Activity of Order n in Continuous Systems

Rodrigo Castro[1,3] and Ernesto Kofman[2,3]

[1]Computer Science Department, University of Buenos Aires, Buenos Aires, Argentina.
[2]French-Argentine International Center for Information and Systems Sciences (CIFASIS), Rosario, Argentina.
[3]National Scientific and Technical Research Council (CONICET), Argentina.

Abstract

In this work we generalize the concept of activity of continuous time signals. We define the activity of order n of a signal and show that it allows to estimate the number of sections of polynomials up to order n which are needed to represent that signal with certain accuracy. Then, we apply this concept to obtain a lower bound for the number of steps performed by quantization–based integration algorithms in the simulation of ordinary differential equations.

We perform an exhaustive analysis over two examples, computing the activity of order n and comparing it with the number of steps performed by different integration methods. This analysis corroborates the theoretical predictions and also allows to measure the suitability of the different algorithms depending on how close they perform in comparison with the theoretical lower bound.

1 Introduction

The concept of activity associated to a continuous signal (in the following *continuous activity*) was introduced by Zeigler and Jammalamadaka [6] in order to measure the rate of change of a signal.

More precisely, in its original definition the continuous activity computes the total change experienced by a signal within a given interval of time. Thus, for a monotonically increasing or decreasing signal, the activity measures the distance between the final and the initial value. When a signal is not monotonic, the activity is computed as the sum of the activities of the monotonic sections.

In either case, the formal definition of the activity for a signal $x_i(t)$ between an initial time t_0 and a final time t_f is given by

$$A_{x_i(t_0,t_f)} \triangleq \int_{t_0}^{t_f} \left| \frac{dx_i(\tau)}{d\tau} \right| \cdot d\tau \tag{1}$$

According to this formulation, the activity only measures distances between final and initial values, without using at all the information about *how* the signal reaches those values. Thus, a monotonic signal that grows (or decreases) with a straight ramp presents the same activity than a monotonic signal that starts and ends at the same values but following a more complex function of time.

When a continuous time signal $x_i(t)$ is the input of a *zero–order* quantization function, the corresponding output trajectory $q_i(t)$ results piecewise constant as shown in Figure 1.

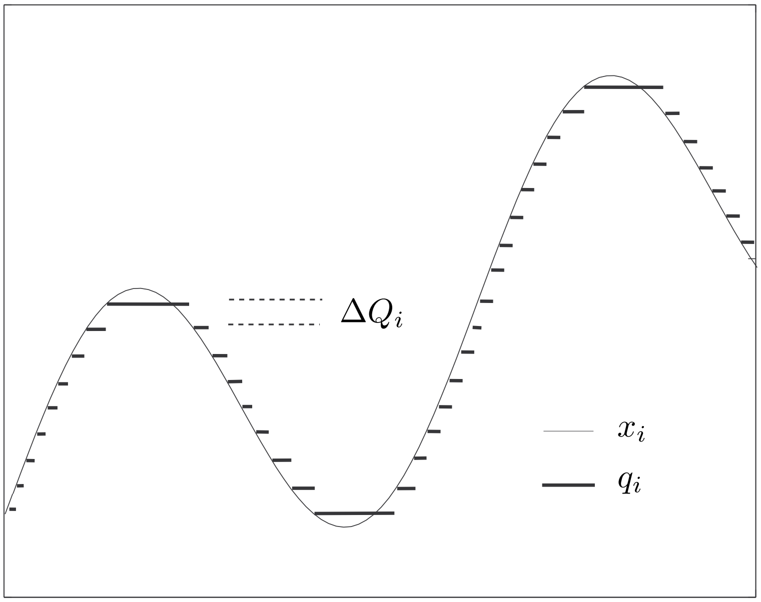

Figure 1: Zero order quantization with quantum ΔQ_i.

In this case, the number of discontinuities of the output trajectory $q_i(t)$ is closely related to the activity of the input $x_i(t)$.

If the amplitude (i.e., the activity) of the j–th monotonic section of $x_i(t)$ is A_j and the quantum size of the quantization function is ΔQ_i, then the number of quantum crossings of $x_i(t)$ in that section is about $A_j/\Delta Q_i$.

Thus, the total number of discontinuities in $q_i(t)$ can be directly computed as

$$k \approx \sum_j \frac{A_j}{\Delta Q_i} = \frac{A_{x_i(t_0,t_f)}}{\Delta Q_i} \tag{2}$$

Zero–order quantization functions like that of Fig.1 are used in some Quantized State Systems (QSS) numerical integration algorithms, such as the first order accurate QSS1 method [11], the Backward QSS (BQSS) method [15], and the first order accurate Linearly Implicit QSS (LIQSS1) algorithm [14].

Thus, Eqs.(1)–(2) can be used to establish a lower bound for the number of steps needed by those methods to simulate a given system with a given quantum. That way, the concept of activity can predict the minimal computational costs required to simulate a system with a given accuracy.

However, there exist higher order quantization functions that are also used in QSS methods of higher order. For instance, the QSS2 method [9] uses *first order* quantization functions that produce piecewise linear output trajectories as shown in Fig.2.

In these cases, nor Eq.(1) neither Eq.(2) provide any help to estimate the number of discontinuities in the output trajectory.

Firstly, as it was already mentioned, the activity definition of Eq.(1) obtains the same result for a straight ramp than for a more complex (monotonic) signal, provided that both signals have the same amplitude. However, a straight ramp can be represented by a single section of a piecewise linear trajectory while a more complex signal, depending on the quantum, would require more segments. Evidently, this difference is not captured by Eq.(1)

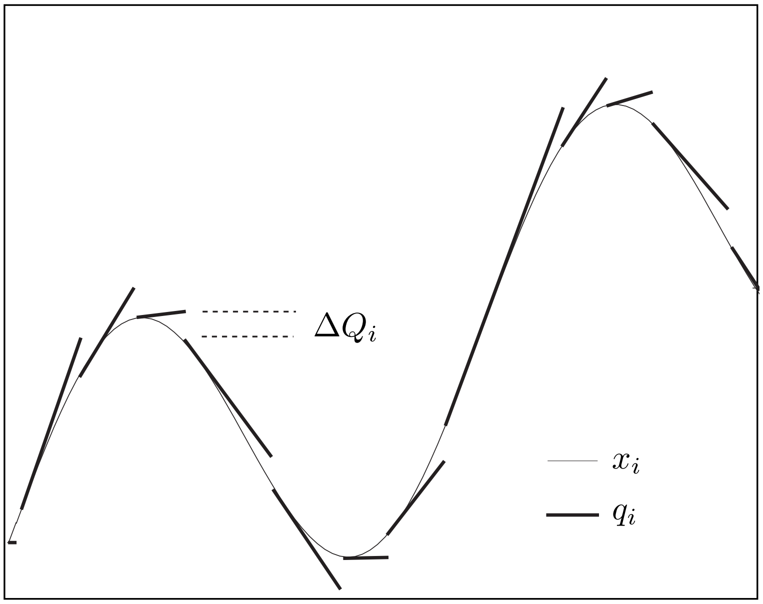

Figure 2: First order quantization.

