

A Quantization Approach for Solving IVPs with DEVS

Jean-Sébastien Bolduc

Hans Vangheluwe

Modelling, Simulation and Design Lab

(<http://moncs.cs.mcgill.ca/MSDL/>)

School of Computer Science

McGill University, Montréal

Abstract

We show how Initial Value Problems can be solved using a quantization algorithm, and analyse various properties of this approach. Quantization is the dual of discretization, in that the dependent variables' space is partitioned (rather than the independent variable's). The quantization approach is appealing as it better matches the discrete-event simulation scheme than the discretization approach. First, a non-adaptive quantization algorithm, expressed in the DEVS formalism and based on the Forward-Euler approximation, is presented. We show that consistency as well as convergence are respected for autonomous systems, but cannot be guaranteed for nonautonomous problems. Absolute-stability as it is usually defined is generally not achieved. We then introduce an adaptive quantization algorithm, which improves the overall performance of its non-adaptive counterpart.

1 Introduction

As an alternative to the traditional discretization approach to solve *Ordinary Differential Equation* (ODE) models, Zeigler [14] has proposed an approach based on a partitioning of the state space into several cells, or *quanta*. The *Discrete Event system Specification* (DEVS) formalism, first introduced by Zeigler [14] to provide a rigorous common basis for discrete-event modelling and simulation, is a natural choice as the target formalism for quantization as it captures the essence of discrete-event formalisms.

The work done in the field [7, 8, 2] has shown that quantization is a promising avenue. Rigorous analysis of the various strategies, however, is limited.

We will limit ourselves here to *Initial Value Problems* (IVPs), which consist of an ODE together with a set of initial conditions:

$$\begin{cases} \dot{\mathbf{x}} &= \mathbf{f}(t, \mathbf{x}) \\ \mathbf{x}(t_I) &= \mathbf{x}_I. \end{cases} \quad (1)$$

where \mathbf{x} is a column vector of length n (the *order* of the system), and \mathbf{f} is a vector function. The system is said to be *autonomous* if \mathbf{f} does not depend explicitly on the independent variable t .

The exact solution to the problem is a function $\mathbf{x} : \mathbb{T} \rightarrow \mathbb{R}^n$, defined and differentiable with respect to t over some interval $\mathbb{T} = [t_I, t_F]$ of \mathbb{R} . The approximate numerical solution \mathbf{X} is a sequence [9]

$$\mathbf{X} = \{(\mathbf{x}_i, h_i) \mid i = 0, 1, \dots, m, \mathbf{x}_i \in \mathbb{R}^n, h_i = t_{i+1} - t_i\}, \quad (2)$$

where h_i are the timesteps, $t_i = t_I + \sum_{u=0}^{i-1} h_u$.

This paper is organized as follows: in section 3 we present a quantization of the state space. In section 4 we first introduce a non-adaptive quantized approach to solve IVPs; we then proceed to study consistency, convergence and absolute-stability of this approach in section 5. A simple adaptive quantized approach is presented in section 6, and it is shown how the approach can be expressed in the DEVS formalism. We begin in the following section by introducing the atomic-DEVS formalism.

2 The DEVS Formalism

Classic DEVS is an intrinsically sequential formalism first introduced by Zeigler [14], which allows for the description of *discrete-event* system behaviour at two levels. At the lowest level, an *atomic-DEVS* describes the autonomous behaviour of a discrete-event system as a sequence of deterministic transitions between states as well as how it reacts to external inputs. At the higher level, a *coupled-DEVS* describes a discrete-event system in terms of a network of coupled components, each an atomic- or coupled-DEVS model.

Since in this article we will map an IVP onto an atomic-DEVS, we will only introduce this part of the formalism.

An atomic-DEVS M is specified by a 7-tuple

$$M = (X, Y, S, \delta_{\text{int}}, \delta_{\text{ext}}, \lambda, t_a),$$

where:

X is the *input set*,

Y is the *output set*,

S is the *partial state set*,

$\delta_{\text{int}} : S \rightarrow S$ is the *internal transition function*,

$\delta_{\text{ext}} : Q \times X \rightarrow S$ is the *external transition function*, where

$Q = \{(s, e) \mid s \in S, 0 \leq e \leq t_a(s)\}$ is the *total state set*, where e is the *elapsed time* since the last transition.

$\lambda : S \rightarrow Y$ is the *output function*,

$t_a : S \rightarrow \mathbb{R}^+ \cup \{0, +\infty\}$ is the *time advance function*.

There are no restrictions on the structure of the sets, which typically are product sets, *i.e.*, $S = S_1 \times S_2 \times \dots \times S_n$. The time base \mathbb{T} is not mentioned explicitly and is continuous (*i.e.*, \mathbb{R}). For a discrete-event

model described by an atomic-DEVS M , the behaviour is uniquely determined by the initial total state $(s_0, e_0) \in Q$ and is obtained by means of the following iterative simulation procedure:

At any time $t \in \mathbb{T}$, the system is in some state $s \in S$. If no external event occurs, the system will stay in state s until the elapsed time e reaches $t_a(s)$. The system then first produces the output value $\lambda(s)$ and subsequently makes a transition to state $s' = \delta_{\text{int}}(s)$. If an external event $x \in X$ occurs before e reaches $t_a(s)$, the system interrupts its autonomous behaviour and instantaneously goes to state $s' = \delta_{\text{ext}}((s, e), x)$. Thus, the internal transition function dictates the system's new state based on its old state in the absence of external events. The external transition function dictates the system's new state whenever an external event occurs, based on this event x , the current state s and how long the system has been in this state, e . After both types of transitions, the elapsed time e is reset to 0.

Note that $t_a(s)$ can take on the values 0 and $+\infty$. In the first case, the system stays in state s for a time so short that no external event can intervene, and we say that s is a *transitory state*. In the latter case, the system stays in state s forever unless an external event occurs, and we say that s is a *passive state*.

3 Quantization

The well-known discretization approaches to solve ODEs are based on a partitioning of the time domain. In the approach studied here, "quantization" is interpreted in very general terms as a tessellation of the state space. The reader is referred to [12] for an alternative definition.

Our quantization scheme is based on a regular tessellation of the space \mathbb{R} . The quantization \mathcal{P} is determined by two parameters: the *quanta size* $D > 0$, and the *quanta phase* ρ , subject to the constraint $0 \leq \rho < D$ (refer to Figure 1). Let the *Quanta Interfaces* (QIs) be the zeros of the harmonic function

$$\sin\left(\frac{\pi}{D}(x + \rho)\right).$$

The QIs are then given by $Dk - \rho$, where $k \in \mathbb{Z}$ is the *quantum index*. From this we define a *quantum* d_k as the interval

$$d_k = \left\{x \mid x \in \mathbb{R}, Dk - \rho \leq x < D(k + 1) - \rho\right\}.$$

In this scheme, we note that the origin is always an element of d_0 . We finally define the quantization of \mathbb{R} as $\mathcal{P} = \{d_k \mid k \in \mathbb{Z}\}$.

To find the quantum a given point x belongs to, we introduce the *index function* $\llbracket \cdot \rrbracket_{\mathcal{P}} : \mathbb{R} \rightarrow \mathbb{Z}$, which returns the index of the quantum that contains the point, under the quantization scheme \mathcal{P} :

$$\llbracket x \rrbracket_{\mathcal{P}} = \left\lfloor \frac{x + \rho}{D} \right\rfloor. \quad (3)$$

So we have $\llbracket x \rrbracket_{\mathcal{P}} = k \Leftrightarrow x \in d_k$. When no confusion is possible, the subscript \mathcal{P} can be dropped.

Quantization of higher-dimensional spaces are based on the quantization of \mathbb{R} . For instance, we need 2 one-dimensional quantizations \mathcal{P}_x and \mathcal{P}_y to construct a quantization \mathcal{P} of the plane:

$$\mathcal{P} = \{d_{kl} = d_k \times d_l \mid d_k \in \mathcal{P}_x, d_l \in \mathcal{P}_y\},$$

and we let $\llbracket(x, y)\rrbracket_{\mathcal{P}} = (\llbracket x \rrbracket_{\mathcal{P}_x}, \llbracket y \rrbracket_{\mathcal{P}_y})$. This results in a rectangular tessellation of the state space where each quantum is a hyper-rectangle.

The algorithm described in section 6 uses a variable-size quantization. To obtain this we first introduce a *base* quantization, defined by D_B and ρ_B (the base quanta size and quanta phase). A *scaling factor* $\omega \in \mathbb{N}_0$ is then used to define the actual quanta size as

$$D_\omega = \frac{D_B}{2^\omega}, \quad (4)$$

and the constraint $0 \leq \rho_\omega < D_\omega$ is satisfied when we use

$$\rho_\omega = \rho_B - D_\omega \left\lfloor \frac{\rho_B}{D_\omega} \right\rfloor. \quad (5)$$

A maximal scaling factor, ω_{\max} , can also be defined. By only allowing quanta sizes to be halved, we guarantee that QIs present in the base quantization are also present when $\omega > 0$. In other words, halving the quanta sizes only adds new QIs. This might be an essential property in some situations, for instance when we want to associate a *state event* [10] with a given QI.

