Mathematical models of time

Conference Paper · September 2022

CITATIONS
0

READS
4

1 author:

Peter Junglas
PHWT Private University of Applied Sciences
51 PUBLICATIONS **220** CITATIONS

SEE PROFILE

Peter Junglas

Mathematical models of time

Abstract. Modeling and teaching of mathematics have close connections: Modelers profit from a broad mathematical background, while mathematics lecturers can use modeling examples for didactical purposes. This is examplified here with a number of models for time from different areas: Time models that are used in applications range from simple discrete sets or the real numbers, over pairs of numbers, to the hyperreals, which contain infinitesimals. The complications that arise, when models include the behaviour of the computer arithmetic, are discussed especially.

Introduction

For lecturers of mathematics, the field of modeling and simulation provides a plethora of examples from many mathematical areas in a huge range of applications. A quick glance at two standard textbooks [1, 2] shows, among many others, linear algebra in computer graphics applications, ordinary differential equations in vibration theory, Markov chains in traffic simulations or optimization methods for machine learning. Such examples can be used to motivate basic ideas of mathematical methods, provide interesting applications, or even can be starting points for the introduction of new mathematical abstractions.

Many of these applications are studying dynamical processes, where the time is an important variable. Every physicist will immediately agree that the modeling of time is a very fascinating subject. But even if one sticks to the classical Newtonian idea of time, without considering relativistic effects or a quantized space-time, one will find an amazing number of very different approaches to model time in a specific domain of application. An immediately obvious distinction exists between continuous and discrete time models, another important aspect is the contrast between abstract mathematical descriptions and more concrete models, that are concerned with the problems of computer numerics.

In the following we will present several different concepts that are used for the modeling of time in various application areas. We will always start with a concrete example, followed by a mathematical definition of the time model used, and sometimes add remarks about implementations

inside a concrete simulation environment. Some of these models could be useful to introduce new concepts in a mathematics lecture. But another point is at least equally important: For a concrete application, there are always many different mathematical abstractions, and a good choice can simplify the modeling or the simulation considerably.

Discrete time

The first model describes a three-year third level school, which is used as a simple example of a discrete state-space model in [3]. The interesting variables are the class sizes $x_i(k),\ i \in \{1, 2, 3\}$ at the beginning of year k, which are non-integer values interpreted as mean values over a number of years. The time evolution of this system is given by

$$\begin{aligned} x_1(k+1) &= x_{in}(k) + R_1\, x_1(k) \\ x_2(k+1) &= (1 - R_1 - D_1)\, x_1(k) + R_2\, x_2(k) \\ x_3(k+1) &= (1 - R_2 - D_2)\, x_2(k) + R_3\, x_3(k) \end{aligned}$$

with constant ratios R_i of pupils that repeat a class and D_i of dropouts, and a given number $x_{in}(k)$ of incoming pupils starting in class 1.

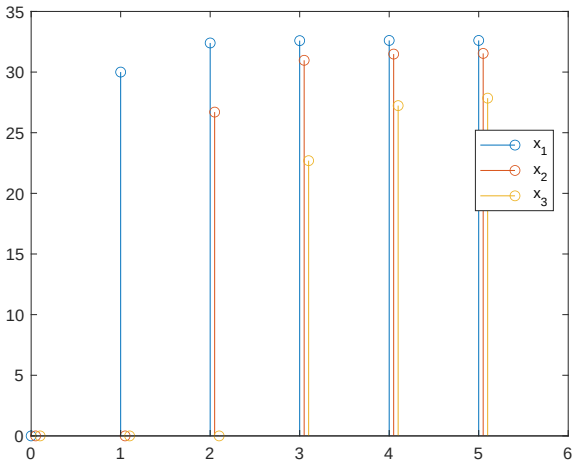

Figure 1: Typical time behaviour of the school model.