Secondly, it is known that in second order accurate methods like QSS2 the number of steps varies with the square root of the quantum ΔQ_i [9, 14]. However, Equation (2) shows a linear dependence, which is clearly wrong.

These facts motivated the need of generalizing the concept of activity so that it can be still applied in presence of higher order quantization.

In this work we study and develop the idea of *activity of order n* of a signal as a property that allows estimating the number of sections of polynomials up to order n that are needed to represent that signal with a given accuracy. Then, we apply this concept to obtain a lower bound for the number of steps performed by quantization–based integration algorithms in the simulation of ordinary differential equations.

The paper is organized as follows. In Section 2 we begin with a historical perspective of activity–based modeling and simulation establishing the relation between discrete and continuous activity. Then in Section 3 we review quantization schemes up to order 3, followed by Section 4 discussing how they are used in quantization–based simulation of continuous systems.

In Section 5 we first derive the expression for n–th order quantization and, based on that, we present the new definition for continuous activity of order n. Then, in Section 6 we apply the new definitions in two practical example models: a first order non-stiff system, and a second order stiff system. In both cases we analyze the correlation between theoretical and practical results obtained through simulation.

Finally, in Section 7 we present the conclusions and provide hints about follow up steps stemming from the concepts introduced in this work.

2 Historical perspective on activity–based modeling and simulation

During the 60's (Lackner [12, 13], Kiviat [7, 8]) and the beginning of the 70's (Fishman [4]) discrete event simulation strategies started being categorized according to so-called world views ("Weltansicht"). World views were originally meant to provide conceptual frameworks that can guide systematically the development of discrete event simulation languages and simulation software. A subsequent more practical reference study is that of Balci ([1]) in the late 80's.

The "classical" world views are: event-scheduling, activity scanning and process-oriented. The adoption of one or several of these can be made according to the needs of the particular goal at hand (e.g. creating a simulation language, a software tool or a single model).

In a nutshell, these world views can be synthesized as follows:

- Event-scheduling: Model dynamics are driven by events scheduled ahead of time, on a continuous time base, according to local rules whose logic cannot be subjected to global state information.

- Activity scanning: Model dynamics are driven by activities, which are phases commenced and ended by events. Such events make a model's phase "active" or "inactive". The occurrence of events can only take place on a clock-driven discrete-time base. At each time slot, models are "scanned" to check whether conditions are satisfied for the triggering events.

- Process interaction: Model dynamics are driven by processes, which can be combinations (possibly complex) of events and/or activities (i.e., of the event scheduling and the activity scanning paradigms).

An updated and more in-depth description of world views can be found in [2].

Recently, in [18] an integrative approach was introduced aiming at combining modeling views and time flow management under a single strategy termed *activity tracking*. The activity tracking pattern merges the three classical world views. Time flow is continuous and dynamics are driven by events, just like in event scheduling. At the same time, active and inactive phases can be handled just like in activity scanning. Nonetheless the latter is achieved by an asynchronous tracking mechanism (handling "marks" that are propagated throughout a hierarchy of models), instead of the classical synchronous scanning mechanism (which queries all models only at permitted time slots).

In [18] it was proposed that the DEVS formalism [23] is a sound candidate for expressing and implementing the activity tracking strategy. The authors proposed activity tracking as a comprehensive world view for modeling and simulation that can improve efficiency and rigorousness.

Regardless of the world view of choice, in the context of discrete event simulations, dynamic systems get ultimately driven by trajectories of state changes on a continuous time base. As time evolves, the occurrence of events dictates state changes at given timestamps. The latter can be counted and then interpreted quantitatively as a measure of the "activity" of the system. Within the *activity tracking* realm, the word *activity* is linked to this quantitative *counting* of a discrete nature, or *discrete activity.*

On the contrary, the *continuous activity* (as introduced in [6] and discussed in the previous section) is a quantitative measure of a continuous nature.

Continuous activity can serve as a formal link between inherent characteristics of a continuous signal and the minimum discrete activity theoretically required to approximate said signal with a desired accuracy, using a quantized–state simulation method in a discrete event setup [22]. This idea applies e.g. to the particular case where the quantized–state simulation method is the QSS methods commented before.

Such formal link can enable establishing a theoretical connection between the analytical expression of a continuous trajectory and the computational effort required to simulate it. An obvious parameter that can serve as said link is the quantum size adopted for the quantization–based approximation.

In the context of the historical evolution of world views, continuous activity is relevant as it can provide quantitative links between the emerging activity tracking pattern and the domain of continuous systems simulation.

Nevertheless, for achieving true generality, a formal foundation must be provided that considers generalized quantization–based numerical techniques of arbitrary order of approximation. Such general formal foundation is the main result presented in this work.

2.1 State of the art in the field

Several works have investigated recently the advantages of applying activity–driven techniques in discrete event simulation. The reader is referred to the references therein for a broader perspective.

In [16] the activity scanning strategy is analyzed in context of other possible world views and provides new definitions (e.g. qualitative and quantitative activity) and essays an overarching multi-level life cycle adapted to activity aware simulations.

In [17] the authors applied the activity tracking paradigm to a one-dimensional Partial Differential Equations (PDEs) solved numerically using state-quantization instead of time-slicing methods. They showed evidence on how to use discrete events as a means to track activity in a simple spatial system, using a diffusion model with known analytical solution for accuracy comparison purposes. They also emphasize that activity of systems can be "tracked" (basically, dealt with) at both modeling and simulation phases.

A recent work in [5] introduces an activity-based framework that links information and energy, and applies it to support energy-aware information processing in wireless sensor nodes that detect and monitor wildfires.

Also recently in [20] a detailed analysis is provided for a practical implementation of the activity tracking paradigm in the context of the object-oriented DEVSimPy simulation framework.

In [19] the authors explore the activity concept in varied modeling domains, and stemming from them identify a general three level architecture for guiding the construction of component based systems.

3 Signal quantization schemes

In this section we review a particular family of quantization functions which are used in the context of quantization–based integration of Ordinary Differential Equations (ODEs).

These quantization functions approximate a continuous time input signal $x_i(t)$ by a piecewise polynomial output signal $q_i(t)$, so that they do not differ more than the quantum ΔQ_i. This is, they ensure that.

$$|x_i(t) - q_i(t)| \leq \Delta Q_i \tag{3}$$

3.1 Zero–order quantization

In zero–order quantization [11, 3] the approximating polynomial segments are of order zero, i.e., the quantized signal $q_i(t)$ is piecewise constant.