4 Non-Adaptive Quantization Algorithm

We introduce a non-adaptive algorithm based on quantization that can solve the simple first-order, autonomous system

$$\begin{cases} \dot{x} &= f(x) \\ x(0) &= x_0. \end{cases} \quad (6)$$

The algorithm is based on the well-known *Forward Euler* approximation:

$$x_{i+1} = x_i + h_i \cdot f(x_i). \quad (7)$$

In the non-adaptive discretization approach, the timestep h_i is constant, and the difference equation is iteratively solved for x_{i+1} to build the state trajectory X . In other words, we can view non-adaptive discretization as the answer to the question “*in which state is the system going to be at a given future time?*”.

The approach is reversed in quantization: knowing x_{i+1} , we solve for the timestep h_i . This matches more closely the discrete-event style of expressing dynamics. We define the next state value x_{i+1} as *the QI closest to the current state x_i in the direction specified by the slope $f(x_i)$* . In quantization, events are associated with the state trajectory leaving a quantum. The next state value is computed with the function σ , which is expressed in terms of the index function (3) as

$$\sigma(x, \dot{x}) = \begin{cases} D(\llbracket x \rrbracket + 1) - \rho & \text{if } \dot{x} > 0, \\ -D(\llbracket -(x + 2\rho) \rrbracket + 1) - \rho & \text{if } \dot{x} < 0, \\ x & \text{if } \dot{x} = 0, \end{cases} \quad (8)$$

so that $x_{i+1} = \sigma(x_i, \dot{x}_i)$. The function returns the QI *above* or *below* x_i , when the slope is positive or negative, respectively. Using this in equation (7) we obtain

$$h_i = \begin{cases} \frac{\sigma(x_i, f(x_i)) - x_i}{f(x_i)} & \text{if } f(x_i) \neq 0, \\ \infty & \text{if } f(x_i) = 0. \end{cases} \quad (9)$$

We note that whenever $f(x_i) = 0$, we get $h_i = \infty$: this is justified by the fact that in autonomous systems, a zero slope corresponds to a fixed point. Non-adaptive quantization answers the questions “when will the state of the system reach a given value?”. This form of quantization has been called *predictive quantization* [4].

We present below the complete definition of an atomic-DEVS that solves IVPs using the above formulas. We note that in DEVS terminology, equation (8) solves for the *internal transition* of the model, while equation (9) solves for its *time-advance*. The strategy can be generalized to higher order systems by computing the timestep as

$$h_i = \min_j \{h_{i,j}\}, \quad (10)$$

where $h_{i,j}$ is the time-advance associated with the j^{th} component of the state vector \mathbf{x} . The next state is computed with

$$\mathbf{x}_{i+1} = \mathbf{x}_i + h_i \cdot \mathbf{f}(\mathbf{x}_i). \quad (11)$$

In the algorithm, all quanta sizes are assumed to be equal for every state x_1, x_2, \dots, x_n , although this could easily be generalized. We refer the reader to [2] for an extension where the atomic-DEVS may receive inputs.

atomic-DEVS — nonadaptive quantization

Given an IVP of the form (1), a quanta size D and a quanta phase ρ , we define:

- **Partial State**

The partial state s of the atomic-DEVS is a tuple

$$s = (\mathbf{x}, h, t)$$

where

\mathbf{x} is the next state of the ODE,

h is the time until the next event,

t is the global time at the next event (used in nonautonomous systems).

- **Internal Transition Function** $s' = \delta_{\text{int}}(s)$

1. Compute the current slope $\dot{\mathbf{x}}_i = \mathbf{f}(s.t, s.\mathbf{x})$;

2. For every component $x_{i,j}$, $j = 1, 2, \dots, n$ of $\mathbf{x}_i = s.\mathbf{x}$, compute $h_{i,j}$ according to equation (9);

3. Compute h_i and \mathbf{x}_{i+1} according to equations (10) and (11), respectively.

$$s' \leftarrow (\mathbf{x}_{i+1}, h_i, s.t + h_i)$$

- **Output Function** $y = \lambda(s)$

$$y \leftarrow s.\mathbf{x}$$

- **Time Advance Function** $h = t_a(s)$

$$h \leftarrow s.h$$

Finally, the initial partial state is given by

$$s = (\mathbf{x}_0, 0, t_I)$$

Figure 2 shows the result of an experiment with the circle test problem, defined as

$$\begin{cases} \dot{x} = y & x(0) = x_0 \\ \dot{y} = -x & y(0) = y_0, \end{cases}$$

and with the exact solution

$$\begin{aligned} x(t) &= x_0 \cdot \cos(t) - y_0 \cdot \sin(t) \\ y(t) &= -x_0 \cdot \sin(t) - y_0 \cdot \cos(t). \end{aligned}$$

The first quadrant of the solution is reproduced in Figure 3 with the quantization grid of size $D = 0.1$, to emphasize that events correspond to QI crossings. As expected in a second-order problem, the values of the approximate solution correspond to a QI of either partition of x or y .

It has been claimed [8, 14] that one of the main advantages of quantization is related to the reduction of the computational cost in solving a continuous system. To verify this claim, we compare the performance of our nonadaptive quantization algorithm to that of the well-known Euler discretization algorithm (refer to Figure 4). We performed several simulations, with both algorithms, of the circle test problem. Every simulation run covers the time interval $[0, 4\pi]$, and uses the same initial conditions, $x_0 = 0$ and $y_0 = 1$. Only the quanta size D and the timestep h are steadily varied, for quantization and discretization respectively, in each experiment. For each simulation run the *Number of Computation Steps* (NCS) is reported on the horizontal axis of the figure. On the vertical axis we show the error associated with an experiment, defined as

$$\text{Err} = \frac{1}{n} \sum_{i=0}^n \|\mathbf{x}_i - \mathbf{x}(t_i)\|,$$

where n is the NCS. The error thus corresponds to the average difference between the numerical and exact solutions, over a given time interval.

Figure 4 suggests that *for a same computational cost discretization will consistently yield a smaller error than quantization*. Conversely, for any given error, quantization is more computationally expensive. Even though the difference between the two curves is qualitatively small, this result seems to indicate an accuracy limit inherent to the quantization method introduced above. Furthermore, the difference between the curves is exacerbated if we use the number of flops (taking into account the cost of individual operations) instead of the NCS as a unit of computation cost.

The same experiment was repeated on a stiff system described by [7]

$$\begin{cases} \dot{x} = \frac{y}{L} & x(0) = x_0 \\ \dot{y} = U - \frac{x}{C} - \frac{R \cdot y}{L} & y(0) = y_0, \end{cases}$$

with $R = 100.01$, $L = 0.01$, $C = 0.01$ and $U = 100$. Figure 5 shows a solution $y(t)$ of the system. In Figure 6, we compare the performance of quantization and discretization on this stiff system in the same manner as we did before. Every simulation run covers the time interval $[0, 7]$, and initial conditions are $x_0 = 0$ and $y_0 = 0$. The error can be evaluated using the exact solution

$$\begin{cases} x(t) = \frac{1}{9999} (e^{-10000t} - 10000e^{-t}) \\ y(t) = \frac{100}{9999} (e^{-t} - e^{-10000t}). \end{cases}$$

The results are consistent with those obtained before: for a same computational cost discretization will consistently yield a smaller error than quantization.

We end this section by presenting the results obtained with a simple first-order system. As opposed to the previous systems, this one is non-autonomous. Consider

$$\dot{x} = \alpha x(t - \beta) \quad x(0) = x_0,$$

with $\alpha = 0.2$, $\beta = 5$. The exact solution is given by

$$x(t) = x_0 \cdot e^{\frac{\alpha t}{2}(t-2\beta)}.$$

Figure 7 shows a sample simulation run obtained with quantization, where the numerical solution is completely wrong. This behaviour will be explained in the next section in terms of the *Zero-slope problem*, when we study consistency and convergence of quantization.

5 Numerical Analysis

In this section we present results related to the numerical properties of quantization. Our analysis of consistency, convergence and stability is based on simple first-order, nonautonomous systems. The results can however be generalized to higher-order systems. This analysis illustrates the *Zero-slope problem* [3]: the fact that the quantized numerical solution cannot be trusted, generally, when the right-hand side of an ODE (1) gets close to zero.

5.1 Consistency

The *local truncation error* τ is defined as

$$\begin{aligned} \tau_{i+1} &= |x_{(i)}(t_{i+1}) - x_{i+1}| \\ &= \frac{h_i^2}{2} |\dot{f}(x(\xi))|, \quad t_i \leq \xi \leq t_{i+1}, \end{aligned} \quad (12)$$

based on a Taylor expansion with remainder. Here, $x_{(i)}(t)$ is the exact solution to the original ODE, but with initial condition $x(t_i) = x_i$.