A typical time trajectory is shown in Figure 1, where all class sizes are defined only at the beginning of a year (but shown slightly displaced for clarity). Often such variables are plotted as in Figure 2, where one assumes that the class sizes are defined during the whole year by constant

continuation. This often leads to a lack of preciseness, when the exact meaning at the jump points is not clearly denoted.

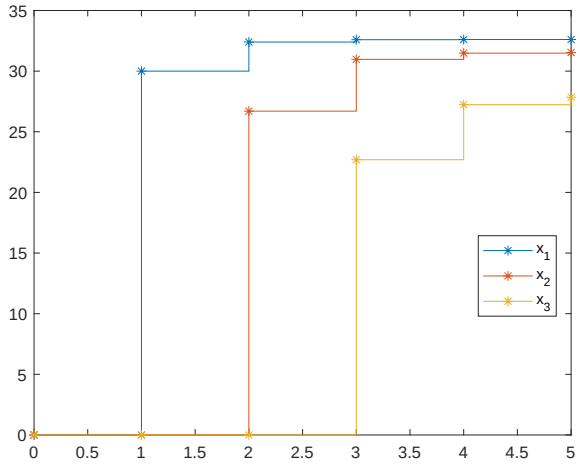

Figure 2: Alternative graphical representation of Fig. 1.

The time model used here is given by a fixed time interval $\Delta t \in \mathbb{R}^{>0}$ and the definition

$$t \in \mathcal{T}_1 := \{n\,\Delta t \,|\, n \in \mathbb{N}\}$$

Here as in the following $\mathbb{N}$ means the set of natural numbers including 0. The definition of the time model $\mathcal{T}_1$ thus allows the inclusion of initial values for the state variables.

The computer simulation of the time values is not completely trivial, since the computer arithmetic only works with a finite subset $\mathbb{FP}$ of the real numbers, and a given number $x \in \mathbb{R}$ is mapped to its floating-point representation $fp(x) \in \mathbb{FP}$. This can lead to arithmetical problems such as

$$fp(n\Delta t) \oplus fp(\Delta t) \neq fp((n+1)\Delta t)$$

for a given natural number n and the floating-point addition $\oplus$. A simple workaround is to use only integer values for the time steps and multiply with the time interval Δt at the very end of a computation – if at all.

Continuous time

The next model is the well-known mathematical pendulum, described by its angle $\varphi(t)$ and the corresponding angular velocity $\omega(t) = \dot{\varphi}(t)$. Its

movement is given by

$$\ddot{\varphi} + \frac{k}{m}\dot{\varphi} + \frac{g}{l}\sin\varphi = 0$$

and initial conditions

$$\varphi(0) = \varphi_0$$
$$\omega(0) = \omega_0,$$

where g ist the earth acceleration, l the length of the string, m the mass of the pendulum and k a damping constant. A typical trajectory is shown in Figure 3.

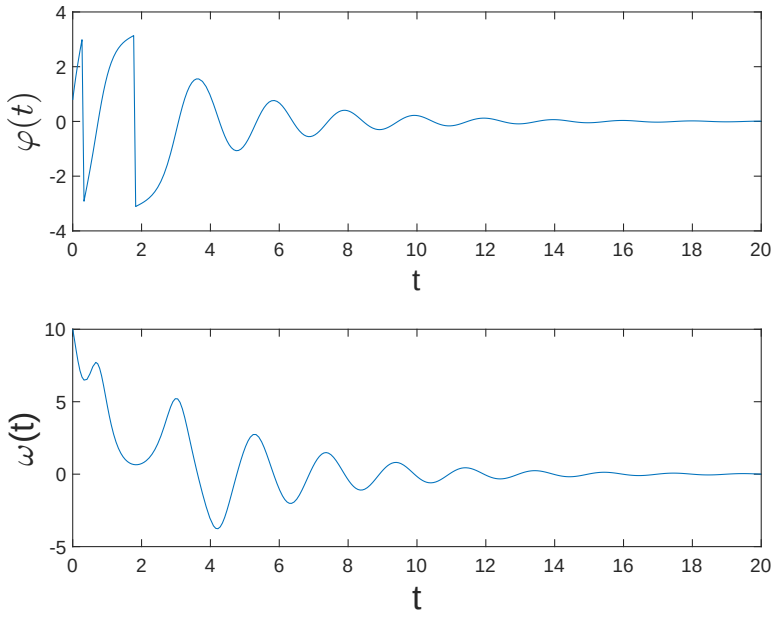

Figure 3: Trajectory of a mathematical pendulum.

