing of each numerical solution, reporting the mean firing time error (only for the simulations that give the correct number of emitted spikes). For the two-state model, we also reported the theoretical error bound given by Eq. (22).

All these errors are reported in Table 2 (two-state model) and Table 3 (single-state model). In all the tables, the column labeled as *Tol.* represents the quantum ΔQ used in QSS methods and the Relative and the Absolute Tolerance used in DOPRI5 algorithm.

The errors reported correspond to those obtained using a particular pseudo-random sequence for the Poisson input spike train. In Appendix B we also report the mean and the standard deviation of these errors after using 30 different pseudo-random sequences. Those results show that the errors obtained using different sequences are very similar, a fact that can be explained from the high rate at which input spikes are received. Thus, in the rest of the work we shall analyze the results obtained using a single pseudo-random sequence.

A noticeable feature is that the errors in all the simulations with the two-state model, irrespective of the quantum size, are small enough to preserve the qualitative features of the reference solution (for instance, the number of emitted spikes, the length of the period between spikes, or the signal shape). This fact was expected from

[3] The QSS Solvers uses the standard stdlib C library for generating pseudo-random sequences.

Method	Tol	I_s [nA]		V_{sub} [mV]		V [mV]	t_θ [s]	
	[nA,mV]	e_{mean}	e_{max}	e_{mean}	e_{max}	e_{mean}	e_{mean}	e_{theor}
QSS	1E-1	2.38E-2	7.25E-2	3.95E-2	1.41E-1	5.15E-2	7.17E-5	2.99E-4
	1E-2	2.51E-3	5.26E-3	4.36E-3	1.45E-2	5.12E-3	8.10E-6	2.95E-5
	1E-3	2.49E-4	6.01E-4	4.90E-4	1.68E-3	5.44E-4	9.65E-7	2.94E-6
	1E-4	2.51E-5	8.14E-5	4.58E-5	1.57E-4	4.85E-5	5.62E-8	2.94E-7
QSS2	1E-1	9.32E-3	8.53E-2	3.44E-2	1.18E-1	5.93E-2	9.41E-5	2.99E-4
	1E-2	3.52E-3	1.34E-2	4.33E-3	1.49E-2	5.99E-3	6.32E-6	2.95E-5
	1E-3	2.68E-4	1.27E-3	2.74E-4	1.46E-3	2.76E-4	3.24E-7	2.94E-6
	1E-4	2.53E-5	1.30E-4	2.58E-5	7.15E-5	3.04E-5	5.99E-8	2.94E-7
QSS3	1E-1	1.28E-2	8.65E-2	2.82E-2	8.79E-2	4.01E-2	6.21E-5	2.99E-4
	1E-2	3.67E-3	1.16E-2	3.73E-3	1.34E-2	5.57E-3	5.17E-6	2.95E-5
	1E-3	2.78E-4	1.14E-3	2.84E-4	1.19E-3	2.98E-4	4.58E-7	2.94E-6
	1E-4	2.47E-5	1.14E-4	2.24E-5	1.10E-4	2.68E-5	4.09E-8	2.94E-7
DOPRI5	1E-1	5.92E-2	4.56E-1	5.91E-2	8.67E-2	7.82E-2	1.17E-4	-
	1E-2	5.86E-2	4.42E-1	6.95E-3	1.03E-2	3.55E-2	1.02E-4	-
	1E-3	5.89E-2	4.39E-1	6.96E-4	1.16E-3	1.24E-2	4.60E-5	-
	1E-4	5.89E-2	4.39E-1	6.68E-5	1.35E-4	3.29E-3	1.21E-5	-

Table 2. Errors in the two-state model.

Method	Tol	V_{sub} [mV]		V [mV]	Output	t_θ [s]
	[nA,mV]	e_{mean}	e_{max}	e_{mean}	spikes	e_{mean}
QSS	1E-1	2.02E+0	1.34E+1	4.06E+0	27	-
	1E-2	4.32E-1	7.72E-1	2.23E+0	17	-
	1E-3	9.76E-2	2.59E-1	7.24E-1	12	-
	1E-4	6.85E-2	1.95E-1	7.80E-2	10	7.99E-5
QSS2	1E-1	3.17E-1	5.05E-1	1.22E+0	7	-
	1E-2	6.20E-2	9.49E-2	1.25E-1	10	1.90E-4
	1E-3	4.59E-3	7.22E-3	1.05E-2	10	1.39E-5
	1E-4	4.80E-4	6.74E-4	7.49E-4	10	1.19E-6
QSS3	1E-1	5.50E-2	1.85E-1	7.96E-2	10	7.38E-5
	1E-2	1.89E-2	3.17E-2	3.76E-2	10	3.68E-5
	1E-3	3.92E-3	5.53E-3	7.32E-3	10	7.78E-6
	1E-4	4.39E-4	6.10E-4	6.49E-4	10	9.40E-7
DOPRI5	1E-3	3.19E-1	4.46E-1	1.56E+0	13	-
	1E-4	3.98E-2	5.78E-2	7.42E-2	10	1.20E-4
	1E-5	2.71E-3	3.65E-3	2.59E-2	10	8.58E-5
	1E-6	2.26E-4	2.97E-4	9.34E-3	10	3.07E-5

Table 3. Errors in QSS Simulation of the single-state model. Note that the number of output spikes in the reference solution is 10.

the distinctive theoretical properties of QSS methods which are not shared by classic numerical integration schemes. In discrete time algorithms, for instance, increasing the step size may result in unstable solutions and/or unacceptable errors due to the missing of zero crossings (Cellier & Kofman, 2006).

In the single-state model the errors are significantly larger (particularly for lower order algorithms, which is consistent with Theorem 2). As a consequence, some features of the qualitative behavior are not properly preserved, resulting, for instance, in a different number of emitted spikes.

An example of this is illustrated in Figure 3 where the evolution of the membrane potential is shown for QSS2 varying the accuracy (ΔQ = 1E-1 and ΔQ = 1E-2) along with the reference solution (DASSL). Here we can observe an almost perfect trajectory for the smallest quantum, where the trajectories cannot be distinguished with the naked eye. However, for the largest quantum, a small sub-threshold error in the membrane potential causes a spike in an instant where the reference solution does not actually reach the threshold.

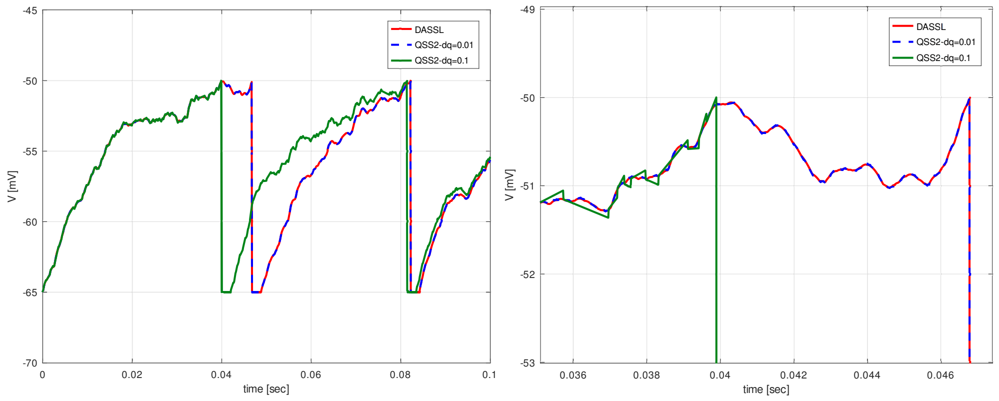

Figure 3. Membrane potential trajectories with QSS2 and the reference high accuracy solution (DASSL). Full simulation (left) and partial details (right).