Formally, given an input signal $x_i(t)$ and a piecewise constant output signal $q_i(t)$, we say that they are related by a *zero–order quantization function* Q_0 with quantum ΔQ_i if they satisfy

$$q_i(t) = q_i(t_j) \text{ for } t_j \le t < t_{j+1} \tag{4a}$$

with the sequence t_j defined by

$$t_{j+1} = \min_{t>t_j} t \text{ subject to } |q_i(t_j) - x_i(t)| = \Delta Q_i \tag{4b}$$

Notice that $q_i(t)$ follows a piecewise constant trajectory that only changes its value when the difference between $q_i(t)$ and $x_i(t)$ becomes equal to the quantum. After each recalculation of the quantized variable it results that $q_i(t) = x_i(t)$. This behavior is depicted in Figure 1.

One consequence of this approach is that a regular grid of evenly spaced *quantization thresholds* can be imagined superimposed to the input and output trajectories offering an intuitive visual perception of the quantization process: new values of $q_i(t)$ are produced as $x_i(t)$ hits the thresholds that verify $|q_i(t_j) - x_i(t)| = \Delta Q_i$.

Unfortunately, as we shall see shortly after, this grid–oriented hint is only possible in the zero order case; such an evenly spaced set of adjacent thresholds will lack any meaning in higher order schemes, starting already with the first–order quantization case.

3.2 First and second order quantization

The same idea presented for zero–order quantization is followed in first–order quantization [9], but this time around resorting to piecewise *linear* segments for constructing $q_i(t)$ rather than piecewise constant as in the preceding case.

Formally, given an input signal $x_i(t)$ and a piecewise linear output signal $q_i(t)$, we say that they are related by a *first–order quantization function* with quantum ΔQ_i if they satisfy

$$q_i(t) = q_i(t_j) + c_{1,j} \cdot (t - t_j) \text{ for } t_j \le t < t_{j+1} \tag{5a}$$

with the sequence t_j defined by

$$t_{j+1} = \min_{t>t_j} t \text{ subject to } |q_i(t_j) + c_{1,j} \cdot (t - t_j) - x_i(t)| = \Delta Q_i \tag{5b}$$

and $c_{1,j}$ computed as:

$$c_{1,j} = \frac{\mathrm{d}x_i}{\mathrm{d}t}(t_j) \tag{5c}$$

The result of this approach is that $q_i(t)$ follows a piecewise linear trajectory that experiences discontinuities at time instants $t = t_j$ when the difference between $q_i(t_j)$ and $x_i(t_j)$ is equal to the quantum ΔQ_i, as shown in Fig.2.

Along the same lines, given an input signal $x_i(t)$ and a piecewise parabolic output signal $q_i(t)$, we say that they are related by a *second–order quantization function* [10] with quantum ΔQ_i if they satisfy

$$q_i(t) = q_i(t_j) + c_{1,j} \cdot (t - t_j) + c_{2,j} \cdot (t - t_j)^2 \text{ for } t_j \le t < t_{j+1} \tag{6a}$$

with the sequence t_j defined by

$$t_{j+1} = \min_{t>t_j} t \text{ subject to } |q_i(t_j) + c_{1,j} \cdot (t - t_j) + c_{2,j} \cdot (t - t_j)^2 - x_i(t)| = \Delta Q_i \tag{6b}$$

and $c_{1,j}$, $c_{2,j}$ computed as:

$$c_{1,j} = \frac{\mathrm{d}x_i}{\mathrm{d}t}(t_j); \quad c_{2,j} = \frac{1}{2!}\frac{\mathrm{d}^2x_i}{\mathrm{d}t^2}(t_j) \tag{6c}$$

which results in $q_i(t)$ following a piecewise parabolic trajectory, changing their polynomial coefficients only at time instants $t = t_j$ when the difference between q_i and x_i becomes equal to the quantum ΔQ_i.

The behavior of a second order quantization function is depicted in Figure 3.

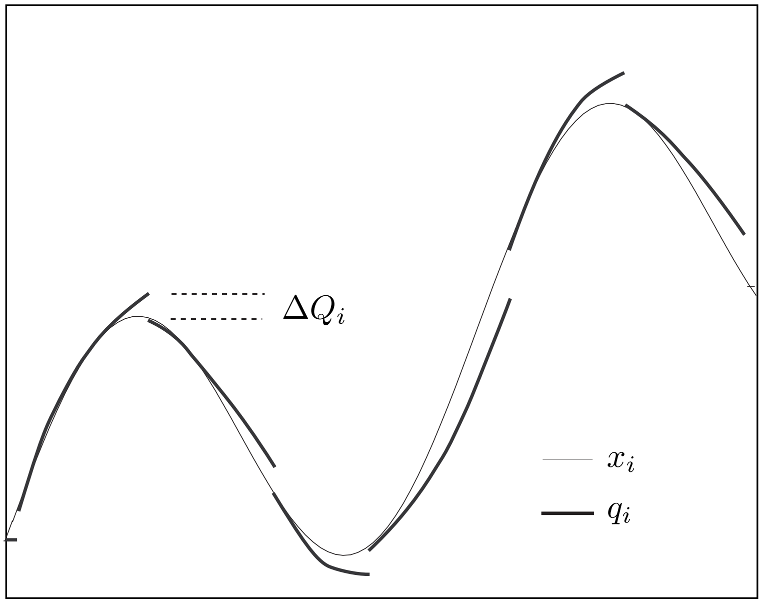

Figure 3: Second order quantization.

4 Quantization–Based Integration

Continuous Time Systems are typically represented by ODEs. Except for very simple cases, these ODEs lack of analytical solutions and they must be approximated by numerical integration algorithms in order to be solved. Classic numerical integration algorithms are based on the discretization of the time variable [3].

In recent years, a new class of ordinary differential equation solvers has been developed that replaces the time discretization by the state quantization [3]. These algorithms, based on Zeigler's idea of representing quantized system as DEVS models [21, 23], are called *Quantized State Systems* (QSS) methods.

A QSS numerical solver operates naturally in an asynchronous mode, i.e., the instants t_j belong to the set of positive real numbers and are not confined to any synchronized pattern of time instants.

Each state variable carries its own simulation clock. If the states of a subsystem change very little, the model equations capturing the dynamics of that subsystem will be executed rarely (or equivalently, their activity will be very low).

In the context of QSS–based simulations a dormant model does not slow down the simulation, as its equations will not get executed (i.e., it will experience null activity).

The quantization schemes presented above are those employed within QSS methods that shall be described in the next section.

4.1 First Order QSS1 Method

Given the system:

$$\dot{\mathbf{x}}_{\mathbf{a}}(t) = \mathbf{f}(\mathbf{x}_{\mathbf{a}}(t), t) \tag{7}$$

with analytical solution $\mathbf{x}_{\mathbf{a}}(t)$, the first order QSS1 method approximates it by

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{q}(t), t) \tag{8}$$

Here, $\mathbf{q}$ is the *quantized state vector*. Its entries $q_i(t)$ are componentwise related with those of the state vector $x_i(t)$ by *hysteretic quantization functions*, defined as in Eq.(4).