The time model is

$$t \in \mathcal{T}_2 := \mathbb{R}^{\geq 0},$$

which is the standard description of a continuous time in most applications. From a mathematical point of view, $\mathbb{R}$ as a linearly ordered field and complete metric space is a natural choice. Nevertheless, it is far from trivial, as its standard construction – as equivalence classes of Cauchy sequences of rational numbers or as Dedeking cuts – clearly shows. In applications, this is not a problem, since the properties of $\mathbb{R}$ are well known and its existence is taken for granted.

For the simulation of this model, usually a standard ODE solver with adaptive step size is employed. As a consequence, only a discrete subset of $\mathbb{R}$ is used. The proposed QSS ("Quantized State System") solvers [4] use a discretization of the state variables instead of the time, a bit similar to the difference between Riemann and Lebesgue integral. This leads to trajectories that are piecewise constant, so that the times of state changes again form a discrete set.

Taking into account the properties of computer arithmetics, the corresponding model is $\mathbb{FP}$ with its slightly different addition $\oplus$ and multiplication $\otimes$. Trying to prove the correctness of an algorithm based on $\mathbb{FP}$ is quite difficult: Operations that lead to equal results with real arithmetic, can give different results in $\mathbb{FP}$ – even the order of computed values can change in the floating-point approximation. In order to guarantee expected results in computer computations, a different time model has been introduced in [5] that only uses integer values, basically to define rational numbers. More precisely, they use the set $\mathbb{I} \subset \mathbb{Z}$ of numbers of type Integer to define time values as $t \hat{=} (n, d, e) \in \mathbb{I}^3$, where the value of such a triple is given by

$$t = \frac{n}{d} \times 2^e$$

This allows to combine fast implementations with simple provable properties.

We see that in the context of computer computations, using $\mathbb{R}$ as time model is neither obvious nor simple. Nevertheless, it is often used, when no numerical problems due to the floating-point arithmetic are expected, and will be the starting point in the following examples.

Continuous time with events

The third model is a ball falling freely under gravity, until it hits the ground, where it is reflected, loosing energy in the process. In addition to the position $x(t)$ and velocity $v(t)$ of the ball, interesting variables are the time t_i of the i-th ground contact or the number $n(t)$ of collisions that have occured until time t. With the earth acceleration g and loss factor

$q < 1$ per collision, the behaviour is given by

$$\begin{aligned} \dot{x} &= v \\ \dot{v} &= -g \\ x(0) &= x_0 > 0 \\ v(0) &= v_0 \\ x(t_i) &= 0 \quad \Rightarrow \quad \text{restart with } x(t_i) = 0,\ v(t_i) = -q\, v(t_i^-). \end{aligned}$$

When the ball hits the ground, the integration of the ODE is stopped and restarted with new values for x and v. This is called an "event". A typical trajectory is shown in Figure 4, where the event times are marked by red stars.

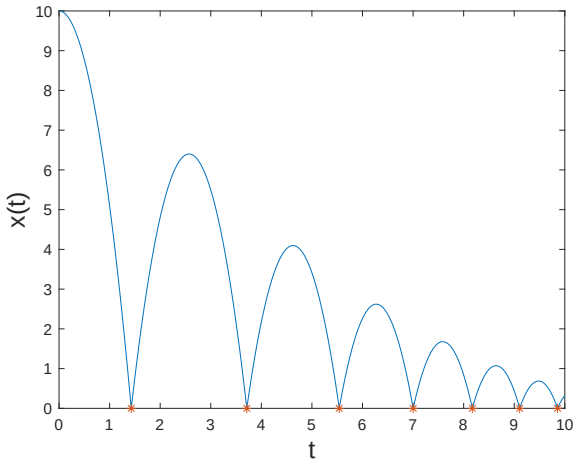

Figure 4: Trajectory of the bouncing ball.