*4.1.1.1. **Synaptic Current.*** The maximum error in I_s (in the two-state model) resulted very similar to the quantum size. According to Theorem 1, that error should be bounded by the quantum itself. However, due to numerical inaccuracies caused by round off errors these maximum errors resulted slightly larger. Measured mean absolute errors were about one order of magnitude less than the quantum size.

The classic algorithm DOPRI5, in turn, produces results in the current that are very similar irrespective of the tolerance, and the errors result larger than those obtained with QSS in all cases.

*4.1.1.2. **Sub-Threshold Membrane Potential.*** In the two-state model, the maximum sub-threshold error was in all cases very similar to the quantum size. According to Eq. (15) in Theorem 1, the maximum sub-threshold error is $|e_V| \leq \Delta Q_V + \frac{2b}{\max(a,c)}\Delta Q_{I_s} = \Delta Q_V + 4 \cdot 10^6 \Delta Q_{I_s} = 5\Delta Q_V$ since we are using in all cases[4] $\Delta Q_V = 10^6 \Delta Q_{I_s}$. This shows that the theoretical estimate was conservative for this

[4] The quantum was expressed in mV and nA in the different tables, but the model considers that the membrane potential and the synaptic current are measured in SI units (V and A, respectively).

state. Like in the previous case, the mean absolute errors were always below the quantum size. DOPRI5 produces similar errors to those of QSS methods in the membrane potential.

In the single-state model simulations, in all cases the errors were larger than those obtained in the other model with the same accuracy. As expected from Eq. (20) the lower order algorithms present larger errors. In the first order method (QSS) these errors can even affect some qualitative features (like the number of emitted spikes), except for the highest accuracy settings. The second and third order schemes obtain errors that are in the order of the quantum size. The best results in terms of accuracy, as expected, are obtained with the third order schemes since they approximate better the time varying input signal $I_s(t)$. QSS3, in particular, preserves the number of emitted spikes for all the considered quantum sizes.

Here, DOPRI5 produces significantly larger errors than QSS2 and QSS3 algorithms so the tolerances were reduced by two orders of magnitude to obtain similar results in terms of accuracy.

4.1.1.3. ***Complete Membrane Potential Trajectory.*** In the two-state model, since the spikes are emitted with a very small time error, the resulting mean error of the membrane potential is in the order of the quantum size.

In the single-state model, there are several situations in which the spikes are not correctly emitted producing a large mean absolute error in the membrane potential (notice the situation shown in Figure 3). As expected, third order schemes are the most accurate.

Regarding DOPRI5, the errors in the membrane potential resulted larger than those of QSS methods in all cases.

4.1.1.4. ***Firing times.*** The firing time errors observed in the two-state model were almost negligible. Taking into account that the firing period of the neuron is on the order of $\frac{1}{10} = 0.1$ sec., the observed errors varied from 0.1% to 0.0001% of this period according to the quantum size.

The observed firing time errors were about one order of magnitude smaller than the theoretical error bound obtained from Eq. (22) that does not take into account the accumulation among different periods and considers a constant input. Thus, the behavior of the QSS algorithms regarding the firing times, which is a crucial feature of a spiking neuron simulation, was even better than what could be expected.

In the single-state model the errors were larger, and in some cases they could not be measured because the number of emitted spikes did not coincide with that of the reference solution. Anyway, the higher order schemes still obtained small errors at least for high accuracy settings.

DOPRI5 firing time errors resulted larger than those produced by QSS algorithms, even when using smaller tolerance settings.

4.1.2. Computational Costs

The results regarding computational costs are summarized in Table 4. A simple observation of these results shows that the number of steps performed by each algorithm grows with the accuracy settings. This is, the number of steps grows with the inverse of the quantum. This growth, as expected, is inversely proportional to the quantum in QSS, inversely proportional to the square root of the quantum in QSS2 and inversely proportional to the cubic root of the quantum in QSS3. Execution times change accordingly.

		Two-state model			Single-state model	
Method	Tol	Simulation Steps		CPU Time	Simulation Steps	CPU Time
	[nA,mV]	I_s	V	[ms]	V	[ms]
QSS	1E-1	11,308	3,753	2.5	5,118	1.8
	1E-2	70,536	42,140	12.6	44,874	6.8
	1E-3	670,681	434,965	116.7	435,617	48.7
	1E-4	6,672,923	4,373,657	1,115.3	4,328,986	463.7
QSS2	1E-1	8,150	1,982	2.0	2,036	1.6
	1E-2	11,167	9,361	3.6	9,299	2.6
	1E-3	35,176	37,937	10.7	37,224	6.4
	1E-4	90,798	114,655	28.9	11,3969	17.3
QSS3	1E-1	7,841	2,970	4.4	3,164	3.3
	1E-2	11,352	7,151	7.4	7,597	4.5
	1E-3	18,683	14,403	12.0	12,555	6.2
	1E-4	31,047	28,452	20.9	22,356	9.7
DOPRI5	1E-1	7,609	7,609	2.9	7,683	3.0
	1E-2	7,651	7,651	2.9	7,648	3.3
	1E-3	7,880	7,880	2.9	7,747	3.0
	1E-4	8,554	8,554	3.1	8,241	3.1

Table 4. Computational costs in the simulation of the two-state and the single-state models

In all QSS simulations the CPU times using the single-state model are less than the ones using the two-state model for the same accuracy settings. However, under identical accuracy settings, the errors obtained using the single-state model are larger. This fact can be observed in Figure 4, that plots the CPU time against the mean error in the sub-threshold membrane potential for both models and for all QSS methods. The plot also shows that QSS2 offers the best trade-off in the two-state model (unless a very small error is requested) while QSS3 obtains the best performance in the single-state model (unless a large errors are accepted). DOPRI simulations of a single neuron are faster on both models for high accuracy settings. However, even using lower tolerance settings, they have larger timing errors than QSS algorithms as it can be observed in Tables 2–3.

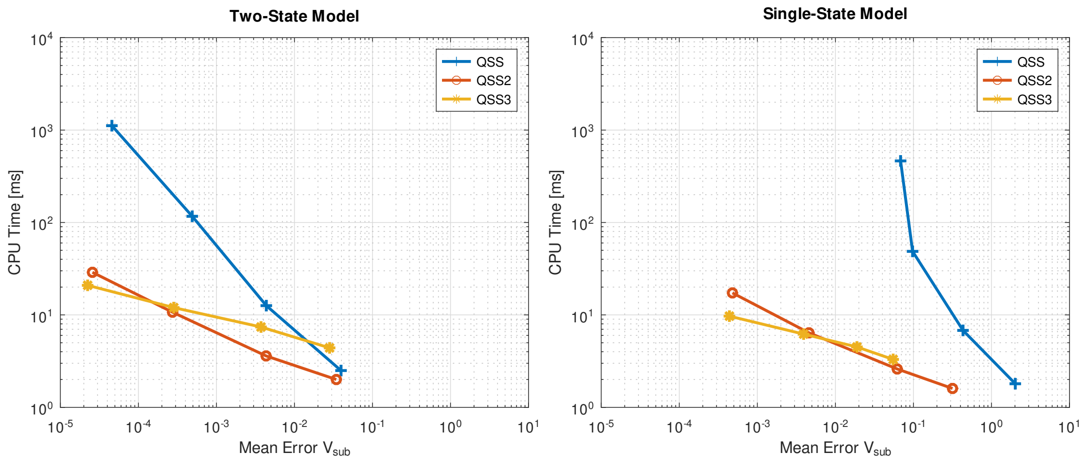

Figure 4. CPU Times vs. Mean Error in the sub-threshold membrane potential (V_{sub}) using different QSS algorithms.