4.2 Other First Order QSS Methods

Besides QSS1, the Backward QSS (BQSS) [15], Centered QSS (CQSS) [15] and Linearly Implicit QSS (LIQSS1) [14] perform first order approximations. They differ from QSS1 in the definition of the quantization function given by Eq.(4), however in all these methods the quantized variables $q_i(t)$ follow piecewise constant trajectories.

BQSS and LIQSS1 were conceived to efficiently simulate stiff systems, while CQSS was proposed to simulate marginally stable systems.

4.3 Higher Order QSS Methods

The accuracy of the simulation is directly related to the quantum ΔQ_i [9]. Thus, if we want to improve the accuracy by a factor of e.g. 100, the quantum must be reduced 100 times. Then, *any* first order QSS method will perform 100 times more steps. This is a serious limitation of first order schemes, since accurate results require performing lots of steps with the corresponding increment in the computational costs.

To overcome this difficulty, higher order methods where developed like the second order QSS (QSS2) [9] and the third order QSS (QSS3) [10].

The QSS2 method is based on the same principles as QSS1, approximating Eq.(7) by Eq.(8). However, it replaces the zero–order quantization function of Eq.(4) and Figure 1 by a first-order quantization function of Eq.(5) and Figure 2.

Consequently, the quantized state trajectories $q_i(t)$ are piecewise linear and each segment starts with a value and slope equal to those of the corresponding state $x_i(t)$. When both trajectories differ by ΔQ_j, a new segment of $q_i(t)$ starts.

It was shown that in QSS2, the number of steps grows with the square root of the accuracy. Thus, if we want to improve the accuracy by a factor of 100, QSS2 performs only 10 times more steps.

The third order QSS3 method is identical to QSS2, except that it replaces the first order quantization function by the second order quantization function of Eq.(6) and Figure 3.

A second order quantization function generates an output piecewise parabolic trajectory , whose value, slope and second slope change when the difference between the output and input of the function becomes bigger than the quantum. Each output segment starts with the same value, slope and quadratic slope than the input.

It was shown that in QSS3, the number of steps grows with the cubic root of the accuracy. Thus, if we want to improve the accuracy by a factor of 1000, QSS3 performs only 10 times more steps.

Besides QSS2 and QSS3, there exist *linearly implicit* QSS methods of orders 2 (LIQSS2) and 3 (LIQSS3) which are particularly suitable to simulate stiff systems [14], while having definitions very similar to those of QSS2 and QSS3.

5 Activity of Order n

In this section we generalize the concept of activity so that it can be applied in the context of higher order quantization functions like those used in QSS2 and QSS3 methods.

Before introducing the new definition of activity of order n, we present a definition of n–th order quantization.

5.1 n–th order quantization

The ideas behind first and second order quantization functions presented in Section 3.2 can be generalized to define the n–th order quantization function.

Given an input signal $x_i(t)$ and a piecewise polynomial output signal $q_i(t)$, we say that they are related by a *n–th–order quantization function* with quantum ΔQ_i if they satisfy

$$q_i(t) = q_i(t_j) + c_{1,j} \cdot (t - t_j) + c_{2,j} \cdot (t - t_j)^2 + \cdots + c_{n,t_j}.(t - t_j)^n \text{ for } t_j \le t < t_{j+1} \tag{9a}$$

with the sequence t_j defined by

$$t_{j+1} = \min_{t>t_j} t \text{ subject to } |q_i(t_j)+c_{1,j}\cdot(t-t_j)+c_{2,j}\cdot(t-t_j)^2+\cdots+c_{n,t_j}.(t-t_j)^n-x_i(t)| = \Delta Q_i \tag{9b}$$

and the coefficients $c_{m,j}$ computed as:

$$c_{m,j} = \frac{1}{m!}\frac{\mathrm{d}^m x_i}{\mathrm{d}t^m}(t_j) \tag{9c}$$

The latter requires all derivatives of $x_i(t)$ to exist at least up the order of the quantization scheme, i.e., n.

5.2 Analytical derivation of Activity of Order n

In this section we develop an analytical expression for activity of order n.

The original definition of activity [6] given by Eq.(1) integrates the rate of change $\left|\frac{\mathrm{d}x_i(t)}{\mathrm{d}t}\right|$ experienced by a continuous time signal $x_i(t)$ in a given interval of time.

When $q_i(t)$ is a piecewise constant approximation of $x_i(t)$ (i.e., the result of a quantization function of order zero), the rate of change $\left|\frac{\mathrm{d}x_i(t)}{\mathrm{d}t}\right|$ coincides with the rate at which the difference $\Delta x_i = |q_i(t)-x_i(t)|$ grows (while $q_i(t)$ remains constant). Consequently, as soon as a quantum ΔQ_i is chosen, the number of constant sections required to approximate the signal gets immediately determined by the division of the activity by the quantum size. We refer to this activity as *activity of order zero*.

However, if $q_i(t)$ is obtained from a quantization function of order $n-1$ with $n \ge 2$, the rate at which the difference $|q_i(t) - x_i(t)|$ grows follows a different law:

$$\Delta x_i(t) = x_i(t) - q_i(t) = x_i(t) - \left[x_i(t_j) + \frac{\mathrm{d}x_i(t_j)}{\mathrm{d}t} \cdot (t - t_j) + \cdots + \frac{\mathrm{d}^{n-1}x_i(t_j)}{\mathrm{d}t^{n-1}} \cdot \frac{(t-t_j)^{n-1}}{(n-1)!}\right] \tag{10}$$

where t_j is the time of the last discontinuity of $q_i(t)$.

Evidently, for the same given accuracy, the number of sections of polynomials up to order $n-1$ required to approximate the signal requires a reformulation.

We proceed as follows. Replacing $x_i(t)$ in Eq.(10) by its Taylor series expansion:

$$x_i(t) = x_i(t_j) + \frac{\mathrm{d}x_i(t_j)}{\mathrm{d}t} \cdot (t - t_j) + \cdots + \frac{\mathrm{d}^n x_i(t_j)}{\mathrm{d}t^n} \cdot \frac{(t - t_j)^n}{n!} + \cdots$$

it results that

$$\Delta x_i(t) = \frac{\mathrm{d}^n x_i(t_j)}{\mathrm{d}t^n} \cdot \frac{(t - t_j)^n}{n!} + \frac{\mathrm{d}^{n+1} x_i(t_j)}{\mathrm{d}t^{n+1}} \cdot \frac{(t - t_j)^{n+1}}{(n+1)!} + \cdots$$

When the difference $t - t_j$ is small or when the n–th derivative of $x(t)$ is constant (as it happens in QSSn methods) the difference between $q_i(t)$ and $x_i(t)$ results:

$$\Delta x_i(t) \approx \frac{\mathrm{d}^n x_i(t_j)}{\mathrm{d}t^n} \cdot \frac{(t - t_j)^n}{n!} \tag{11}$$