A numerical approximation scheme is said to be *consistent* if the local truncation error goes to zero as the timestep h goes to zero. It is thus clear from equation (12) that Euler is consistent in the discretization case. In quantization, we derive from equation (7) the following inequality:

$$\begin{aligned} h_i &= \frac{x_{i+1} - x_i}{f(x_i)} \\ &\leq \frac{D \cdot \text{sign}(f(x_i))}{f(x_i)}. \end{aligned} \quad (13)$$

Hence the timestep is undefined when the slope is zero. To be consistent with equation (9) we use

$$h_i \leq \frac{D}{|f(x_i)|}. \quad (14)$$

We now show that our strategy is consistent when applied to a first-order, autonomous system, *i.e.*, $\tau \rightarrow 0$ as $D \rightarrow 0$.

First, observe that the existence and uniqueness of a solution to the problem (6) is guaranteed if f and f_x are continuous and bounded over the infinite strip

$$\mathbb{D} = \{(t, x) \mid t_I \leq t \leq t_F, |x| \leq \infty\}.$$

From this it follows that there exist positive constants C and L such that $\forall x, |f(x)| \leq C$ and $|f_x(x)| \leq L$ (the Lipschitz constant).

Let us assume that there exists a constant ε such that for all $x, |f(x)| \geq \varepsilon > 0$. Using equation (14) in (12) we get (for $t_i \leq \xi \leq t_{i+1}$)

$$\begin{aligned} \tau_{i+1} &\leq \frac{D^2}{2 \cdot |f(x_i)|^2} \cdot |\dot{f}(x(\xi))| \\ &= \frac{D^2}{2 \cdot |f(x_i)|^2} \cdot |f(x(\xi)) \cdot f_x(x(\xi))| \\ &\leq \frac{D^2}{2 \cdot \varepsilon^2} \cdot CL, \end{aligned}$$

in which case $\tau \in O(D^2)$.

Let us now remove the assumption: if $f(x) = 0$, then τ cannot be bounded from above. However we observe that in an autonomous system,

$$f(x(t_1)) = 0 \quad \Rightarrow \quad f(x(t_2)) = 0, \quad \forall t_2 \geq t_1.$$

This corresponds to a fixed point. It follows that even “after” an infinite time step, we have $\tau = 0$, which is in $O(D^2)$.

The proof can be generalized to higher-order systems. In the case of nonautonomous systems however,

$$f(t_1, x(t_1)) = 0 \quad \not\Rightarrow \quad f(t_2, x(t_2)) = 0, \quad \forall t_2 \geq t_1.$$

So unless we can assume, as before, a lower bound ε on the slope, we cannot guarantee consistency as an infinite timestep will result in an infinite τ .

5.2 Convergence

Connecting the points $x_0, x_1, x_2 \dots etc.$ by straight lines we obtain the *Euler polygon* $\tilde{x}(t)$, a continuous piecewise linear function that approximates $x(t)$ over \mathbb{T} . Kofman and Junco [7] recently proved an upper bound on the *global error* $E(t)$:

$$\begin{aligned} E(t) = |x(t) - \tilde{x}(t)| &\leq D(e^{Lt} - 1) \\ &\in O(D). \end{aligned} \tag{15}$$

This suggests linear convergence of the numerical solution towards the exact solution as D decreases. The same proof can be used with our quantization strategy, and we obtain the same error bound for autonomous systems. We will show however that, under our strategy, nonautonomous systems are problematic.

We observe that the slope of $\tilde{x}(t)$ is piecewise constant: for any t , the slope $\dot{\tilde{x}}(t)$ is the function f evaluated at the last QI crossed by the Euler polygon. In the case of a nonautonomous system, we have

$$\dot{\tilde{x}}(t) = f(t - \mu(t), \tilde{x}(t) - \delta(t)),$$

where $\delta(t)$ is the distance between $\tilde{x}(t)$ and the last QI crossed, while $\mu(t)$ is the time elapsed since that last QI crossing. The global error is then evaluated as

$$\begin{aligned} E(t) &= \left| \int_{t_I}^t (\dot{x}(s) - \dot{\tilde{x}}(s)) ds \right| \\ &\leq \int_{t_I}^t \left| f(s, x(s)) - f(s - \mu(s), \tilde{x}(s) - \delta(s)) \right| ds. \end{aligned}$$

Using a Taylor expansion of the second term in the integrand around $f(s, \tilde{x}(s))$ and using the fact that f is Lipschitz continuous in x , we have

$$\begin{aligned} E(t) &\leq L \int_{t_I}^t |x(s) - \tilde{x}(s)| ds + L \int_{t_I}^t |\delta(s)| ds + \\ &\quad + CL \int_{t_I}^t |\mu(s)| ds. \end{aligned}$$

In the case of autonomous systems, the last term is absent. Observing that $|\delta(t)| < D$, one can prove the bound (15) using the *Gronwall-Bellman inequality* [6], which states that

$$\begin{aligned} f(t) &\leq \alpha(t) + \int_0^t \beta(s) f(s) ds \\ \Rightarrow f(t) &\leq \alpha(t) + \int_0^t \alpha(s) \beta(s) e^{\int_s^t \beta(\tau) d\tau} ds. \end{aligned} \tag{16}$$

For a nonautonomous system however, we have

$$|\mu(t)| = \left| \frac{\delta(t)}{\dot{\tilde{x}}(t)} \right|,$$

which is undefined whenever $\dot{\tilde{x}}(t) = 0$. Thus we cannot prove convergence in that case.

5.3 Stability

We will look here at *absolute-stability*. For this we use the test equation

$$\begin{aligned} \dot{x} &= \lambda x, \\ x(0) &= x_0 \end{aligned}$$

whose exact solution is $x(t) = x_0 e^{\lambda t}$. We are interested in the case where $\lambda < 0$, which yields a necessary requirement for absolute stability [1],

$$|x_i| \leq |x_{i-1}|. \quad (17)$$

We start our analysis by rewriting equation (13) as

$$h_i = \frac{D \cdot \text{sign}(f(x_i))}{f(x_i)},$$

where we have equality if we assume without loss of generality that x_0 corresponds to a QI. Substituting into equation (7), we get

$$x_{i+1} = x_i + D \cdot \text{sign}(f(x_i)).$$

Instead of examining the characteristic equation of this when applied to the test problem, we will study stability by drawing the iterative map (see Figure 8).

It is clear that since all x_i fall on a QI, our numerical solution will reach the fixed point of the map only if that point falls on a QI. Otherwise, the solution will oscillate between two values $x_a > 0$ and $x_b < 0$, $x_a - x_b = D$ as shown in Figure 8.

We can say that *in general* the absolute stability requirement (17) is not met. However, the numerical solution will never diverge, and the oscillations are usually small since the quanta size D is typically small.

5.4 Discussion

Through our analysis, we showed that the presence of zero slopes is problematic: in the case of autonomous systems, it might result in a timestep that is infinite. In the case of nonautonomous systems, we cannot generally show consistency nor convergence. We observe that the dual problem in discretization— infinite slope—is ruled out by the existence and uniqueness conditions imposed by Lipschitz continuity in the dependent variables.

One obvious possibility around that problem would be to transform nonautonomous systems into autonomous ones, by introducing a new dependent variable $z = t$ such that $\dot{z} = 1$. This strategy results in the time axis being quantized—or partitioned—just as in discretization. So we end up with a hybrid approach that combines both quantization and discretization, with no immediate benefits.

Another possibility around the Zero-slope problem would be to saturate the timestep when the slope is “small”. Observe that saturating h with some constant c would not work, since then we still obtain $\tau \in O(1) \supseteq O(D^2)$. A better approach would be to impose $h \leq D$. However, this strategy is very close to the first one, resulting once again in a hybrid approach.

6 Adaptive Quantization

In this section we present a first attempt at an adaptive quantized approach based on the non-adaptive algorithm presented in section 4. Earlier references to “dynamic quantization” can be found in [13, 11].

Adaptation is used in discretization to reduce, on the average, the number of approximations x_i , while still guaranteeing a certain accuracy in terms of the local truncation error. The simple idea is to use at each step the largest timestep h_i that will keep the local truncation error τ_{i+1} below a predefined tolerance TOL .

The same rationale motivates the use of adaptation in quantized algorithms. Another important reason for adaptation in quantized algorithms is illustrated by our results on stability: an adaptive algorithm will guarantee that the oscillations encountered while studying absolute-stability are limited by the tolerance TOL .

We will once again consider first-order, autonomous systems of the form (6).

Adaptive strategies are based on an approximation Δ_i of the local truncation error τ_i . At each step the quanta size D_i are recursively halved until we respect the condition

$$\Delta_i \leq TOL. \quad (18)$$

To allow for the quanta size to grow, the initial guess for D_{i+1} is tentatively set to $2D_i$. Using this approach, $D_i \leq D_i^{\text{opt}}$, where D_i^{opt} is the largest quanta size such that $\Delta_i = TOL$. Note that doubling the quanta size is *not* recursive: it follows that over a simulation, the scaling factor ω can jump from a low to a high value, but it can only increase smoothly.