In order to check that the mean number of steps performed by each method between consecutive spikes is in agreement with the analysis of Section 3.4, we compute next the activity of orders one to three and the expected number of steps per period of the synaptic current for the different accuracy settings in the two-state model.

The synaptic current variation is $\Delta I = J = 8.78 \cdot 10^{-2}$ nA. Then, from Eq. (28) with $c = \frac{1}{\tau_s}$ and $T = \frac{1}{\nu_{\text{ext}}}$ it results $I_0 \approx 0.519$nA and $I_m = I_0 + \Delta I \approx 0.607$ nA. Thus, according to Eq. (30) and Eq. (31), the activity of orders 1 to 3 and the expected number of steps[5] in each period of the synaptic current for the different quantum values are those reported in Table 5.

Order	A$^{(n)}$ [nA]	ΔQ_i [nA]	$k^{(n)}(\Delta Q_i)+1$	QSS$_n$
1	$I_m - I_0 = 0.0878$	1E-1	1.878	1.51
		1E-2	9.78	9.39
		1E-3	88.8	89.28
		1E-4	879	888.3
2	$\dfrac{2(\sqrt{I_m} - \sqrt{I_0})}{\sqrt{2!}} = 0.083$	1E-1	1.26	1.08
		1E-2	1.83	1.49
		1E-3	3.62	4.68
		1E-4	9.28	12.09
3	$\dfrac{3(\sqrt[3]{I_m} - \sqrt[3]{I_0})}{\sqrt[3]{3!}} = 0.071$	1E-1	1.15	1.04
		1E-2	1.33	1.51
		1E-3	1.71	2.49
		1E-4	2.53	4.13

Table 5. Act. of order n, estimated and measured number of steps per period in I_s.

A very close agreement can be noticed for all methods. A final remark is that it is impossible to simulate with less than one step per period without missing input spikes. Table 5 shows that the number of steps per period in second order methods (up to ΔQ_i = 1E-2) and third order methods (up to ΔQ_i = 1E-3) are close to that minimum.

[5] The expected number of steps is the result obtained from Eq. (31) plus one step corresponding to the arrival of the input spike.

4.2. Model of Multiple Neurons

We consider now a model formed by N unconnected instances of the neuron model in order to analyze the growth of the computational costs with the number of neurons when using QSS and classic numerical algorithms.

In these configurations (using the single-state and the two-state model), each neuron has an independent input spike train following, as before, a Poisson distribution with a constant rate v_{ext} generated from a pseudo-random sequence with a fixed seed.

For all neurons, the parameters were those of Table 1. The initial states were randomly chosen with uniform distribution $V(t = 0) \sim U[-65, -64]$mV, $I_s(t = 0) \sim U[0.4, 0.5]$nA.

We simulated both, the single-state and the two-state models varying the number of neurons from 10 to $10,000$ until a a final simulation time of $t_f = 0.1$ sec. We used QSS2 for the two-state model and QSS3 for the single-state model, as these methods offered the best performance in terms of CPU-time vs. accuracy in Figure 4. We selected different quantum values in order to measure the growth of the computational costs with the accuracy settings and to check for convergence in the number of emitted spikes. We also simulated the models using DOPRI5.

The results are reported in Table 6 and the growth of the simulation time (CPU time) with respect to the number of neurons (N) is plotted in Figure 5.

Tol	N	Steps	CPU Time	Out	Steps	CPU Time	Out
			[ms]	Spikes		[ms]	Spikes
		Two-state model - QSS2			Single-state model - QSS3		
1E-1	10	9,589	2.0	10	10,738	3.6	10
	100	95,622	23.8	59	106,990	36.6	61
	1,000	955,748	329.4	660	1,069,560	460.1	666
	10,000	9,532,780	5,538.3	6,280	10,676,992	6,649.7	6,389
1E-2	10	21,798	3.7	9	15,225	5.1	9
	100	218,267	44.1	53	151,505	51.8	54
	1,000	2,179,150	631.4	628	1,514,438	643.6	637
	10,000	21,776,481	10,622.2	5,901	15,122,208	9,324.5	6,015
1E-3	10	73,694	10.0	9	20,216	6.4	9
	100	736,682	116.6	52	201,980	67.8	52
	1,000	7,360,380	1,778.2	623	2,016,058	860.7	625
	10,000	73,541,327	29,225.3	5,839	20,136,964	12,423.9	5,874
1E-4	10	206,240	25.9	9	30,033	9.2	9
	100	2,062,500	305.7	52	300,646	99.7	52
	1,000	20,601,643	4,649.8	623	3,001,045	1,234.7	623
	10,000	205,876,468	77,555.3	5,837	29,987,701	18,209.9	5,840
		Two-state model - DOPRI			Single-state model - DOPRI		
1E-4	10	7,607	12.7	9	7,599	14.3	9
	100	75,450	1,081.4	52	75,410	1,211.1	52
	1,000	754,199	108,660.0	624	753,749	120,215.0	624
	10,000	7,523,758	1.104E+7	5852	7,520,462	1.20E+7	5,852

Table 6. Computational costs of the both models with N disconnected neurons

We can observe that:

- The number of steps grows linearly with the number of neurons N in QSS methods for all accuracy settings. This is an almost obvious result since each step is

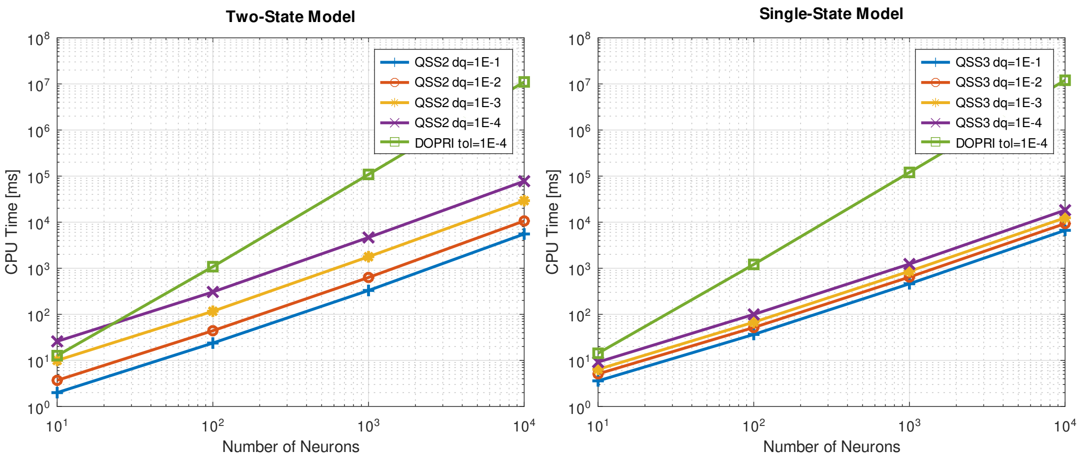

Figure 5. CPU Time vs Number of Neurons using different tolerance settings.

local to a state variable.