To combine the continuous time flow and the times of the events, the notion of "hybrid time" has been introduced in [6]. The corresponding model is described by

$$\begin{aligned} t \in \mathcal{T}_3 \ := \ \big\{(s,j) \,|\, & s \in \mathbb{R}^{\geq 0} \wedge j \in \mathbb{N} \\ & \wedge \left((s',j') \in \mathcal{T}_3 \wedge j > j' \Rightarrow s \geq s'\right)\big\} \end{aligned}$$

Less formally, this means: Time is given by a pair (s, j), where s describes the ordinary continuous time, while j denotes the number of the last event. The event numbers are ordered according to their time. In principal – though not in our concrete example – it is possible that several events happen at the same time. Such an example of a hybrid time set is shown

in Figure 5. Figure 6 displays a trajectory for the bouncing ball over hybrid time.

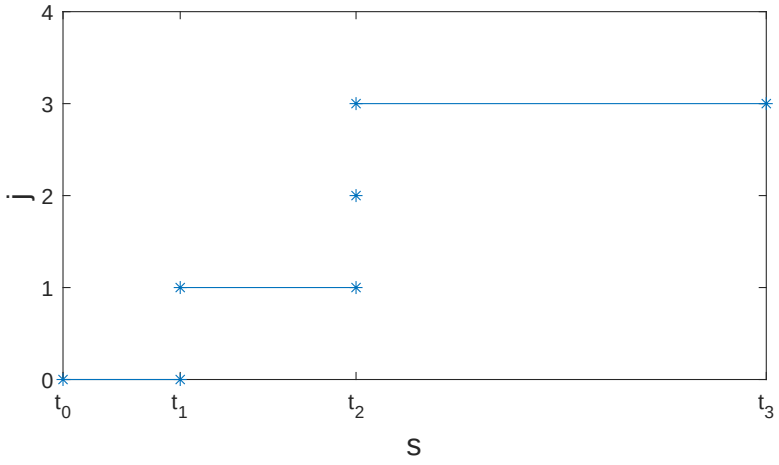

Figure 5: Example of a hybrid time set.

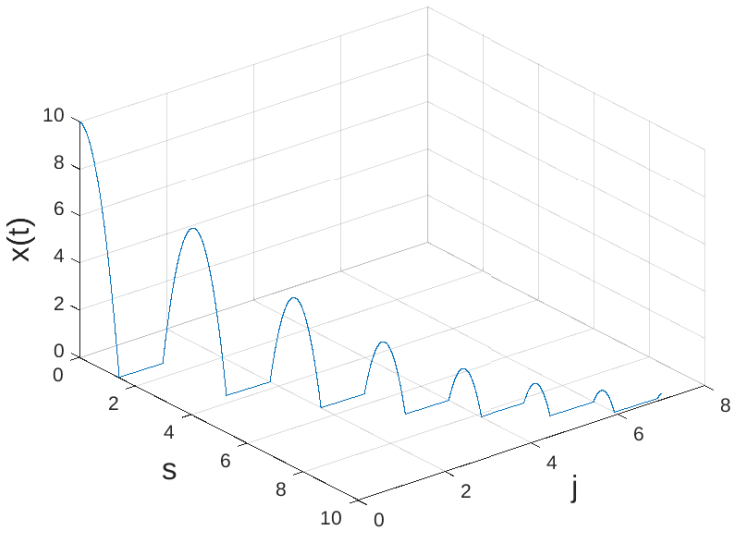

Figure 6: Bouncing ball trajectory over hybrid time.