The exact local truncation error τ_i is defined in equation (12). Based on the *Richardson extrapolation* [5], the approximate local truncation error Δ_i is evaluated as

$$\Delta_{i+1} = |\hat{x}_{i+1} - x_{i+1}|,$$

where \hat{x}_k is a higher-order numerical approximation, obtained by taking 2 half-steps (see Figure 9):

$$\begin{aligned} \hat{x}_{i+\frac{1}{2}} &= x_i + \frac{h_i}{2} f(x_i) \\ \hat{x}_{i+1} &= \hat{x}_{i+\frac{1}{2}} + \frac{h_i}{2} f(\hat{x}_{i+\frac{1}{2}}). \end{aligned}$$

So we obtain

$$\Delta_{i+1} = \frac{h_i}{2} \left| f(\hat{x}_{i+\frac{1}{2}}) - f(x_i) \right|, \quad (19)$$

The timestep h_i is determined by the quanta size D_i through equation (9). Whereas in adaptive discretization we normally choose the highest-order approximation \hat{x}_{i+1} for the next step, we use the lowest-order approximation x_{i+1} in adaptive quantization. Choosing otherwise would not guarantee that the next approximation corresponds to a QI, as illustrated in Figure 9.

This strategy does not generalize well to higher-order systems, where it might only be necessary to halve one of the n quanta sizes to respect the tolerance. Nevertheless, the atomic-DEVS we describe in the algorithm below extends the idea by imposing the conservative constraint that all quanta sizes (and phases) be identical at each time, resulting in a homogeneous quantization.

The 3-state FSA shown in Figure 10 controls the dynamics of the atomic-DEVS. Each state in the figure corresponds to the possible values for the partial state element $s.\alpha$. Arrows represent internal transitions, and are labelled with the previous values returned by the time advance and output functions. When more than one transition leaves a FSA state, the precondition is specified within brackets.

atomic-DEVS — adaptive quantization

Given an IVP of the form (1), a base quanta size D_B and a base quanta phase ρ_B , we define:

- **Partial State**

The partial state s of the atomic-DEVS is a tuple

$$s = (\alpha, \mathbf{x}_i, \dot{\mathbf{x}}_i, t_i, \mathbf{x}_{i+1}, \omega_i, D_i, \rho_i, \omega_{i+1}, h_i, \Delta_{i+1}, \Delta_i)$$

where

$\alpha \in \{0, 1, 2\}$ is the FSA state,

\mathbf{x}_i is the current state of the ODE,

$\dot{\mathbf{x}}_i$ is the current derivative (stored for efficiency),

t_i is the current global time (used in nonautonomous systems),

\mathbf{x}_{i+1} is the current guess of the next state of the ODE (stored for efficiency),

ω_i is the current scaling factor,

D_i is the current quanta size (stored for efficiency),

ρ_i is the current quanta phase (stored for efficiency),

ω_{i+1} is the initial guess of the scaling factor for the next step,

h_i is the timestep to get to the next event,

Δ_{i+1} is the approximation of the local truncation error while going to the next step,

Δ_i is the approximation of the local truncation error while going from the previous to the current step (remembered for output).

- **Internal Transition Function** $s' = \delta_{\text{int}}(s)$

- If $s.\alpha = 0$

1. For every component $x_{i,j}$, $j = 1, 2, \dots, n$ of $s.\mathbf{x}_i$, compute $h_{i,j}$ according to equation (9);
2. Compute h_i , \mathbf{x}_{i+1} and Δ_{i+1} according respectively to equations (10), (11) and (19).

$$s' \leftarrow \begin{cases} (1, s.\mathbf{x}_i, s.\dot{\mathbf{x}}_i, s.t_i, \mathbf{x}_{i+1}, s.\omega_i, \dots \\ \quad s.D_i, s.\rho_i, \max\{s.\omega_i - 1, 0\}, h_i, \Delta_{i+1}, s.\Delta_i) & \text{if } \Delta_{i+1} \leq TOL \\ (s.\alpha, s.\mathbf{x}_i, s.\dot{\mathbf{x}}_i, s.t_i, s.\mathbf{x}_{i+1}, \min\{s.\omega_i + 1, \omega_{\max}\}, \dots \\ \quad D_{s'.\omega_i}, \rho_{s'.\omega_i}, \omega_{i+1}, h_i, \Delta_{i+1}, \Delta_i) & \text{if } \Delta_{i+1} > TOL \end{cases}$$

- If $s.\alpha = 1$

$$s = (2, \mathbf{x}_i, \dot{\mathbf{x}}_i, t_i, \mathbf{x}_{i+1}, \omega_i, D_i, \rho_i, \omega_{i+1}, h_i, \Delta_{i+1}, \Delta_i)$$

- If $s.\alpha = 2$

Compute $\dot{\mathbf{x}}_{i+1} = \mathbf{f}(t_{i+1}, s.\mathbf{x}_{i+1})$.

$$s = (0, s.\mathbf{x}_{i+1}, \dot{\mathbf{x}}_{i+1}, s.t_i + s.h_i, \emptyset, s.\omega_{i+1}, D_{s'.\omega_i}, \rho_{s'.\omega_i}, \emptyset, \emptyset, \emptyset, \Delta_{i+1})$$

- **Output Function** $y = \lambda(s)$

- If $s.\alpha \in \{0, 2\}$ — mute states

$$y \leftarrow \emptyset$$

- If $s.\alpha = 1$

$$y \leftarrow \{s.x_i, s.D_i, s.h_i, s.\Delta_i\}$$

The fields $s.D_i$ and $s.\Delta_i$ are included in the output for our analysis only.

- **Time Advance Function** $h = t_a(s)$

- If $s.\alpha \in \{0, 1\}$ — transitory states

$$h \leftarrow 0$$

- If $s.\alpha = 2$

$$h \leftarrow s.h_i$$

Finally, the initial state is given by

$$s_0 = (0, \mathbf{x}_0, \dot{\mathbf{x}}_0, t_I, \emptyset, \omega_0, D_0, \rho_0, \emptyset, \emptyset, \emptyset, 0),$$

where $\omega_0 \geq 0$ is the initial scaling factor.

Results on the circle test problem are show in Figure 11, which demonstrates the effect of TOL on the solution. Figure 12 is shows the evolution of the approximate local truncation error Δ_i , and how the quanta size D_i varies accordingly.

Conclusion

We presented a simple non-adaptive quantization algorithm in section 4. Through our analysis, we showed that the presence of zero slopes is problematic: in the case of autonomous systems, it might result in a timestep that is infinite. In the case of nonautonomous systems, consistency and convergence cannot be guaranteed.

In section 6 we presented an adaptive quantized algorithm. Although very simple, we see an improvement in the performance of the new algorithm with respect to the non-adaptive version. However, adaptive quantization suffers from the same problems as the non-adaptive version.

References

- [1] Uri M. Ascher and Linda R. Petzold. *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*. SIAM, Philadelphia, PA, 1998.
- [2] Jean-Sébastien Bolduc and Hans Vangheluwe. Expressing ODE models as DEVS: Quantization approaches. In Fernando Barros and Norbert Giambiasi, editors, *Proceedings of the AIS'2002 Conference (AI, Simulation and Planning in High Autonomy Systems)*, pages 163–169. Society for Modeling and Simulation International (SCS), April 2002.
- [3] Jean-Sébastien Bolduc and Hans Vangheluwe. Mapping ODEs to DEVS: Adaptive quantization. In *Proceedings of the 2003 Summer Simulation MultiConference (SCSC'03)*, pages 401–407, Montréal, Canada, July 2003.

- [4] Hyup J. Cho, Bernard P. Zeigler, and Jeong K. Kim. Analysis of the behavior of quantization: A renewal theory approach. In *Proceedings of the AIS'00 Conference (AI, Simulation & Planning in High Autonomy Systems)*, pages 305–316, Tucson, AZ, March 2000.
- [5] Ernst Hairer, Syvert Paul Nørsett, and Gerhard Wanner. *Solving Ordinary Differential Equations I: Nonstiff Problems*. Springer-Verlag, Berlin, second revised edition, 2000.
- [6] Hassan K. Khalil. *Nonlinear Systems*. Prentice Hall, Upper Saddle River, NJ, second edition, 1996.
- [7] Ernesto Kofman and Sergio Junco. Quantized-state systems: A DEVS approach for continuous system simulation. *Transactions of SCS*, 18(3):123–132, 2001.
- [8] Ernesto Kofman, Jong S. Lee, and Bernard P. Zeigler. DEVS representation of differential equation systems: Review of recent advances. In *In Proceedings of the 2001 European Simulation Symposium*, Marseille, France, October 2001.
- [9] Harbir Lamba and Andrew M. Stuart. Convergence proofs for numerical IVP software. Technical Report SCCM-98-05, Stanford University, CA, 1998.
- [10] Pieter J. Mosterman and Gautam Biswas. A hybrid modeling and simulation methodology for dynamic physical systems. *SIMULATION: Transactions of The Society for Modeling and Simulation International*, 78(1):5–17, January 2002.
- [11] Alexandre Muzy and Gabriel A. Wainer. Cell-DEVS quantization techniques in a fire spreading application. pages 542–548, San Diego, CA, USA, December 2002.
- [12] Olaf Stursberg, Stefan Kowalewski, and Sebastian Engell. On the generation of timed discrete approximations for continuous systems. *Mathematical and Computer Modelling of Dynamical Systems*, 6(1):51–70, 2000.
- [13] Gabriel A. Wainer and Bernard P. Zeigler. Experimental results of timed cell-DEVS quantization. Tucson, AZ, USA, 2000.
- [14] Bernard P. Zeigler, Herbert Praehofer, and Tag Gon Kim. *Theory of modeling and simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems*. Academic Press, San Diego, CA, second edition, 2000.