- In DOPRI5, the number of steps also grows linearly with N. This is due to the fact that the rate of occurrence of discontinuities grows with N and thus the maximum step size becomes proportional to $1/N$.
- The CPU time grows with $N \log(N)$ in QSS methods. This is due to the fact that the number of steps grows linearly with N and the simulation engine uses a binary-tree scheduler to find the time of the next change, which adds a $\log N$ extra cost per step.
- The CPU time in DOPRI5 grows quadratically. The reason is that the number of steps grows with N and the cost of each step is also proportional to N since DOPRI5 steps are computed on the whole system.
- When the number of neurons is large, the number of emitted spikes in QSS methods differs depending on the quantum, but it eventually converges as the quantum becomes smaller. The reason is that a situation like that of Figure 3, where a very small error in the membrane potential can cause a spurious or a missing spike, becomes more likely as the number of neurons grows.
- DOPRI5 errors become negligible as the number of neurons grows. The reason is that the step size becomes smaller (limited by the time elapsed between discontinuities) and the numerical errors fall far below the accuracy settings.

4.3. Model of a Network of Neurons

We consider now a model of $N = 10,000$ interconnected neurons, with $8,000$ of excitatory type and $2,000$ of inhibitory type.

The structure of the network was formed such that each neuron has m randomly chosen input synaptic connections, where 80% of these incoming connections come from excitatory neurons and the remaining 20% come from inhibitory neurons following the parameters found in (Schmidt et al., 2018).

Taking $m = 100$, for instance, every neuron receives spikes from 80 excitatory neurons and from 20 inhibitory neurons. Those 80 excitatory neurons are randomly chosen (for that specific neuron) out of the 8000 excitatory neurons with equal probability. Similarly, the 20 inhibitory neurons are also randomly chosen out of the 2000 inhibitory neurons (also with equal probability).

That way, each neuron has both, its own input spike train with the same Poisson distribution as before, and the spikes received from their incoming synaptic connections.

The synaptic strength of the excitatory connections were randomly chosen with values uniformly distributed in the interval $J \pm 0.1J$. Inhibitory synaptic strengths were chosen with values $gJ \pm 0.1gJ$ with the scaling parameter $g = 5$. This scaling parameter was selected such that the rate of the emitted spikes does not change significantly with the number m of incoming connections. A synaptic delay $\tau_d =$ 1E-3 sec. was used.

We simulated the system varying the number of connections of each neuron m from 10 to $1,000$ in order to measure the growth in the computational costs as the network increases its connectivity. We used QSS2 method for the two-state model and QSS3 for the single-state model. In both cases, we also changed the quantum size until observing convergence in the number of emitted spikes. Like in the previous case, the final simulation time was $t_f = 0.1$ sec.

The simulation results for both models are reported in Table 7.

		Two-state model - QSS2			Single-state model - QSS3		
Tol	m	Steps	CPU Time [ms]	Out Spikes	Steps	CPU Time [ms]	Out Spikes
1E-1	10	10,038,883	6,265.6	6,367	10,700,559	7,307.3	6,472
	100	10,203,316	6,934.8	7,599	10,798,311	7,290.1	7,342
	1,000	11,081,929	10,263.8	7,163	11,365,275	9,106.4	7,449
1E-2	10	20,579,013	11,262.5	5,950	15,164,106	9,678.8	6,096
	100	20,892,009	12,528.5	6,967	15,483,527	10,317.5	7,234
	1,000	24,148,242	17,226.2	6,481	18,869,024	14,067.2	8,087
1E-3	10	73,279,806	40,832.9	5,923	20,181,053	13,537.3	5,952
	100	73,318,426	39,057.1	6,963	20,523,676	13,167.6	6,733
	1,000	80,969,462	43,008.3	8,520	23,996,908	16,744.9	5,642
1E-4	10	205,737,401	88,732.6	5,923	30,017,686	21,058.2	5,923
	100	204,734,130	93,031.9	6,956	30,245,928	23,277.5	6,917
	1,000	214,334,158	100,814.0	7,112	34,979,710	23,924.3	7,610
1E-5	10	620,024,618	272,732.0	5,923	51,816,833	32,689.3	5,923
	100	615,513,913	274,602.0	6,956	51,796,598	34,278.3	6,956
	1,000	632,190,653	283,982.0	7,071	56,367,979	39,815.5	7,023

Table 7. Computational Costs in a network of interconnected neurons (two-state and single-state models)

Analyzing the results, we can make the following remarks:

- The number of steps performed by each method (and the CPU-time) is similar to that of the disconnected network of $N = 10,000$ neurons in Section 4.2. It only grows by a small percentage for $m = 1,000$. This can be explained by the fact that the number of spikes that a neuron receives from the network (except for $m = 1,000$) is significantly smaller than the number of spikes it receives from its Poisson-distributed input train.
- The number of emitted spikes converges only for a small quantum, particularly when m is large. The reason is that now the error introduced by a missing or a spurious spike is propagated through the network, and this propagation becomes faster and wider as the connectivity grows.

In this case, the chaotic nature of the system makes impossible that two simulations under different accuracy settings provide identical results in the long term. Thus, the *best* quantum size selection would depend on the goals according to the application. A network with $N = 10,000$ neurons and $m = 1,000$ incoming connections per neuron can be simulated in about 9-10 seconds using a quantum size $\Delta Q = 0.1$ and QSS2 or QSS3 method or in about 40 seconds using QSS3 and a quantum size that produces much less spurious or missed spikes.

While the number of spurious or missed spikes seems too large for large quantum sizes, the number of emitted spikes also changes significantly when some model parameters or initial states are slightly changed. This reveals that the problem is the chaotic nature of the system rather than the errors introduced by the numerical approximation itself. Having said that, further research is needed to check if systematic long-term errors are introduced by the QSS approximation, which could, for example, increase or decrease the mean firing rates.

5. Conclusions

We presented an exhaustive analysis of the performance of QSS algorithms in the simulation of Leaky Integrate and Fire spiking neurons. We first derived theoretical properties that established error bounds and computational cost estimates for the QSS simulation of a single neuron. Such properties cannot be in general obtained for classic numerical algorithms in these type of models.

Then, we performed comprehensive simulation experiments using different QSS methods under different accuracy settings. The results of these experiments corroborated the theoretical analysis, exhibiting promising advantages of the QSS algorithms with respect to classic numerical integration schemes. It is remarkable that the second and third order QSS methods integrate the model with very small errors performing about one step per input spike period. It is also remarkable that the errors in the firing times were almost negligible.

We then simulated networks with a growing number of neurons and with a growing level of connectivity. As the number of neurons N grows, the computational costs grow with $N \log(N)$ due to the logarithmic cost associated to the binary tree scheduler while classic discrete-time algorithms grow with N^2 due to the reduction of the time between discontinuities.

When the level of connectivity grows, CPU times are not significantly affected but the model becomes more chaotic and the effect of a missing or a spurious spike (caused by a small error in the membrane potential) is rapidly and widely propagated throughout the network.

A general conclusion of the analysis is that QSS methods (particularly QSS2 for the two-state model and QSS3 for the single-state model) allow to simulate large spiking neural networks with high accuracy and very low computational costs. This is achieved without any modification or specialization of the algorithms, as it is a natural consequence of the way the QSS methods work in the Stand-Alone QSS Solver.