This model shows a very strange behaviour, the so-called "Zenon effect": Since the length of the time intervals between events decreases geometrically, the model has an infinite number of events in a finite time. After the limit point t^* of these events, its behaviour is not defined by the equations above, but of course is set to $x(t) = 0$ for $t > t^*$. Figure 7 shows the Zenon effect over the usual time coordinate, while Figure 8 displays x over the hybrid time plane.

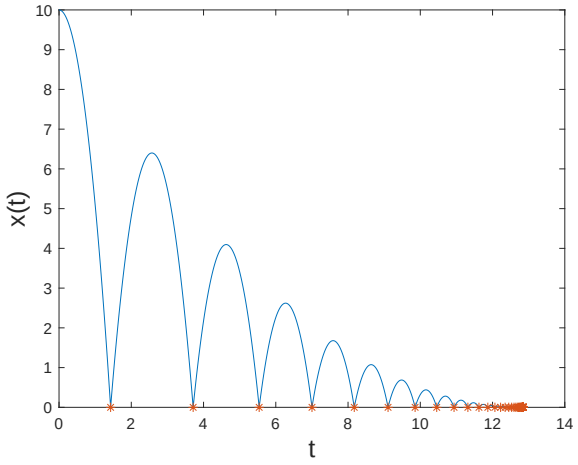

Figure 7: Zenon effect shown with usual time coordinate.

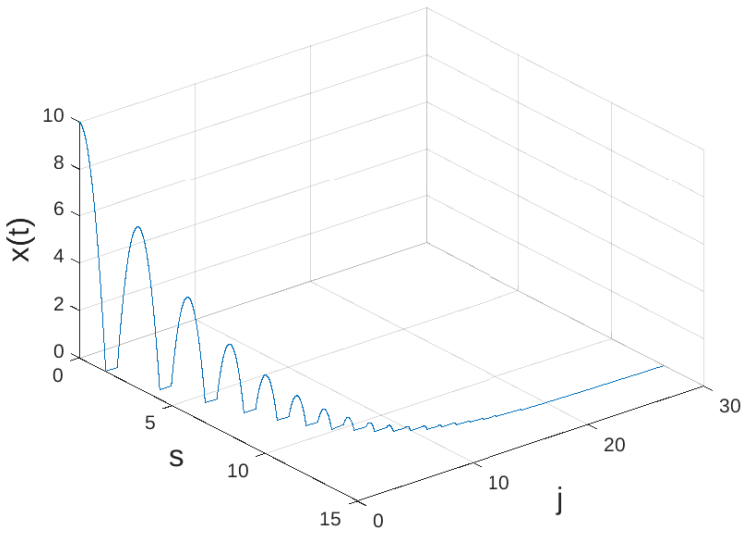

Figure 8: Zenon effect shown over hybrid time.

Discrete events I

A fundamental example for discrete event modeling is the simple queueing system. It consists of a generator G that creates entities in fixed time intervals t_G and sends them to a queue Q, which is connected to a server S with fixed service time t_S. Entities leaving the server are terminated at T (cf. Figure 9). At their creation the entities are provided with internal consecutive numbers ("ids") starting with 1. A mathematical description of the behaviour of the components and the complete system can be done using the PDEVS formalism [7].

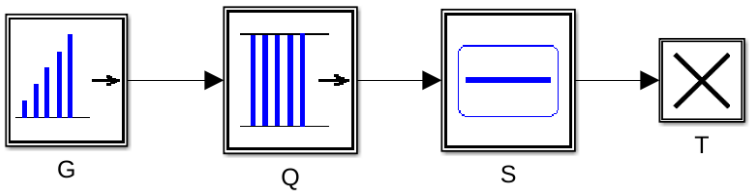

Figure 9: Simple queueing system.