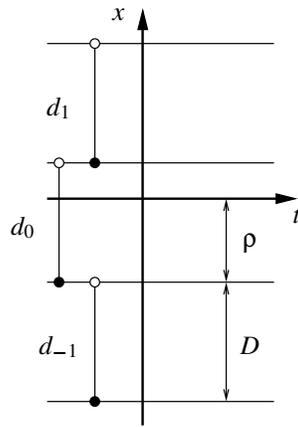


Figure 1: Quantization \mathcal{P} of \mathbb{R} .

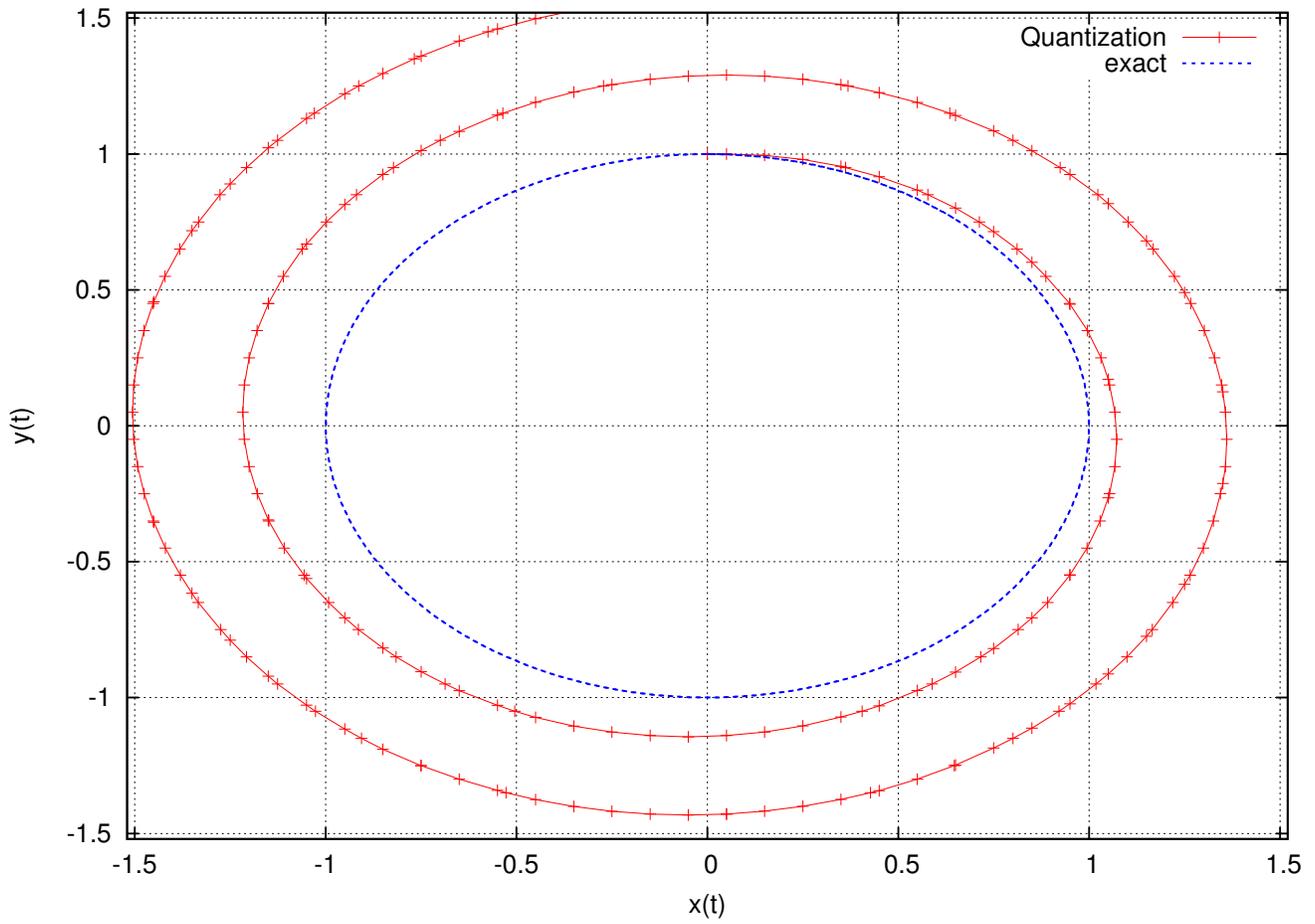


Figure 2: Circle test problem: $x_0 = 0, y_0 = 1; D = 0.1, \rho = 0.05$.

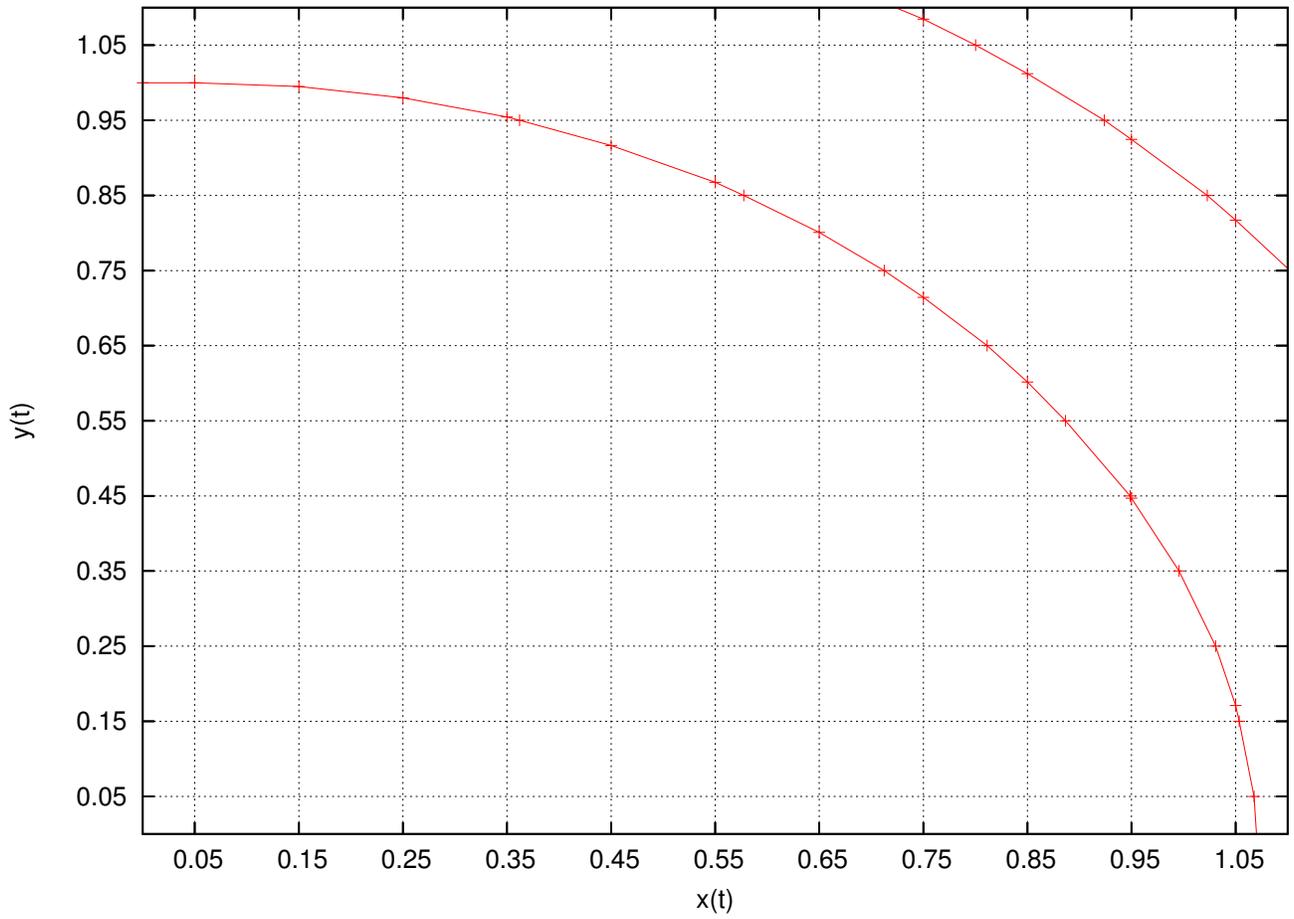


Figure 3: First quadrant of Figure 2, with quantization grid.