We are currently working on specializing the simulation algorithm for the features of this model. In particular, the presence of the synaptic delay implies that there is no direct interaction between the different neuron during this period. This fact can be exploited, as it was done in (Hanuschkin, Kunkel, Helias, Morrison, & Diesmann, 2010b), to perform a sequential simulation of the N neurons for that period of time without taking care about the ordering in which the steps are performed. With this, we can avoid the factor $\log(N)$ and reduce the computational cost to scale linearly with the number of neurons N. We are also working on exploiting this idea in the context of parallel simulation.

Future work can also be done to further specialize the QSS methods themselves. For instance, in the two-state model we could enforce a step in the membrane potential V whenever a step in the synaptic current I_s is performed. That will synchronize the steps in both states improving in this case the computational costs and the accuracy.

It is also worth exploring the possibility of using a smaller quantum size as the membrane potential approaches the firing threshold to prevent spurious or missing spikes. This can be easily done by using a membrane potential relative to the threshold such that it is close to zero near the firing condition. That way, using logarithmic quantization will reduce the quantum near the threshold.

Acknowledgments

This work was supported by ANPCYT under Grant PICT–2017 2436, CONICET under Grant PIP 2022-2024 11220210100093CO and CNRS PRIME eXplAIn research team.

References

Brette, R., Rudolph, M., Carnevale, T., Hines, M., Beeman, D., Bower, J. M., ... others (2007). Simulation of networks of spiking neurons: a review of tools and strategies. *Journal of computational neuroscience*, *23*(3), 349–398.

Brunel, N., Hakim, V., & Richardson, M. J. (2003). Firing-rate resonance in a generalized integrate-and-fire neuron with subthreshold resonance. *Physical Review E*, *67*(5), 051916.

Burkitt, A. N. (2006). A review of the integrate-and-fire neuron model: I. homogeneous synaptic input. *Biological cybernetics*, *95*(1), 1–19.

Castro, R., & Kofman, E. (2015). Activity of order n in continuous systems. *Simulation*, *91*(4), 337–348.

Cellier, F., & Kofman, E. (2006). *Continuous System Simulation*. New York: Springer.

Cohen, S. D., Hindmarsh, A. C., & Dubois, P. F. (1996). Cvode, a stiff/nonstiff ode solver in c. *Computers in physics*, *10*(2), 138–143.

D'Abreu, M. C., & Wainer, G. A. (2005). M/cd++: modeling continuous systems using modelica and devs. In *Modeling, analysis, and simulation of computer and telecommunication systems, 2005. 13th ieee international symposium on* (pp. 229–236).

Dormand, J. R., & Prince, P. J. (1980). A family of embedded runge-kutta formulae. *Journal of computational and applied mathematics*, *6*(1), 19–26.

Ermentrout, B. (1996). Type i membranes, phase resetting curves, and synchrony. *Neural computation*, *8*(5), 979–1001.

Fernandez, J., Kofman, E., & Bergero, F. (2017). A parallel quantized state system solver for odes. *Journal of Parallel and Distributed Computing*, *106*, 14–30.

Fernández, J., & Kofman, E. (2014). A Stand-Alone Quantized State System Solver for Continuous System Simulation. *Simulation: Transactions of the Society for Modeling and Simulation International*, *90*(7), 782–799.

Fourcaud-Trocmé, N., & Brunel, N. (2005). Dynamics of the instantaneous firing rate in response to changes in input statistics. *Journal of computational neuroscience*, *18*(3), 311–321.

Fourcaud-Trocmé, N., Hansel, D., Van Vreeswijk, C., & Brunel, N. (2003). How spike generation mechanisms determine the neuronal response to fluctuating inputs. *Journal of neuroscience*, *23*(37), 11628–11640.

Gerstner, W., Kistler, W. M., Naud, R., & Paninski, L. (2014). *Neuronal dynamics: From single neurons to networks and models of cognition.* Cambridge University Press.

Grinblat, G. L., Ahumada, H., & Kofman, E. (2012). Quantized state simulation of spiking neural networks. *Simulation*, *88*(3), 299–313.

Hansel, D., Mato, G., Meunier, C., & Neltner, L. (1998). On numerical simulations of integrate-and-fire neural networks. *Neural Computation*, *10*(2), 467–483.

Hanuschkin, A., Kunkel, S., Helias, M., Morrison, A., & Diesmann, M. (2010a). A general and efficient method for incorporating precise spike times in globally time-driven simulations. *Frontiers in neuroinformatics*, *4*, 113.

Hanuschkin, A., Kunkel, S., Helias, M., Morrison, A., & Diesmann, M. (2010b). A general and efficient method for incorporating precise spike times in globally time-driven simulations. *Frontiers in neuroinformatics*, *4*, 113.

Hindmarsh, A. C., Brown, P. N., Grant, K. E., Lee, S. L., Serban, R., Shumaker, D. E., & Woodward, C. S. (2005). Sundials: Suite of nonlinear and differential/algebraic equation solvers. *ACM Transactions on Mathematical Software (TOMS)*, *31*(3), 363–396.

Hodgkin, A. L., & Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology*, *117*(4), 500.

Izhikevich, E. M. (2003a). Simple model of spiking neurons. *IEEE Transactions on neural networks*, *14*(6), 1569–1572.

Izhikevich, E. M. (2003b). Which model to use for cortical spiking neurons. *IEEE Transactions on Neural Networks*, *14*, 1569–1572.

Izhikevich, E. M. (2007). *Dynamical systems in neuroscience.* MIT press.

Jammalamadaka, R. (2003). Activity characterization of spatial models: Application to the discrete event solution of partial differential equations. *Master's thesis, University of Arizona, Tucson, Arizona, USA.*

Kaabi, M. G., Tonnelier, A., & Martinez, D. (2011). On the performance of voltage stepping for the simulation of adaptive, nonlinear integrate-and-fire neuronal networks. *Neural computation*, *23*(5), 1187–1204.

Kofman, E. (2002). A Second Order Approximation for DEVS Simulation of Continuous Systems. *Simulation: Transactions of the Society for Modeling and Simulation International*, *78*(2), 76–89.

Kofman, E. (2004a). Discrete Event Simulation of Hybrid Systems. *SIAM Journal on Scientific Computing*, *25*(5), 1771–1797.

Kofman, E. (2004b). Discrete event simulation of hybrid systems. *SIAM Journal on Scientific Computing*, *25*(5), 1771–1797.

Kofman, E. (2005). Non-conservative ultimate bound estimation in lti perturbed systems. *Automatica*, *41*(10), 1835–1838.

Kofman, E., Fernández, J., & Marzorati, D. (2021). Compact sparse symbolic jacobian computation in large systems of odes. *Applied Mathematics and Computation*, *403*, 126181.

Kofman, E., & Junco, S. (2001). Quantized State Systems. A DEVS Approach for Continuous System Simulation. *Transactions of SCS*, *18*(3), 123-132.

Lapicque, L. (1907). Recherches quantitatives sur l'excitation electrique des nerfs. *J. Physiol. Paris.*, *9*, 620–635.

Mascart, C., Scarella, G., Reynaud-Bouret, P., & Muzy, A. (2022). Scalability of large neural network simulations via activity tracking with time asynchrony and procedural connectivity. *Neural Computation*, *34*(9), 1915–1943.

Mattsson, S. E., Elmqvist, H., & Otter, M. (1998). Physical system modeling with modelica. *Control Engineering Practice*, *6*(4), 501–510.

Migoni, G., Bortolotto, M., Kofman, E., & Cellier, F. (2013). Linearly Implicit Quantization-Based Integration Methods for Stiff Ordinary Differential Equations. *Simulation Modelling Practice and Theory*, *35*, 118–136.