The description of the system behaviour is given by the list of all events, such as arrivals or departures of an entity at a component, together with their time, often in a tabular form. A simple example using the values $t_G = 1$ and $t_S = 1.5$ is shown in the following table that displays the arrival of entities given by their id at the components:

t	G	Q	S	T
1.0	1			
1.0		1		
1.0			1	
2.0	2			
2.0		2		
2.5				1
2.5			2	
3.0	3			
3.0		3		
4.0	4			2
4.0			3	
4.0		4		

Often, several events happen at the same time. In some cases the order of such events may be fixed, as for the first three events, where entity 1 moves from G to Q to S. Such events are called "causally connected". In other cases the order is arbitrary, e. g. when at $t = 4$ entity 2 leaves S and entity 4 is generated in G. In such cases a concrete model can either define a fixed order, an arbitrary order (given randomly) or a real parallel occurence.

Generally, one is interested in component-related data such as the number of entities in the queue ("queue length") $q(t)$ over time. Figure 10 shows the corresponding graph, together with the ids of the departing entities at the time of their departure.

The time model used here is basically $\mathbb{R}^{\geq 0}$, but in a concrete model

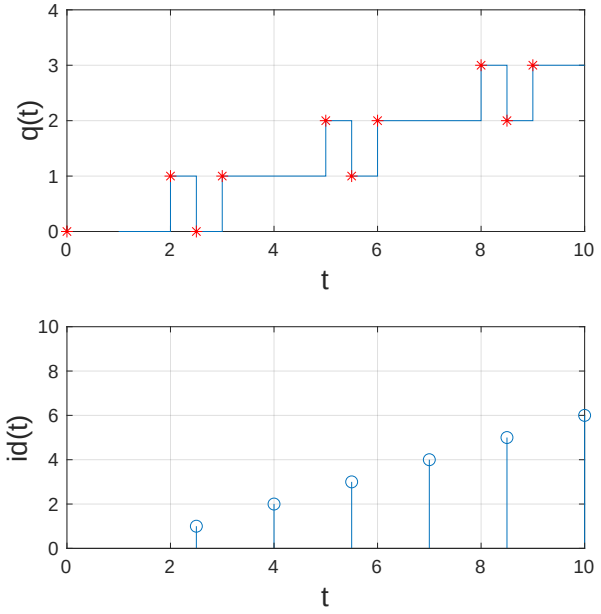

Figure 10: Queue length and departure times in the queue-server model.

only a discrete set of time values are possible:

$$t \in \mathcal{T}_4 := \{t_0, t_1, ..., t_N\} \subset \mathbb{R}^{\geq 0} \quad \text{with } i > j \Rightarrow t_i \geq t_j$$

The possible values t_i result from the dynamic behaviour of the model, strictly speaking they are functions of the model. In the PDEVS formalism times are given explicitely by a "time advance function" ta and implicitely by the connection of the components. The interesting functions are only defined for $t \in \mathcal{T}_4$, but are often (e. g. for $q(t)$) constantly extended in between. In many applications the constant values for t_G and t_S are replaced by random values according to given probability distributions. In these cases all t_i are random variables, which makes the formal definition of the time model much more involved.

Discrete events II

We will still study the queue-server model, but add an implementation detail: When the server is busy, the queue must not send entities. Therefore the server sends a blocking signal to the queue to inform it about its state (cf. Figure 11). The queue sets its own state to "Free" or "Blocked" accordingly.

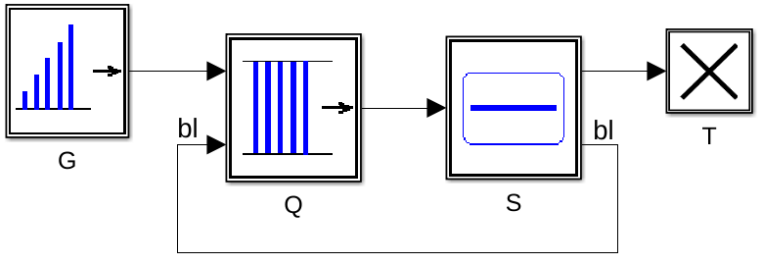

Figure 11: Queueing system with blocking signal.