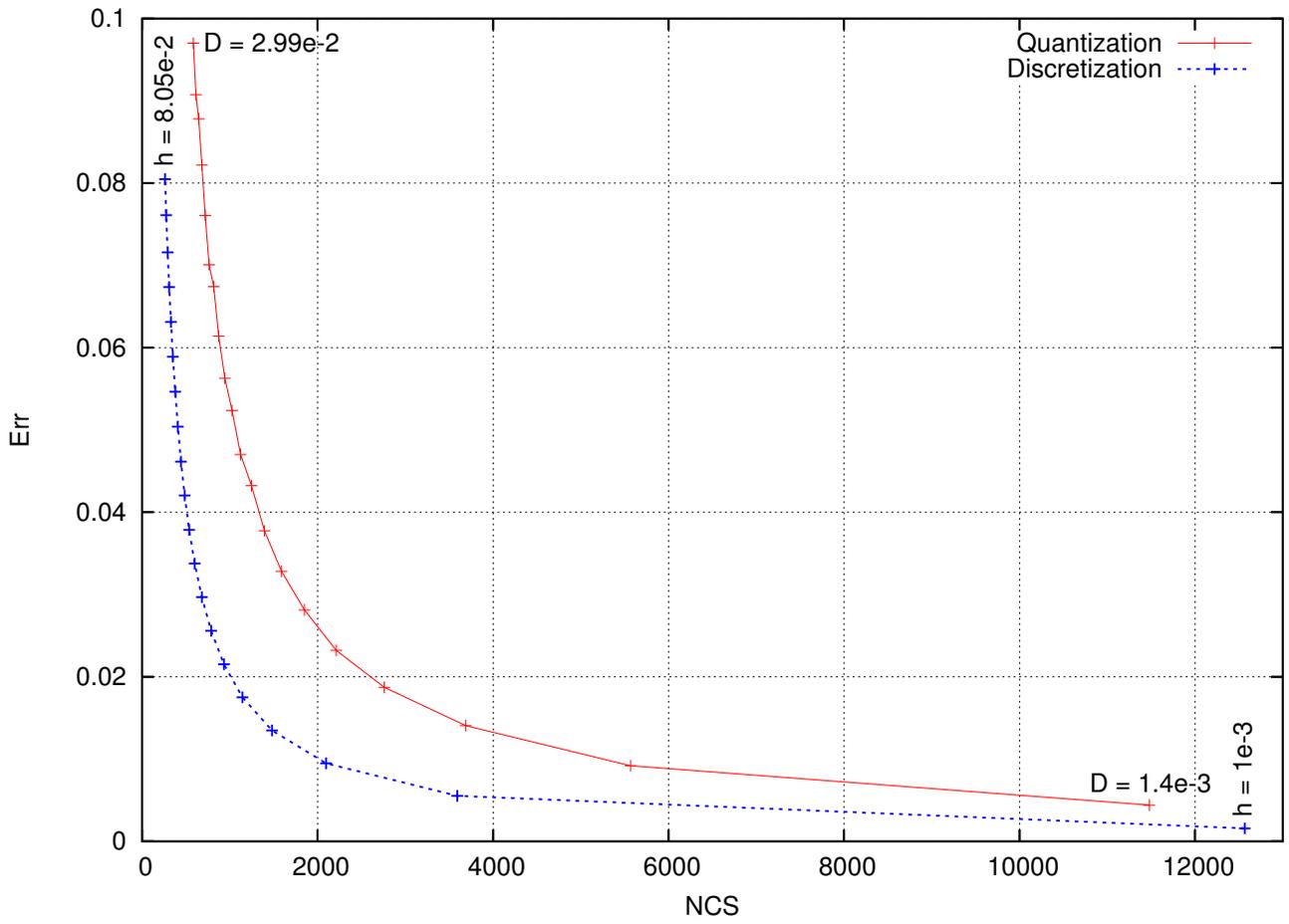


Figure 4: Circle test problem: performance comparison of nonadaptive Quantization and Discretization.

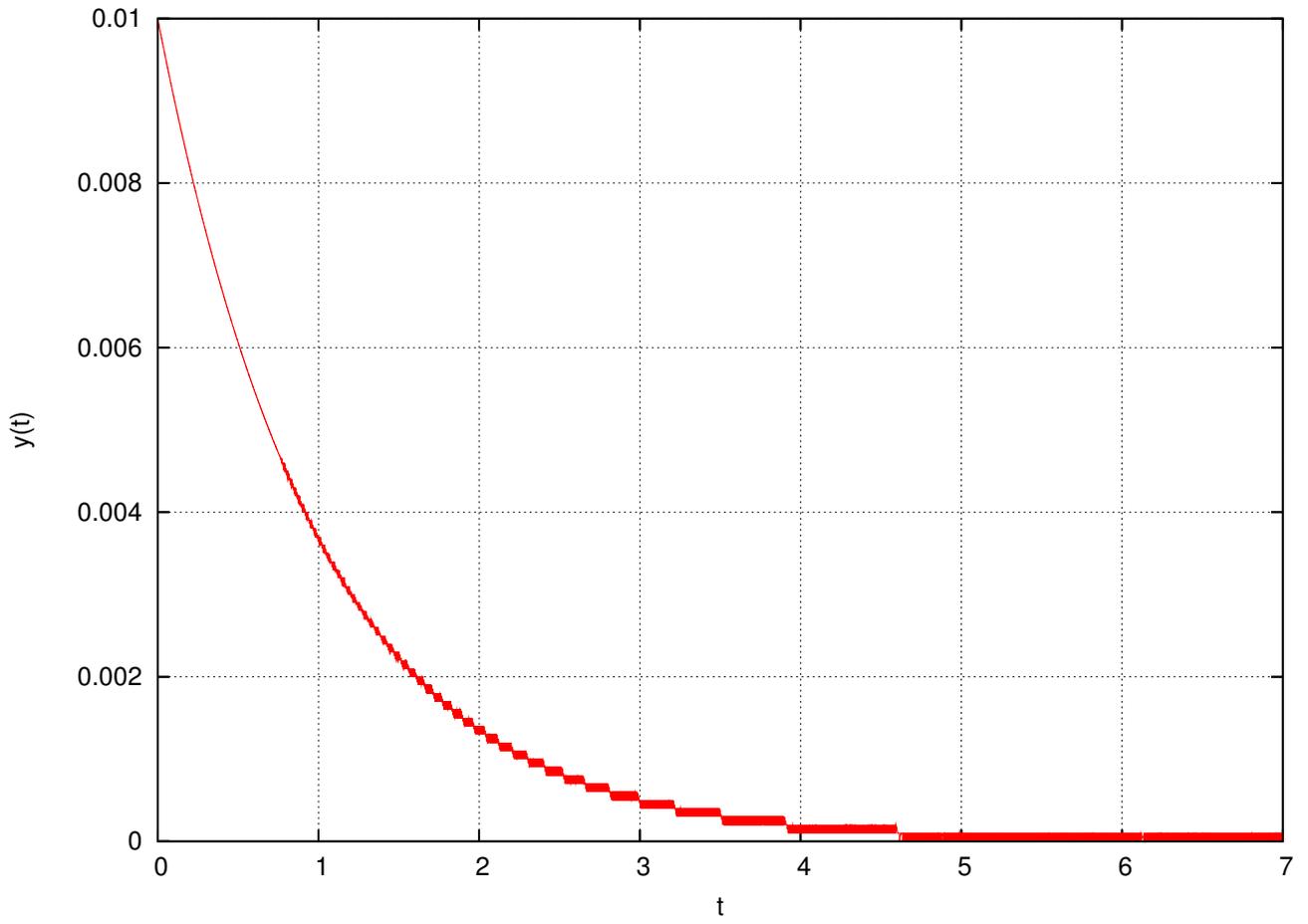


Figure 5: Stiff problem: $x_0 = 0, y_0 = 0; D = 1e - 4, \rho = 0$.