After $t = t_j$, the next discontinuity in $q_i(t)$ occurs at $t = t_{j+1}$, where $|\Delta x_i(t)| = \Delta Q_i$. Then, from Eq.(11) it results that

$$\Delta Q_i \approx \left| \frac{\mathrm{d}^n x_i(t_j)}{\mathrm{d}t^n} \right| \cdot \frac{(t_{j+1} - t_j)^n}{n!}$$

Dividing the latter by ΔQ_i and computing the $1/n$ power at both sides, it results

$$1 \approx \left| \frac{\frac{\mathrm{d}^n x_i(t_j)}{\mathrm{d}t^n}}{n!} \right|^{1/n} \cdot \left(\frac{1}{\Delta Q_i} \right)^{1/n} \cdot (t_{j+1} - t_j)$$

This equation holds for $j = 0, \cdots, k-1$ in the interval (t_0, t_k). Then, we can compute the summation for j at both sides:

$$\sum_{j=0}^{k-1} 1 \approx \sum_{j=0}^{k-1} \left| \frac{\frac{\mathrm{d}^n x_i(t_j)}{\mathrm{d}t^n}}{n!} \right|^{1/n} \cdot \left(\frac{1}{\Delta Q_i} \right)^{1/n} \cdot (t_{j+1} - t_j)$$

and approximating the summatory by the integral, we finally obtain

$$k \approx \left(\frac{1}{\Delta Q_i} \right)^{1/n} \int_{t_0}^{t_k} \left| \frac{\frac{\mathrm{d}^n x_i(t)}{\mathrm{d}t^n}}{n!} \right|^{1/n} \mathrm{d}t$$

which provides an expression for the number of discontinuities in $q_i(t)$ on the interval (t_0, t_k).

From this last expression, it makes sense to define the n–th order activity of the signal $x_i(t)$ on the interval (t_0, t_f) as

$$A^{(n)}_{x_i(t_0,t_f)} \triangleq \int_{t_0}^{t_f} \left| \frac{\frac{\mathrm{d}^n x_i(t)}{\mathrm{d}t^n}}{n!} \right|^{1/n} \mathrm{d}t \tag{12}$$

In that way, given a continuous time signal $x_i(t)$ we can estimate the *number of discontinuities* for an approximation of order n using a quantum ΔQ_i as:

$$k^{(n)}_{x_i(t_0,t_f)}(\Delta Q_i) \approx \frac{A^{(n)}_{x_i(t_0,t_f)}}{(\Delta Q_i)^{1/n}} \tag{13}$$

The following is a list of considerations about Eq.(12) and some of its relevant implications:

- When $n = 1$ Eq.(12) coincides with the original definition of activity of Eq.(1) and the formulae for estimating the number of discontinuities given by Eqs. (13) and (2) become identical.
- With Eq.(12) we also extended to the n–th order the concept that the activity measure is a property inherent to a signal, in contrast to the number of discontinuities computed by Eq.(13), which depends on the choice of the quantum size according the required accuracy.
- While the activity of order 1 measures the rate of change of the continuous signal, the activity of n–th order takes into account the rate of change of the signal's derivatives.
- Eq.(12) generalizes the original definitions and results found in [22] that foresee applications of continuous activity beyond the efficient simulation of differential equations, e.g. to improvements in techniques of data sensing, data compression, communication in multi–stage computations, or spatial quantization.

6 Examples

In this section we apply the new concept of activity of order n on two simple linear systems.

6.1 A first order linear system

The first order linear system:

$$\dot{x}(t) = a \cdot x(t)$$

has solution $x(t) = x(0) \cdot e^{a \cdot t}$.

The n–th order activity of the solution $x(t)$ is, according to Eq.(12),

$$A^{(n)}_{x(0,t_f)} = n \cdot |1 - e^{a \cdot t_f / n}| \cdot \left| \frac{x_0}{n!} \right|^{1/n} \tag{14}$$

When a is negative and $|a \cdot t_f| >> n$ it results that

$$A^{(n)}_{x(0,t_f)} \approx n \cdot \left| \frac{x(0)}{n!} \right|^{1/n}$$

Notice that the activity in this case is independent on the eigenvalue a.

Using the parameter $a = -1$, initial condition $x(0) = 1$ and a final time $t_f = 5$ the activities of order 1, 2, and 3, according to Eq.(14), result

$$A^{(1)}_{x(0,t_f)} = 0.993262; \quad A^{(2)}_{x(0,t_f)} = 1.298128; \quad A^{(3)}_{x(0,t_f)} = 1.339137 \tag{15}$$

We simulated the system with QSS1, QSS2 and QSS3 methods using in each case quanta $\Delta Q = 10^{-2}$, $\Delta Q = 10^{-4}$ and $\Delta Q = 10^{-6}$. Then, we compared the number of steps performed by each method with the number of steps predicted by Eq.(13). The results are shown in Table 1.

6.1.1 Analysis of the results

The results agree with the theoretical predictions. There are only two cases in which the results do not coincide: the QSS1 simulation with a small quantum and the QSS3 simulation with a large quantum.

	QSS1		QSS2		QSS3	
	$k^{(1)} = \frac{A^{(1)}}{\Delta Q}$	Steps	$k^{(2)} = \frac{A^{(2)}}{\sqrt{\Delta Q}}$	Steps	$k^{(3)} = \frac{A^{(3)}}{(\Delta Q)^{1/3}}$	Steps
$\Delta Q = 10^{-2}$	99.3262	100	12.981	13	6.2157	12
$\Delta Q = 10^{-4}$	9932.62	9933	129.81	130	28.8508	28
$\Delta Q = 10^{-6}$	993262	983881	1298.1	1298	133.914	133

Table 1: Theoretical and real number of quantum crossings.

In the first case, the steps are very small (there are more than 980000 steps in 5 seconds). As a consequence, the round–off errors become significant.

In the second case, the difference is due to the fact that QSS3 starts with a first order approximation (in the first step it does not have information about the first derivative) and then it follows with a second order one. Only after the third step it really performs a third order approximation.

It is also important to recall that QSS methods usually introduce some *spurious oscillations*, i.e., oscillations of the numerical solution that do not exist in the analytical solution. The presence of spurious oscillations provoke additional steps that are not predicted by the activity of the analytical solution (i.e., the activity of the numerical solution may be larger than the activity of the analytical solution).

6.2 A second order stiff system

Stiff systems are a class of dynamical systems where slow and fast dynamics coexist. Stiffness enforces classic explicit numerical solvers to use very small steps in order to obtain numerically stable solutions.

Unfortunately, non–stiff quantization–based algorithms, such as QSS methods, experience similar difficulties. When a stiff system is solved by a QSS method, spurious high frequency oscillations appear in the numerical solution [3, 15, 14]. In this context, we call *spurious* oscillations to those that are exhibited by the numerical solution but are not present in the analytical solution.