The more explixit system behaviour is now given by the following table, which is extended by a column Q_S for the queue state ("F" or "B") and a corresponding value ("f" or "b") for the signal from the server:

t	G	Q	Q_S	S	T
1.0	1		F		
1.0		1	F		
1.0			F	1	
1.0		b	F		
1.0			B		
2.0	2		B		
2.0		2	B		
2.5			B		1
2.5		f	B		
2.5			F		
2.5			F	2	
2.5		b	F		
2.5			B		
3.0	3		B		

The behaviour has become much more complicated: More events occur at the same time, but the order of events is very important. Especially the incoming blocking signal from the server has to have precedence over other events to prevent the queue from sending entities to a busy server.

In order to be able to easily sort concurrent events, the notion of "superdense time" has been introduced in [8]:

$$t \in \mathcal{T}_5 := \{(s, j) \,|\, s \in \mathcal{T}_4 \wedge j \in \mathbb{N}\}$$

Time values are again given by pairs. The additional number j defines the order of concurrent events during one time instant, it starts with 0 for each discrete time value s.

Discrete events III

For the final time model we will again use the queue-server system with blocking signal and try to solve a basic problem with superdense time: Who orders the concurrent events? In a PDEVS environment this has to be defined in the so called "confluent function", which sometimes is a difficult and error-prone endeavour.

A solution that is based on yet another time model is proposed in [9]. It starts from the observation that concurrent events are actually due to an oversimplification: Every signal transport and every state change is always connected with a small time delay. But to include these delays in a model would introduce a plethora of small parameters, which are not known and whose values are not really needed – as long as they are small and larger than zero. An obvious solution (at least for a physicist) is the introduction of an infinitesimal time delay ε for all transports and formerly immediate state changes. The corresponding behaviour of such a model is given by the following table:

t	G	Q	Q_S	S	T
1.0	1		F		
$1.0+\varepsilon$		1	F		
$1.0+2\varepsilon$			F	1	
$1.0+3\varepsilon$		b	F		
$1.0+4\varepsilon$			B		
2.0	2		B		
$2.0+\varepsilon$		2	B		
2.5			B		1
$2.5+\varepsilon$		f	B		
$2.5+2\varepsilon$			F		
$2.5+3\varepsilon$			F	2	
$2.5+4\varepsilon$		b	F		
$2.5+5\varepsilon$			B		
3.0	3		B		

For a physicist, this approach might seem reasonable, but many mathematicians are sure that the use of "infinitesimals" is impossible in a mathematically precise way. And yet, infinitesimals can be introduced in a sound way, as has been proven by the explicit construction of the hyperreal numbers ${}^*\mathbb{R}$ and the introduction of non-standard analysis [10].

For conveniance, the most important properties of ${}^*\mathbb{R}$ will be recapitulated here: ${}^*\mathbb{R}$ is a linearly ordered field, which is not order complete. It contains $\mathbb{R}$ as a subset, as well as an infinitesimal number $\varepsilon > 0$, which is smaller than any positive real number. Being a field, it then contains numbers such as 2ε, $-\varepsilon^2$ and non-finite values $\omega := 1/\varepsilon$ and ω^2. A useful theorem states that every finite element $h \in {}^*\mathbb{R}$ can be written uniquely as

$$h = r + \delta$$

with real r and infinitesimal δ. In this case, r is called the "standard part" $st(h)$ of h. The construction of ${}^*\mathbb{R}$ is very sophisticated, it uses ultrafilters and Zorn's Lemma. Its elements are equivalence classes of sequences of real numbers, but as with the construction of $\mathbb{R}$, the basic properties are all one needs, once the existence is established.

Using the hyperreals, the time model is a simple extension of $\mathcal{T}_4$:

$$t \in \mathcal{T}_6 := \{t_0, t_1, ..., t_N\} \subset {}^*\mathbb{R} \quad \text{with } i > j \Rightarrow t_i \geq t_j$$

For a concrete implementation in a simulation program, numbers of the form $t = a + b\,\varepsilon$ with $a, b \in \mathbb{FP}$ are sufficient. The infinitesimal parts are used internally in the simulator to dynamically order formerly concurrent events. For user outputs all time values are reduced to their standard parts, a typical output graph is identical to Figure 10. For debugging purposes one can replace ε with a small real time interval to make the internal order of events visible. The corresponding graph is shown in Figure 12.