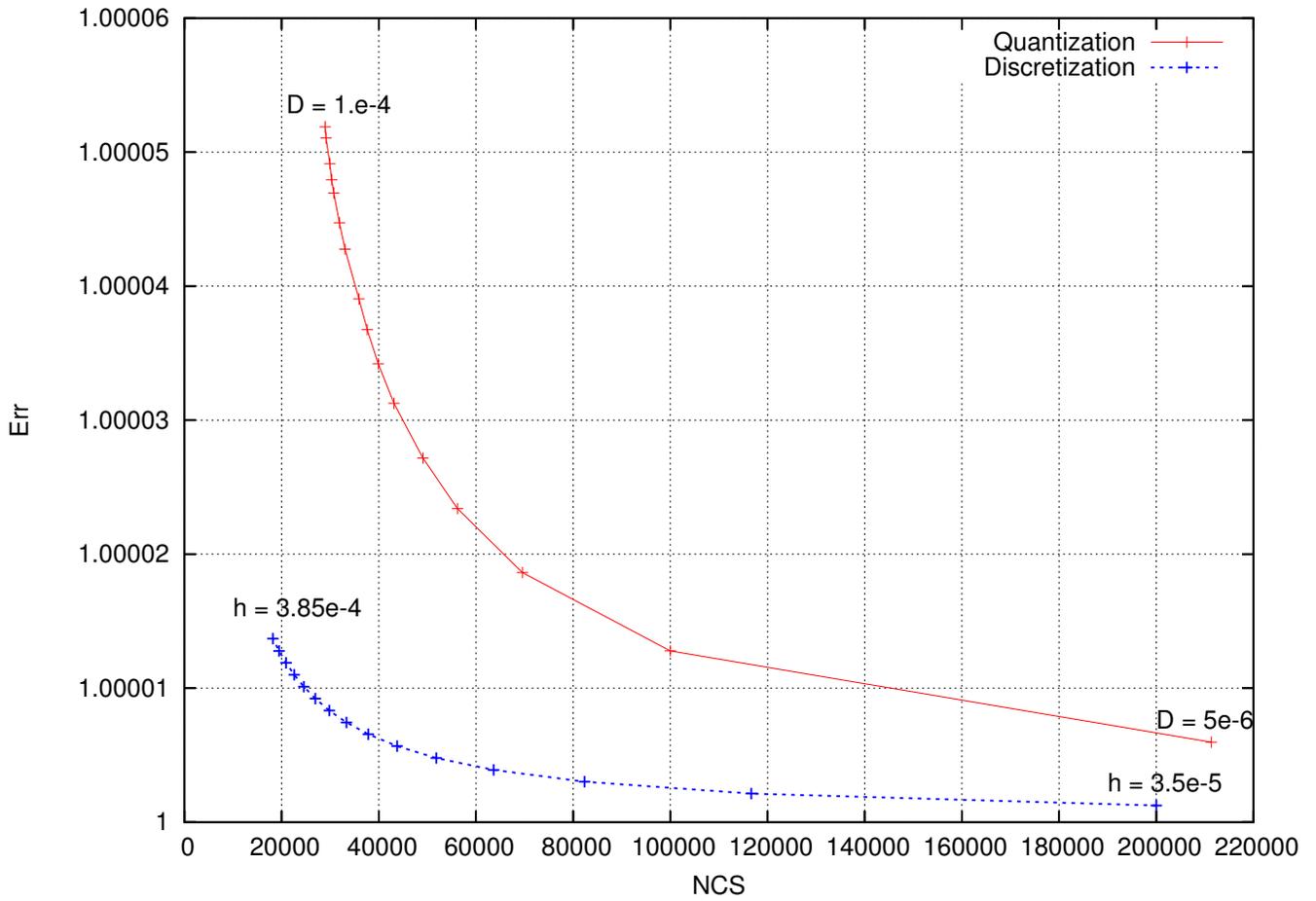


Figure 6: Stiff problem: performance comparison of nonadaptive Quantization and Discretization.

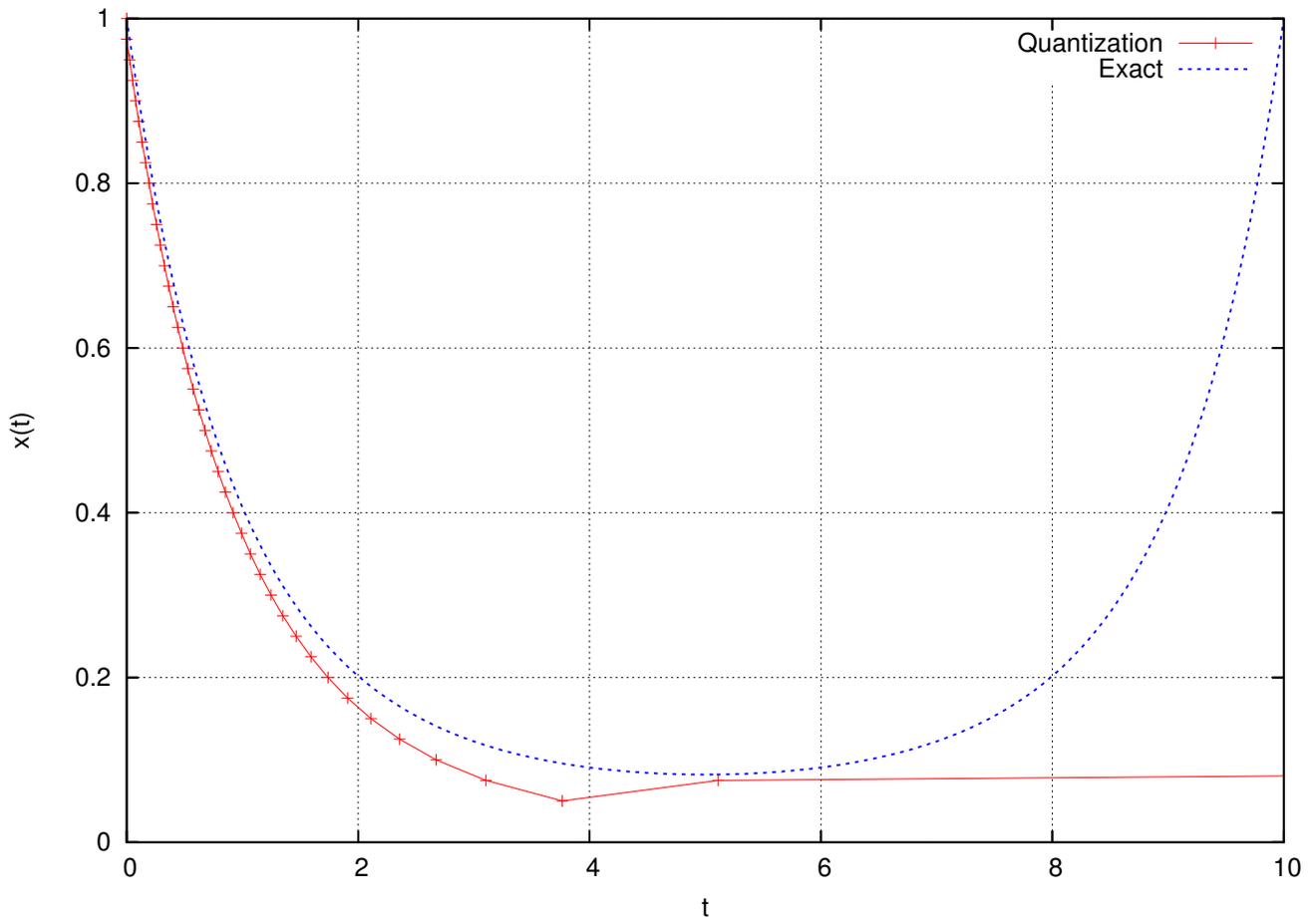


Figure 7: Nonautonomous problem: $x_0 = 1$; $D = 2.5e - 2$, $\rho = 0$.

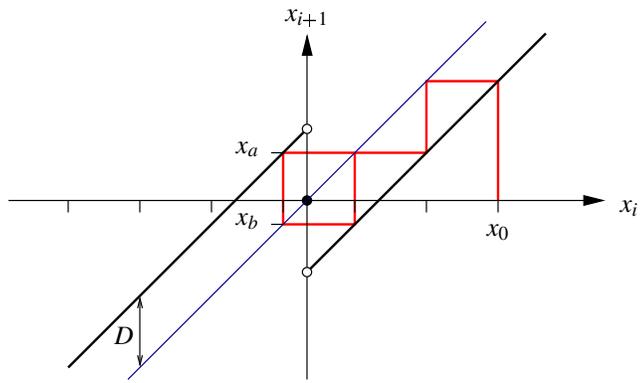


Figure 8: iterative map for the test problem

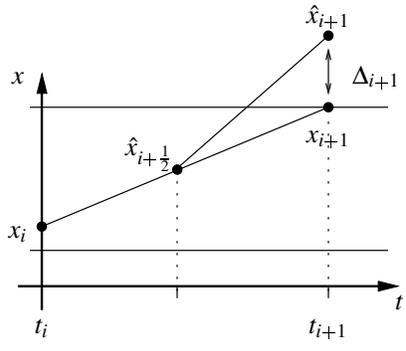


Figure 9: Approximation of local truncation error Δ_i with step halving.