Morrison, A., & Diesmann, M. (2007). Maintaining causality in discrete time neuronal network simulations. In *Lectures in supercomputational neurosciences* (pp. 267–278). Springer.

Muzy, A., Jammalamadaka, R., Zeigler, B. P., & Nutaro, J. J. (2011). The activity-tracking paradigm in discrete-event modeling and simulation: The case of spatially continuous distributed systems. *Simulation*, *87*(5), 449–464.

Park, T., & Barton, P. I. (1996). State event location in differential-algebraic models. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, *6*(2), 137–165.

Petzold, L. R. (1982). *Description of dassl: a differential/algebraic system solver* (Tech. Rep.). Sandia National Labs., Livermore, CA (USA).

Potjans, T. C., & Diesmann, M. (2014). The cell-type specific cortical microcircuit: relating structure and activity in a full-scale spiking network model. *Cerebral cortex*, *24*(3), 785–806.

Quesnel, G., Duboz, R., Ramat, É., & Traoré, M. K. (2007). Vle: a multimodeling and simulation environment. In *Proceedings of the 2007 summer computer simulation conference* (pp. 367–374).

Richardson, M. J., Brunel, N., & Hakim, V. (2003). From subthreshold to firing-rate resonance. *Journal of neurophysiology*, *89*(5), 2538–2554.

Schmidt, M., Bakker, R., Shen, K., Bezgin, G., Diesmann, M., & van Albada, S. J. (2018). A multi-scale layer-resolved spiking network model of resting-state dynamics in macaque visual cortical areas. *PLOS computational biology*, *14*(10), e1006359.

Shelley, M. J., & Tao, L. (2001). Efficient and accurate time-stepping schemes for integrate-and-fire neuronal networks. *Journal of Computational Neuroscience*, *11*(2), 111–119.

Stein, R. B. (1967). Some models of neuronal variability. *Biophysical journal*, *7*(1), 37–68.

Vogels, T. P., & Abbott, L. F. (2005). Signal propagation and logic gating in networks of integrate-and-fire neurons. *Journal of neuroscience*, *25*(46), 10786–10795.

Zeigler, B. P., Kim, T. G., & Praehofer, H. (2000). *Theory of modeling and simulation*. Academic press.

Zeigler, B. P., Muzy, A., & Kofman, E. (2018). *Theory of modeling and simulation: Discrete event & iterative system computational foundations*. Academic Press.

Zheng, G., Tonnelier, A., & Martinez, D. (2009). Voltage-stepping schemes for the simulation of spiking neural networks. *Journal of computational neuroscience*, *26*(3), 409–423.

Appendix A. Proof of Theorems and Propositions

The following lemma, from (Kofman, 2005), will be used as an auxiliary result:

Lemma 1. *Consider the first order equation with complex coefficient*

$$\dot{x} = \lambda(x + \Delta x) + B\Delta u \tag{A1}$$

where λ, x, $\Delta x \in \mathbb{C}$, $\Delta u \in \mathbb{C}^k$ and $B \in \mathbb{C}^{1\times k}$. Assume also that $\mathbb{R}e(\lambda) < 0$, $|\Delta x| \leq \Delta x_{\max}$ and $|\Delta u| \leq \Delta u_{\max}$.

Let $x(t)$ be a solution of (A1) from the initial condition $x(t_0) = 0$. Then, for all $t \geq t_0$ it results that

$$|x(t)| \leq \left|\frac{\lambda}{\mathbb{R}e(\lambda)}\right| \Delta x_{\max} + \left|\frac{B}{\mathbb{R}e(\lambda)}\right| \Delta u_{\max}$$

A.1. Theorem 1

Proof. Defining $\Delta x_i \triangleq q_i - x_i$, Eq. (14) can be rewritten as a perturbed equation

$$\begin{aligned}\frac{\mathrm{d}x_1}{\mathrm{d}t}(t) &= -a(x_1(t) + \Delta x_1(t)) + b(x_2(t) + \Delta x_2(t)) + u_1(t)\\ \frac{\mathrm{d}x_2}{\mathrm{d}t}(t) &= -c(x_2(t) + \Delta x_2(t)) + u_2(t)\end{aligned} \tag{A2}$$

Defining also the error as $e_i \triangleq x_i - z_i$, and substracting Eq. (12) from Eq. (A2), the error dynamics results

$$\frac{\mathrm{d}e_1}{\mathrm{d}t}(t) = -a(e_1(t) + \Delta x_1(t)) + b(e_2(t) + \Delta x_2(t)) \tag{A3a}$$

$$\frac{\mathrm{d}e_2}{\mathrm{d}t}(t) = -c(e_2(t) + \Delta x_2(t)) \tag{A3b}$$

Notice that $e_{1,2}(t = t_0) = 0$ since we assume that the initial state is known and the initial error results equal to zero.

Then, we can apply Lemma 1 to the system of Eq (A3b) taking $\lambda = -c < 0$ and $B = 0$. $e_2(t_0) = x_2(t_0) - z_2(t_0) = 0$. Thus,

$$|e_2(t)| \leq \Delta Q_2, \tag{A4}$$

since $|\Delta x_2(t)| = |q_2(t) - x_2(t)| \leq \Delta Q_2$ from the relationship between the state, the quantized state and the quantum.

Then, defining $\tilde{e}_1 \triangleq e_1 + m \cdot e_2$, with

$$m = \begin{cases} 0 & \text{if } a \geq c \\ \dfrac{b}{c} & \text{if } a < c \end{cases} \tag{A5}$$

and differentiating w.r.t. the time, we obtain:

$$\begin{aligned}\frac{\mathrm{d}\tilde{e}_1}{\mathrm{d}t}(t) &= -a(e_1(t) + \Delta x_1(t)) + b(e_2(t) + \Delta x_2(t)) + m(-c)(e_2(t) + \Delta x_2(t))\\ &= -a(e_1(t) + \Delta x_1(t)) + (b - cm)(e_2(t) + \Delta x_2(t))\\ &= -a(\tilde{e}_1(t) - me_2(t) + \Delta x_1(t)) + (b - cm)(e_2(t) + \Delta x_2(t))\\ &= -a(\tilde{e}_1(t) + \Delta x_1(t)) + ame_2(t) + (b - cm)(e_2(t) + \Delta x_2(t))\end{aligned}$$

If $a \geq c \Rightarrow m = 0 \Rightarrow \tilde{e}_1(t) = e_1(t) \Rightarrow \dot{\tilde{e}}_1(t) = \dot{e}_1(t)$, then

$$\frac{\mathrm{d}e_1}{\mathrm{d}t}(t) = \frac{\mathrm{d}\tilde{e}_1}{\mathrm{d}t}(t) = -a(\tilde{e}_1(t) + \Delta x_1(t)) + b(e_2(t) + \Delta x_2(t)) \tag{A6}$$

Then, applying Lemma 1 to this system with $\lambda = -a$, $B = b$, $\Delta u(t) = e_2(t) + \Delta x_2(t)$, and taking into account that $|\Delta x_i(t)| = |q_i(t) - x_i(t)| \leq \Delta Q_i$ and considering also Eq. (A4), we obtain

$$|e_1(t)| \leq \Delta Q_1 + \left|\frac{b}{-a}\right| 2\Delta Q_2 = \Delta Q_1 + \frac{2b}{a}\Delta Q_2 \tag{A7}$$