Due to these oscillations, the activity of the numerical solution results much higher than the activity of the analytical solution. Thus, the number of steps performed by non–stiff QSS methods loose any relation with the theoretical figures predicted by Eq.(13).

However, a special branch of state-quantization based methods has been recently developed that efficiently simulate many stiff systems. Backward and Linearly Implicit QSS methods (BQSS, LIQSS) [15, 14] tend to eliminate the spurious oscillations.

In this example, we consider the following second order system that illustrates these facts

$$\begin{aligned} \dot{x}_1 &= 0.01 \cdot x_2(t) \\ \dot{x}_2 &= -100 \cdot x_1(t) - 100 \cdot x_2(t) + u \end{aligned} \tag{16}$$

The system above has eigenvalues $\lambda_1 \approx -0.01$ and $\lambda_2 \approx -99.99$, which implies that it is a stiff system. Its analytical solution has the following structure:

$$\begin{aligned} x_1(t) &= c_1 \cdot e^{\lambda_1 \cdot t} + c_2 \cdot e^{\lambda_2 \cdot t} + c_3 \\ x_2(t) &= c_4 \cdot e^{\lambda_1 \cdot t} + c_5 \cdot e^{\lambda_2 \cdot t} + c_6 \end{aligned} \tag{17}$$

with coefficients c_i $(i = 1, \ldots, 6)$ depending on the initial conditions $x_1(0), x_2(0)$ and the constant input term u.

The n–th order activity of the solutions $x_1(t)$ and $x_2(t)$ results, according to Eq.(12), as follows:

$$
\begin{aligned}
A_1^{(n)}(t_f) = A^{(n)}_{x_1(0,t_f)} &= n\cdot\left|1-e^{\lambda_1\cdot t_f/n}\right|\cdot\left|\frac{c_1}{n!}\right|^{1/n} + n\cdot\left|1-e^{\lambda_2\cdot t_f/n}\right|\cdot\left|\frac{c_2}{n!}\right|^{1/n}\\
A_2^{(n)}(t_f) = A^{(n)}_{x_2(0,t_f)} &= n\cdot\left|1-e^{\lambda_1\cdot t_f/n}\right|\cdot\left|\frac{c_4}{n!}\right|^{1/n} + n\cdot\left|1-e^{\lambda_2\cdot t_f/n}\right|\cdot\left|\frac{c_5}{n!}\right|^{1/n}
\end{aligned}
\tag{18}
$$

Selecting initial conditions $x_1(0) = 0, x_2(0) = 20$ and the input[1] $u = 2000 + 200/9$ we obtain $c_1 = 0.0000224287, c_2 = 20.2222, c_3 = 0, c_4 = -0.224265, c_5 = 20.2243, c_6 = 0$.

Finally, for a final simulation time of $t_f = 500$, the theoretical activity for orders of approximation $n = 1$ to 3 results as follows:

$$A_1^{(1)} \approx 21;\quad A_1^{(2)} \approx 5.8;\quad A_1^{(3)} \approx 3.7 \tag{19}$$

for variable $x_1(t)$, and

$$A_2^{(1)} \approx 20.3;\quad A_2^{(2)} \approx 6.5;\quad A_2^{(3)} \approx 4.6 \tag{20}$$

for variable $x_2(t)$.

As in the previous example, we simulated the system with quantization–based methods of orders n=1 to 3. This time around, besides using QSS methods we also simulated using the LIQSS family for stiff systems.

We grouped the experiments according to the order n, adopting an initial quantum of $\Delta Q_i = 1$ and a final quantum of $\Delta Q_i = 10^{-3\cdot n}$ with decrements by one order of magnitude.

Finally, we compared the number of steps performed by QSS and LIQSS methods against what is predicted by Eq.(13). The results are shown in Tables 2, 3 and 4 and analyzed below.

	Number of steps at q_1			Number of steps at q_2		
	$k^{(1)} = \frac{A^{(1)}}{\Delta Q_i}$	QSS1	LIQSS1	$k^{(1)} = \frac{A^{(1)}}{\Delta Q_i}$	QSS1	LIQSS1
$\Delta Q_i = 1$	21	21	20	20.3	17279	24
$\Delta Q_i = 10^{-1}$	200.1	202	201	203.1	17224	206
$\Delta Q_i = 10^{-2}$	2008.6	2010	2009	2031.2	16256	2032
$\Delta Q_i = 10^{-3}$	20086	20087	20086	20312.4	26657	20287

Table 2: First order non-stiff (QSS1) and stiff (LIQSS1) methods. Theoretical and real number of steps.

6.2.1 Analysis of results

We analyze the behavior of the quantized state variables q_1 and q_2 present in the quantized version of system (16):

$$
\begin{aligned}
\dot{x}_1(t) &= 0.01\cdot q_2(t)\\
\dot{x}_2(t) &= -100\cdot q_1(t) - 100\cdot q_2(t) + u
\end{aligned}
\tag{21}
$$

[1] The input value u was chosen so that it is not a multiple of the quantum ΔQ_i. Otherwise, under certain conditions, the first order accurate QSS1 may not exhibit spurious oscillations, as analyzed in [3]

	Number of steps at q_1			Number of steps at q_2		
	$k^{(2)} = \frac{A^{(2)}}{\sqrt{\Delta Q_i}}$	QSS2	LIQSS2	$k^{(2)} = \frac{A^{(2)}}{\sqrt{\Delta Q_i}}$	QSS2	LIQSS2
$\Delta Q_i = 1$	5.8	10	7	6.5	65453	10
$\Delta Q_i = 10^{-1}$	18.5	19	19	20.6	65443	21
$\Delta Q_i = 10^{-2}$	58.4	58	59	65	65440	65
$\Delta Q_i = 10^{-3}$	184.9	184	184	205.8	65412	207
$\Delta Q_i = 10^{-4}$	584.4	585	585	650.8	65379	650
$\Delta Q_i = 10^{-5}$	1848.1	1848	1848	2058	65257	2095
$\Delta Q_i = 10^{-6}$	5844.4	5842	5843	6507.7	64056	7506

Table 3: Second order stiff and non-stiff methods. Theoretical and real number of steps.