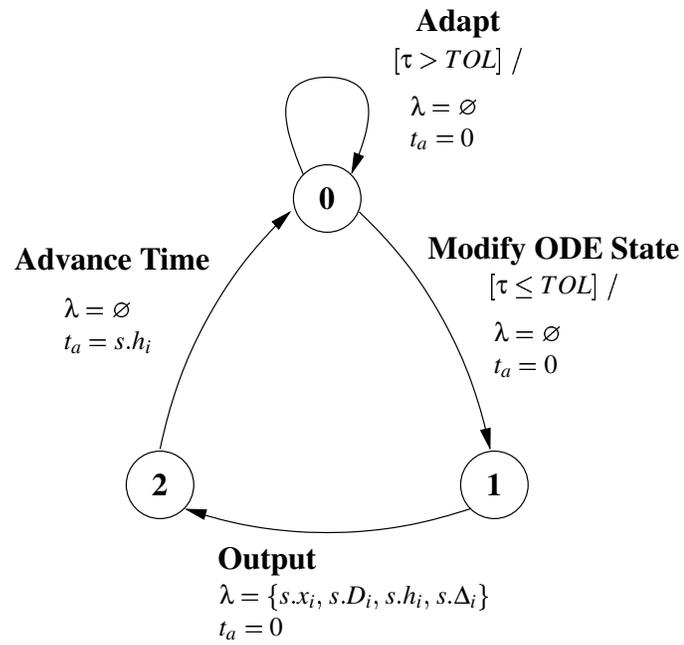


Figure 10: Dynamics of the atomic-DEVS.

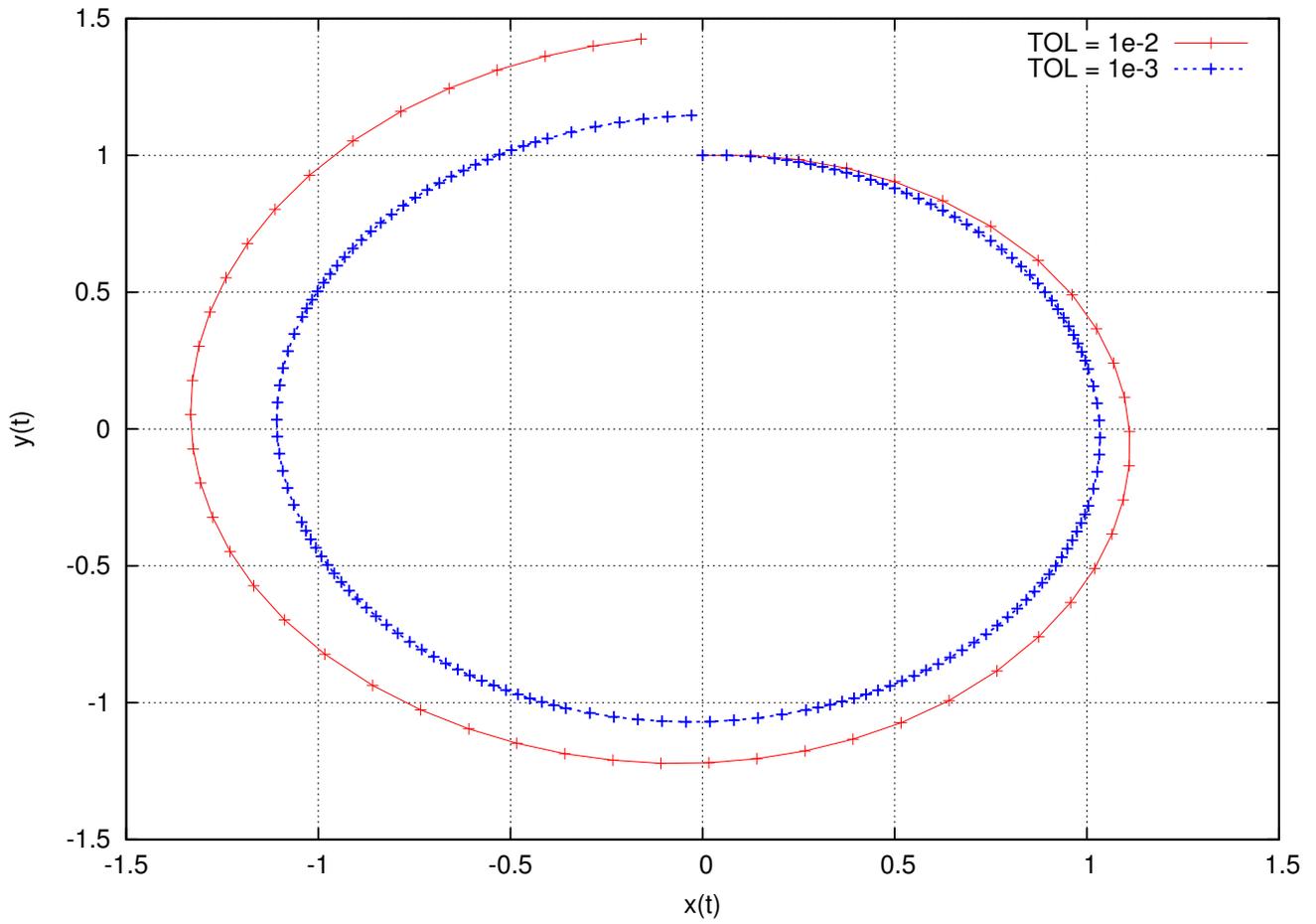


Figure 11: Circle test problem: $x_0 = 0, y_0 = 1; D_B = 1, \rho_B = 0$.

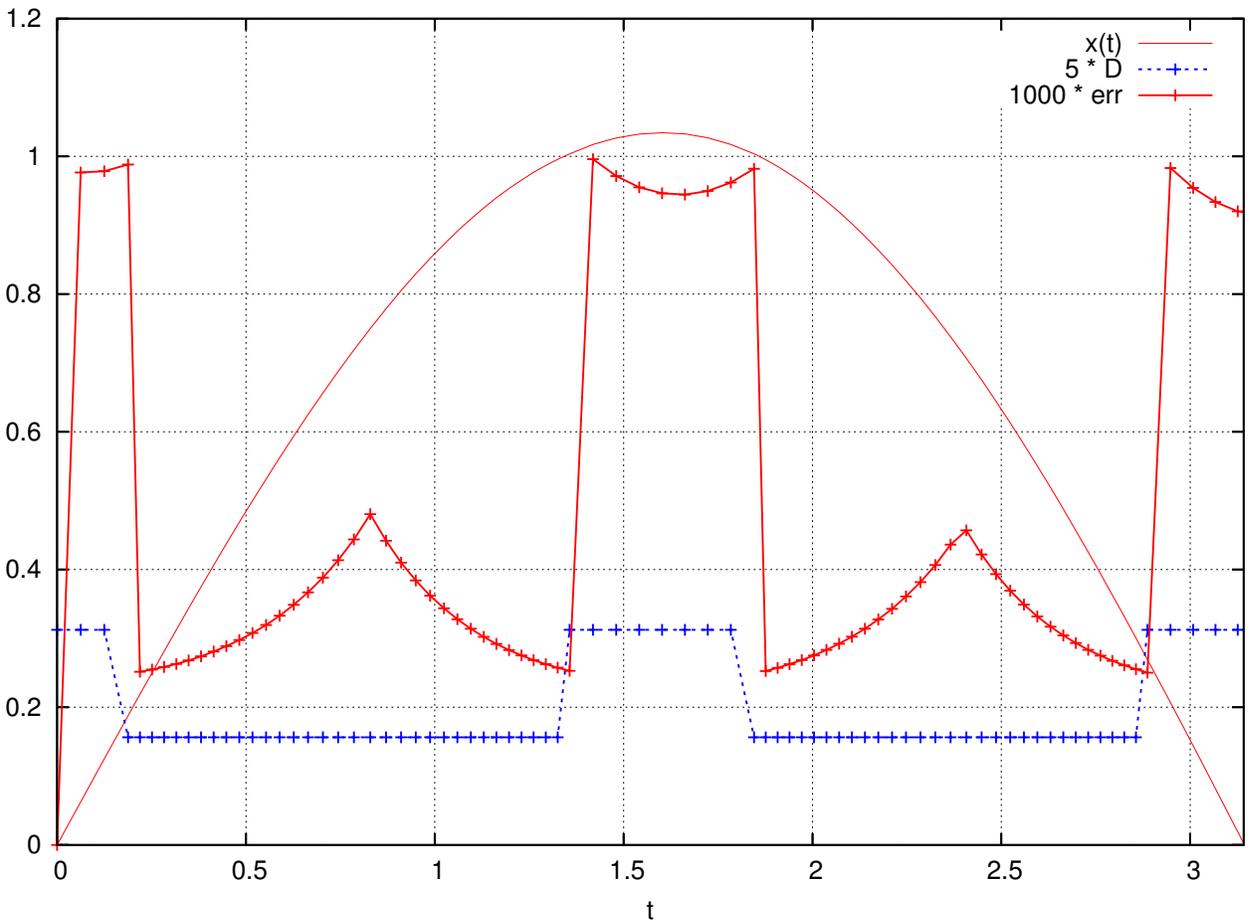


Figure 12: Circle test problem: evolution of quanta size and local truncation error ($TOL = 1e-3$).