On the other hand, if $a < c \Rightarrow m = \frac{b}{c} \Rightarrow \tilde{e}_1(t) = e_1(t) + \frac{b}{c} \cdot e_2(t)$ and

$$\begin{aligned} \frac{\mathrm{d}\tilde{e}_1}{\mathrm{d}t}(t) &= -a(e_1(t) + \Delta x_1(t)) + \frac{ab}{c} \cdot e_2(t) + (b - c\frac{b}{c})(e_2(t) + \Delta x_2(t)) \\ &= -a(\tilde{e}_1(t) + \Delta x_1(t)) + \frac{ab}{c} \cdot e_2(t) \end{aligned}$$

Thus, proceeding as before we obtain

$$|\tilde{e}_1(t)| \leq \Delta Q_1 + \left|\frac{ab}{(-a)c}\right| \Delta Q_2 = \Delta Q_1 + \frac{b}{c}\Delta Q_2 \tag{A8}$$

And then,

$$|e_1(t)| = \left|\tilde{e}_1(t) - \frac{b}{c} \cdot e_2(t)\right| \leq |\tilde{e}_1(t)| + \left|\frac{b}{c} \cdot e_2(t)\right| \leq \Delta Q_1 + \frac{2b}{c}\Delta Q_2 \tag{A9}$$

Eqs. (A7) and (A9) show that Eq. (15) holds, completing the proof. □

A.2. Theorem 2

Proof. Defining $\Delta x_1 \triangleq q_1 - x_1$, Eq. (18) can be rewritten as a perturbed equation

$$\frac{\mathrm{d}x_1}{\mathrm{d}t}(t) = -a(x_1(t) + \Delta x_1(t)) + v(t) \tag{A10}$$

Defining also the error as $e \triangleq x_1 - z_1$, and substracting Eq. (17) from Eq. (A10), the error dynamics results

$$\frac{\mathrm{d}e}{\mathrm{d}t}(t) = -a(e(t) + \Delta x_1(t)) + v(t) - u(t) \tag{A11}$$

Then, considering $e = e_1 + e_2$, we can split the derivative of the error as:

$$\frac{\mathrm{d}e_1}{\mathrm{d}t}(t) = -a(e_1(t) + \Delta x_1(t)) \tag{A12}$$

$$\frac{\mathrm{d}e_2}{\mathrm{d}t}(t) = -ae_2(t) + v(t) - u(t) \tag{A13}$$

Applying Lemma 1 to the system of Eq. (A12), with $\lambda = -a$ and $B = 0$, we obtain $|e_1(t)| \leq \Delta Q_1$

Then, defining $\Delta t = (t - t_k)$ and Δt_{max} the largest Δt, we can assure from the hypothesis that $|v(t) - u(t)| < \gamma \Delta t^n$.

Applying Lemma 1 to the system of Eq. (A13), with $\lambda = -a$, $\Delta x = 0$, $B = 1$, and $\Delta u = v(t) - u(t)$ we obtain $|e_2(t)| \leq \left|\frac{1}{-a}\right| \gamma \Delta t^n_{\max} = \frac{\gamma}{a} \Delta t^n_{\max}$

Then, $|e(t)| \leq |e_1(t)| + |e_2(t)| \leq \Delta Q_1 + \dfrac{\gamma}{a} \cdot \Delta t^n_{max}$ completing the proof.

□

A.3. Theorem 3

Proof. The analytical solution of z_1 is given by $z_1(t) = e^{-at} v_1 + v_2$ with

$$v_1 = V_r - \frac{b\bar{z}_2 + u_1}{a}$$

and

$$v_2 = \frac{b\bar{z}_2 + u_1}{a} \tag{A14}$$

Since at time t_θ the state z_1 reaches the threshold, we have $z_1(t_\theta) = e^{-at_\theta} v_1 + v_2 = \theta$, and then

$$t_\theta = \frac{1}{a} \log(\frac{v_1}{\theta - v_2})$$

In order to verify that the argument of the logarithm is positive, notice that, from Eqs.(21) and (A14) it results

$$v_2 = \frac{b\bar{z}_2 + u_1}{a} > \frac{b\dfrac{a\theta - u_1}{b} + u_1}{a} = \theta$$

and $v_1 = V_r - v_2 < V_r - \theta < 0$.

Then, recalling that error bound between $z_1(t)$ and $x_1(t)$ according to Eq. (15) in Theorem 1 is given by Δ_1, we can assure that the numerical solution $x_1(t)$ will be able to reach the threshold only after the analytical solution satisfies $z_1(t) \geq \theta - \Delta_1$, so the minimum time at which $x_1(t)$ can reach the threshold is given by:

$$\tilde{t}_\theta^{\min} = \frac{1}{a} \log(\frac{v_1}{(\theta - \Delta_1) - v_2})$$

So, the difference between the firing time of the analytical solution and the minimum possible firing time of the numerical solution can be computed as

$$\begin{aligned} t_\theta - \tilde{t}_\theta^{\min} =& \frac{1}{a}\left(\log(\frac{v_1}{\theta - v_2}) - \log(\frac{v_1}{(\theta-\Delta_1) - v_2})\right) \\ =& \frac{1}{a}\log\left(\frac{v_1}{\theta - v_2}\cdot\frac{(\theta-\Delta_1)-v_2}{v_1}\right) \\ =& \frac{1}{a}\log\left(1 - \frac{\Delta_1}{\theta - v_2}\right) \\ =& \frac{1}{a}\log\left(1 - \frac{\Delta Q_1 + \dfrac{2b}{\max(a,c)}\Delta Q_2}{\theta - \dfrac{b\bar{x}_2 + u_1}{a}}\right) \end{aligned}$$

An identical analysis shows that the numerical solution $x_1(t)$ will not be able to reach the threshold after the analytical solution satisfies $z_1(t) \geq \theta + \Delta_1$. Thus, the maximum time in which $x_1(t)$ could reach the threshold is given by

$$\tilde{t}_\theta^{\max} = \frac{1}{a}\log(\frac{v_1}{(\theta+\Delta_1) - v_2}).$$

Then, the difference between the firing time of the analytical solution and the maximum firing time of the numerical solution is

$$|\tilde{t}_\theta^{\max} - t_\theta| = \left|\frac{1}{a}\log\left(1 + \frac{\Delta_1}{\theta - v_2}\right)\right|. \tag{A15}$$

Notice that, since $\dfrac{\Delta_1}{\theta - v_2} < 0$ it results

$$|t_\theta - \tilde{t}_\theta^{\min}| = \left|\frac{1}{a}\log\left(1 - \frac{\Delta_1}{\theta - v_2}\right)\right| < \left|\frac{1}{a}\log\left(1 + \frac{\Delta_1}{\theta - v_2}\right)\right| = |t_\theta - \tilde{t}_\theta^{\max}|$$

Then, $|t_\theta - \tilde{t}_\theta| < |\tilde{t}_\theta^{\max} - t_\theta|$, completing the proof after replacing in Eq. (A15) with the corresponding expressions for Δ_1 and v_2 .

□

A.4. Proposition 1

Proof. Since $V_r \leq z_1(t) \leq \theta$, then, from Eq. (12)

$$-aV_r + b\bar{z}_2 + u_1 \leq \dot{z}_1(t) \leq -a\theta + b\bar{z}_2 + u_1 = -aV_r + b\bar{z}_2 + u_1 + aV_r - a\theta$$

Notice that $|b\bar{z}_2 + u_1| >> a(\theta - V_r)$ implies $b\bar{z}_2 + u_1 + a(V_r - \theta) \approx b\bar{z}_2 + u_1$, and,

$$-aV_r + b\bar{z}_2 + u_1 \leq \dot{z}_1(t) \leq -aV_r + b\bar{z}_2 + u_1 + a(V_r - \theta) \approx -aV_r + b\bar{z}_2 + u_1$$

implying that $\dot{z}_1(t)$ is almost constant, *i.e.*, $\dot{z}_1 \approx d_1 \triangleq -aV_r + b\bar{z}_2 + u_1$.