	Number of steps at q_1			Number of steps at q_2		
	$k^{(3)} = \frac{A^{(3)}}{(\Delta Q_i)^{1/3}}$	QSS3	LIQSS3	$k^{(3)} = \frac{A^{(3)}}{(\Delta Q_i)^{1/3}}$	QSS3	LIQSS3
$\Delta Q_i = 1$	3.7	412	9	4.6	92391	11
$\Delta Q_i = 10^{-1}$	8	412	12	10	92395	16
$\Delta Q_i = 10^{-2}$	17.1	412	17	21.6	92397	27
$\Delta Q_i = 10^{-3}$	36.5	413	36	46.5	92384	50
$\Delta Q_i = 10^{-4}$	79.6	409	79	100.2	92393	101
$\Delta Q_i = 10^{-5}$	171.5	394	171	215.9	92415	212
$\Delta Q_i = 10^{-6}$	369.5	439	369	465.2	92458	461
$\Delta Q_i = 10^{-7}$	796.1	783	795	1002.2	92511	996
$\Delta Q_i = 10^{-8}$	1715.1	1715	1713	2159.1	92721	2154
$\Delta Q_i = 10^{-9}$	3695.1	3693	3692	4651.7	93169	4651

Table 4: Third order stiff and non-stiff methods. Theoretical and real number of steps.

First order methods

In Table 2, for the quantized state variable q_1 we can observe how, as expected, as ΔQ decreases the QSS1 Steps grow approximately linearly with the growth in the precision demand.

In contrast, the QSS1 Steps for the quantized variable q_2 appear unrelated with the accuracy settings and are dominated by large numbers.

Figures 4 and 5 explain this fact. The presence of fast spurious oscillations in $q_2(t)$ is the reason for the number of steps performed by that variable. These oscillations appear only in $q_2(t)$ because the expression of $\dot{x}_2(t)$ in Eq.(21) contains a large term $-100 \cdot q_2(t)$ and consequently, a change in $q_2(t)$ provokes a large change in $\dot{x}_2(t)$ which soon provokes a new change in $q_2(t)$.

The congruence between the theoretical activity and the practical number of QSS1 Steps is therefore very good for q_1, although unrecognizable for q_2.

Meanwhile, Figure 6 shows the evolution of $q_1(t)$ and $q_2(t)$ using the LIQSS1 approximation. LIQSS methods were designed to efficiently simulate stiff systems avoiding spurious oscillations, which in this case disappeared.

Consequently, in Table 2, the LIQSS1 method retains for both quantized variables the correspondence between theoretical activity and number of LIQSS1 Steps as the precision grows.

The analysis above evidences how activity acts as a theoretical baseline (in fact, as a lower bound) against which the performance of quantization–based methods can be compared.

Second and third order methods.

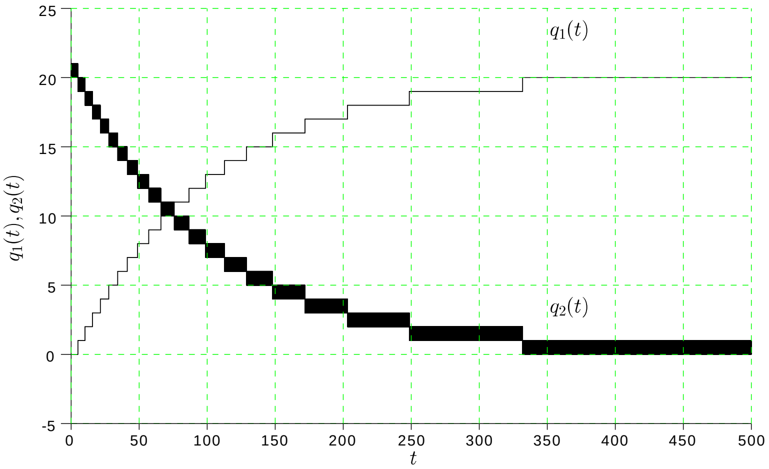

Figure 4: QSS1 solution of the stiff system of Eq.(16) with $\Delta Q_i = 1$ showing spurious oscillations on $q_2(t)$

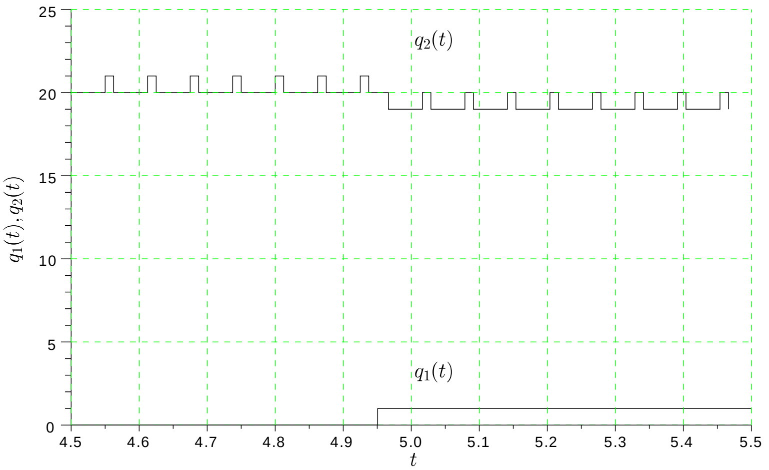

Figure 5: QSS1 solution of the stiff system of Eq.(16) with $\Delta Q_i = 1$ showing spurious oscillations on $q_2(t)$ (detail)

The considerations made before for first order methods also apply to higher order algorithms. Tables 3 and 4 show the expected evolution of QSS2 Steps and QSS3 Steps for q_1 as the precision grows, except when the quantum is large in QSS3. Here, the problem is that the large spurious oscillations in the slope of $q_2(t)$ are reflected in the second derivative $\ddot{x}_1(t)$ and thus QSS3 performs several steps.

Otherwise, as expected, in the QSS2 case steps grow with the square root of the increment in

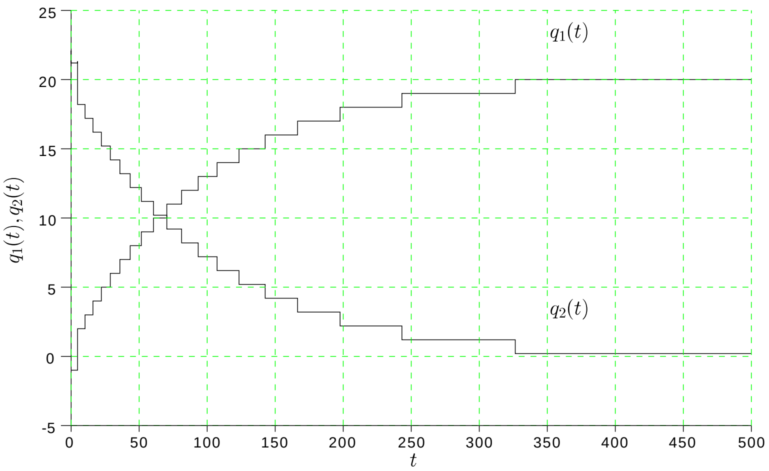

Figure 6: LIQSS1 solution of the stiff system of Eq.(16) with $\Delta Q_i = 1$.

the precision, and in the QSS3 case with the cubic root. For q_2 the number of practical steps again do not adhere to any relation with the precision nor the theoretical activity, presenting very high figures as a consequence of the already known spurious oscillations.

On the other hand, the stiff solvers LIQSS2 and LIQSS3 do not exhibit spurious oscillations and the number of steps they perform is consistent with what is predicted by the theoretical activity.