That way, the firing period of z_1 is approximately

$$T \approx \frac{\theta - V_r}{d_1} \tag{A16}$$

and taking into account that the error between x_1 and z_1 is bounded by Δ_1, the maximum error in the period is given by

$$|\Delta T| \leq \Delta T_{\max} \approx \frac{\Delta_1}{d_1} \tag{A17}$$

and dividing Eq. (A17) with Eq. (A16) we obtain the bound of Eq. (25) completing the proof. □

Appendix B. Statistical Error Analysis

In order to verify that the errors reported in Tables 2-3 do not change significantly with the pseudo-random sequence that generates the input spike train, we repeated all the simulations using 30 different seeds and we measured the different errors in each case. Then, we computed the mean value and the standard deviation of each reported error.

Table B1 shows the results corresponding to the synaptic current I_s in the two-state model, corroborating that the errors on that state are very similar for all the simulations (the standard deviation is from one to two orders of magnitude below the mean error value).

Method	Tol	I_s [nA]			
	[nA,mV]	$\mu\ e_{\text{mean}}$	$\sigma\ e_{\text{mean}}$	$\mu\ e_{\max}$	$\sigma\ e_{\max}$
QSS	1E-1	2.15E-2	1.10E-4	5.00E-2	2.24E-5
	1E-2	2.50E-3	1.37E-5	5.90E-3	6.27E-4
	1E-3	2.50E-4	1.33E-6	6.00E-4	7.28E-5
	1E-4	2.51E-5	1.55E-7	6.77E-5	6.42E-6
QSS2	1E-1	7.63E-3	1.34E-4	4.99E-2	7.14E-5
	1E-2	3.54E-3	2.87E-5	1.38E-2	3.66E-4
	1E-3	2.66E-4	2.54E-6	1.27E-3	8.45E-6
	1E-4	2.53E-5	1.68E-7	1.27E-4	3.25E-6
QSS3	1E-1	1.18E-2	1.55E-4	4.99E-2	7.81E-5
	1E-2	3.68E-3	1.92E-5	1.21E-2	5.02E-4
	1E-3	2.79E-4	3.04E-6	1.11E-3	2.51E-5
	1E-4	2.47E-5	1.89E-7	1.14E-4	1.18E-6

Table B1. Mean value and standard deviation of the measured errors in the synaptic current I_s (two-state model) over 30 simulations using different pseudo-random sequences.

Table B2 reports the results regarding the membrane potential, also for the two-state model. There, all the sub-threshold errors have also a very small deviation. In the case of the full trajectory of the membrane potential, the mean errors are still similar to the quantum, but there is more variation due to some errors in the number of emitted spikes for the largest quantum.

Table B3 repeats the results of Table B2 for the single-state model. The conclusions are also very similar, except that this time the errors in the number of emitted spikes

Method	Tol	V_{sub} [nA]				V [nA]		Out Spikes
	[nA,mV]	$\mu\ e_{\text{mean}}$	$\sigma\ e_{\text{mean}}$	$\mu\ e_{\text{max}}$	$\sigma\ e_{\text{max}}$	$\mu\ e_{\text{mean}}$	$\sigma\ e_{\text{mean}}$	$\mu\ e_{\text{mean}}$
QSS	1E-1	3.44E-2	3.54E-3	1.41E-1	1.86E-2	1.49E-1	1.46E-1	0.26667
	1E-2	4.00E-3	2.96E-4	1.54E-2	1.14E-3	4.02E-2	9.99E-2	0.066667
	1E-3	4.16E-4	3.84E-5	1.57E-3	9.84E-5	6.30E-3	3.10E-2	0.033333
	1E-4	4.24E-5	2.55E-6	1.59E-4	1.17E-5	5.05E-5	6.05E-6	0
QSS2	1E-1	3.54E-2	2.36E-3	1.11E-1	9.68E-3	2.51E-1	2.46E-1	0.53333
	1E-2	4.65E-3	3.22E-4	1.54E-2	1.15E-3	2.07E-2	5.81E-2	0
	1E-3	2.89E-4	1.23E-5	1.19E-3	2.02E-4	3.49E-4	3.49E-4	0
	1E-4	2.91E-5	1.12E-6	6.86E-5	1.90E-5	2.80E-5	2.38E-6	0
QSS3	1E-1	2.85E-2	1.63E-3	1.10E-1	2.07E-2	6.86E-2	9.21E-2	0.033333
	1E-2	3.67E-3	2.07E-4	1.29E-2	1.44E-3	8.24E-3	1.44E-2	0
	1E-3	2.81E-4	2.03E-5	1.10E-3	7.81E-5	3.59E-4	2.90E-4	0
	1E-4	2.52E-5	1.57E-6	1.13E-4	5.89E-6	2.69E-5	2.32E-6	0

Table B2. Mean value and standard deviation of the measured errors in the membrane potential V (two-state model) over 30 simulations using different pseudo-random sequences.

are more frequent.

Method	Tol	V_{sub} [nA]				V [nA]		Out Spikes
	[nA,mV]	$\mu\ e_{\text{mean}}$	$\sigma\ e_{\text{mean}}$	$\mu\ e_{\text{max}}$	$\sigma\ e_{\text{max}}$	$\mu\ e_{\text{mean}}$	$\sigma\ e_{\text{mean}}$	$\mu\ e_{\text{mean}}$
QSS	1E-1	8.18E-1	1.47E-1	2.07E+0	2.38E+0	4.19E+0	2.52E-1	20.7333
	1E-2	1.89E-1	3.71E-2	9.61E-1	2.59E+0	2.53E+0	2.59E-1	7.5333
	1E-3	4.01E-2	1.59E-2	1.25E-1	6.15E-2	8.15E-1	2.97E-1	2.1
	1E-4	1.01E-2	1.20E-2	3.58E-2	3.67E-2	2.40E-1	2.40E-1	0.4
QSS2	1E-1	2.38E-1	3.66E-2	4.51E-1	5.67E-2	1.09E+0	3.09E-1	2.5
	1E-2	4.51E-2	4.83E-3	7.90E-2	8.24E-3	2.72E-1	2.10E-1	0.5
	1E-3	3.40E-3	3.54E-5	5.89E-3	2.89E-4	2.66E-2	5.23E-2	0.1
	1E-4	3.52E-4	2.40E-6	5.88E-4	1.42E-5	8.96E-4	5.73E-4	0
QSS3	1E-1	4.38E-2	2.05E-2	1.23E-1	4.53E-2	3.51E-1	2.89E-1	0.7
	1E-2	1.48E-2	2.16E-3	2.95E-2	5.48E-3	1.03E-1	1.27E-1	0.1
	1E-3	2.94E-3	2.13E-4	4.94E-3	3.92E-4	1.16E-2	2.20E-2	0
	1E-4	3.31E-4	7.61E-6	5.23E-4	1.35E-5	6.96E-4	3.66E-4	0

Table B3. Mean value and standard deviation of the measured errors in the membrane potential V (single-state model) over 30 simulations using different pseudo-random sequences.