7 Conclusions and open issues

We have presented a generalization of the concept of activity for continuous time signals. While the original definition of activity [6] measures the rate of change of the signal, the new definition of activity of n–th order takes into account the rate of change of its higher order derivatives.

By so doing, this new concept allows to estimate the number of steps performed not only by first order quantization–based numerical integration algorithms such as QSS1, but also the number of steps performed by higher order methods.

This fact was analyzed over two simple examples, where the number of steps performed by the QSSn and LIQSSn algorithms and the theoretical estimations based on the activity of order n agreed in most cases for different orders and accuracy settings.

The second example also evidenced that, when trying to simulate a stiff system with non–stiff solvers like QSSn, spurious oscillations appear and the activity of the analytical and numerical solutions are far away from each other. Consequently, the number of steps performed by the solver is higher than what the activity predicts.

For this reason, the theoretical estimation provided by the activity of order n is in fact a lower bound for the number of steps performed by an algorithm of order n. This lower bound can be compared with the actual number of steps given by an algorithm measuring how suitable is that algorithm for simulating the system.

We remark that the results presented in this work are, in principle, of theoretical value. The exact computation of the activity of order n (including the original case of $n = 1$) requires knowing the analytical solution of the system, which is impossible to obtain except for very simple cases.

However, the new concept formalizes the relationship between activity and quantization–based simulation of continuous systems for higher order algorithms. It also establishes a formal proof about the relationship between computational costs (which depend on the number of steps performed) and the accuracy of a simulation (which depends linearly on the quantum ΔQ_i) for a method of order n. Thus, after performing a simulation with a method of order n with a quantum ΔQ_i, we now formally know that a new simulation with a quantum $\Delta Q_i/M$ will perform $\sqrt[n]{M}$ times more steps (provided that the numerical and analytical activity are similar, i.e., that there are not spurious oscillations).

Some possible next steps arise from the results presented in this work, which are open issues for future research to be done in the activity area:

- Explore how the knowledge of the activity for each variable in a given system can be exploited to derive optimal model partitioning and mapping to multiple parallel processing nodes (cores, processors, servers) in order to maximize speedups as compared against a serial (single node) simulation.
- Derive a possible definition of a *vector activity* that measures the complete activity of multidimensional signals. This measure should estimate the number of steps needed by classic discrete time numerical algorithms. Then, a comparison between the *vector* activity and the *scalar* activity in a system could be used to decide on the convenience of using discrete–time or quantization–based numerical algorithms.
- As we mentioned above, computing the activity requires knowing the analytical solution of a system. Work is needed on establishing conditions under which the activity can be directly computed from a numerical solution.
- As suggested by a reviewer, it would be useful to study how n–th order activity can be estimated from observations of a system's behaviour as early or easily as possible, for instance, determining (or approximating) bounds on the system's n–th derivative, which would in turn determine a bound on the theoretical minimum number of steps required to simulate the system in a given time interval.

We are currently exploring some of these research lines.

References

[1] O. Balci. The implementation of four conceptual frameworks for simulation modeling in high-level languages. In *Proceedings of the 20th conference on Winter simulation*, pages 287–295. ACM, 1988.

[2] Jerry Banks, John S. Carson, Barry L. Nelson, and David M. Nicol. *Discrete-Event System Simulation (5th Edition).* Prentice Hall, 5 edition, 2009.

[3] Francois E. Cellier and Erenesto Kofman. *Continuous System Simulation.* Springer, New York, 2006.

[4] G.S. Fishman. *Concepts and methods in discrete event digital simulation.* John Wiley and Sons, New York, 1973.

[5] X. Hu and B. P. Zeigler. Linking information and energy - activity-based energy-aware information processing. *Simulation: transactions of the society of modeling and simulation international*, 89(4):435–450, 2013.

[6] Rajanikanth Jammalamadaka. Activity characterization of spatial models: Application to discrete event solution of partial differential equations. Master's thesis, The University of Arizona, 2003.

[7] P.J. Kiviat. Digital computer simulation: Modeling concepts. Technical report, RAND Corporation, Santa Monica, California, 1967.

[8] P.J. Kiviat. Digital computer simulation: Computer programming languages. Technical report, RAND Corporation, Santa Monica, California, 1969.

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

[10] E. Kofman. A Third Order Discrete Event Simulation Method for Continuous System Simulation. *Latin American Applied Research*, 36(2):101–108, 2006.

[11] E. Kofman and S. Junco. Quantized State Systems. A DEVS Approach for Continuous System Simulation. *Transactions of SCS*, 18(3):123–132, 2001.

[12] M.R. Lackner. Toward a general simulation capability. In *Proceedings of the Spring Joint Computer Conference*, pages 1–14, San Francisco, California, 1962. ACM.

[13] M.R. Lackner. Digital simulation and system theory. Technical Report SP-1612, System Development Corp., Santa Monica, Califoria, 1964.

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

[15] G. Migoni, E. Kofman, and F. Cellier. Quantization-Based New Integration Methods for Stiff ODEs. *Simulation: Transactions of the Society for Modeling and Simulation International*, 88(4):387–407, 2012.

[16] A. Muzy and D.R.C. Hill. What is new with the activityworld view in modeling and simulation ? using activity as a unifying guide for modeling and simulation. In *Proceedings of the 2011 Winter Simulation Conference (WSC)*, pages 2882–2894. IEEE/SMC-ACM/SIGSIM, 2011.

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

[18] A. Muzy and B.P. Zeigler. Introduction to the activity tracking paradigm in component-based simulation. *Open Cybernetics & Systemics Journal*, 2:30–38, 2008.

[19] Alexandre Muzy, Franck Varenne, Bernard P Zeigler, Jonathan Caux, Patrick Coquillard, Luc Touraille, Dominique Prunetti, Philippe Caillou, Olivier Michel, and David RC Hill. Refounding of the activity concept? towards a federative paradigm for modeling and simulation. *Simulation*, 89(2):156–177, 2013.

[20] J. F. Santucci and L. Capocchi. Implementation and analysis of devs activity-tracking with devsimpy. *ITM Web of Conferences*, 1, 2013.

[21] B. Zeigler and J.S. Lee. Theory of quantized systems: formal basis for DEVS/hla distributed simulation environment. In *SPIE Proceedings*, pages 49–58, 1998.

[22] Bernard P. Zeigler, Rajanikanth Jammalamadaka, and Salil R. Akerkar. Continuity and change (activity) are fundamentally related in devs simulation of continuous systems. In TagGon Kim, editor, *Artificial Intelligence and Simulation*, volume 3397 of *Lecture Notes in Computer Science*, pages 1–13. Springer Berlin Heidelberg, 2005.

[23] B.P. Zeigler, T.G. Kim, and H. Praehofer. *Theory of Modeling and Simulation. Second edition.* Academic Press, New York, 2000.