Conclusions

The examples have shown that the choice of a time model is non-trivial and depends on the requirements of the system under study. A broad variety of models can be used, ranging from very simple ones ($\mathcal{T}_1$, $\mathcal{T}_2$, $\mathcal{T}_4$) or unexpected ones ($\mathcal{T}_3$, $\mathcal{T}_5$) to quite heavyweight ones ($\mathcal{T}_6$). If one introduces random numbers, the models get more complex, and even more so, if one includes models of the computer arithmetic.

For teachers or users of mathematics, there are a few insights following from these examples:

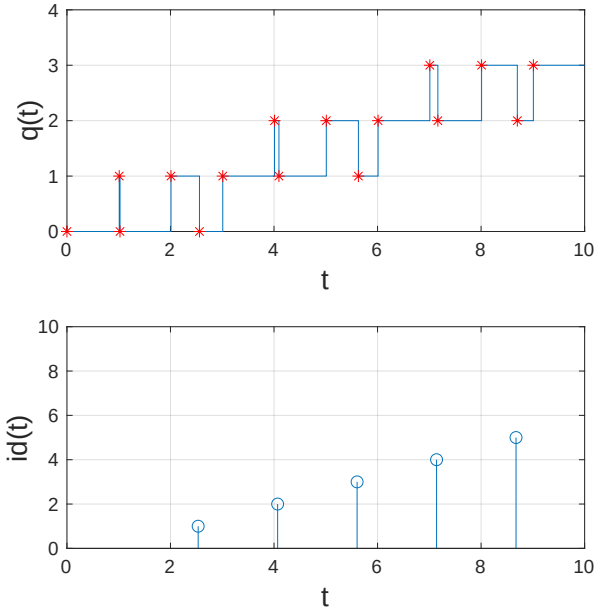

Figure 12: Queue length and departure times in debugging mode.

- $\mathbb{R}$ is not the only model of time, not even the simplest or best one in many applications.
- Many mathematical models of time have very different properties than their real-life computer implementations.
- Hyperreal numbers are useful. Though their construction is difficult, their use is not, and their properties are easy to understand. The question, whether introductory analysis courses should be based on ${}^*\mathbb{R}$, is still open.

Bibliography

[1] **Bungartz, H.-J., Zimmer, S.; Buchholz, M.; Pflüger, D.**: *Modellbildung und Simulation: Eine anwendungsorientierte Einführung.* Springer-Verlag Berlin Heidelberg, 2th ed. (2013).

[2] **Stickler, B., Schachinger, E.**: *Basic concepts in computational physics.* Springer-Verlag Switzerland, 2th ed. (2016).

[3] **Junglas, P.**: *Probabilistic state space models – A theoretical framework with practical relevance.* SNE Simulation News Europe, 1, **29**, 33–38 (2019).

[4] **Kofman, E.; Junco, S.**: *Quantized-state systems: a DEVS Approach for continuous system simulation.* Trans. Soc. Modeling and Simulation Int., 3, **18**, 123-132 (2001).

[5] **Vicino, D.; Dalle, O.; Wainer, G.**: *A data type for discretized time representation in DEVS.* In: Proc. 7th Int. Conf. on Simulation Tools and Techniques, Lisbon, Portugal (2014).

[6] **Sanfelice, R.**: *Analysis and design of cyber-physical systems: a hybrid control systems approach.* In: Cyber-Physical Systems, CRC Press, 3-31 (2015).

[7] **Zeigler, B., Muzy, A.; Kofman, E.**: *Theory of Modeling and Simulation.* Academic Press San Diego, 3rd ed. (2019).

[8] **Sarjoughian, H.; Sundaramoorthi, S.**: *Superdense time trajectories for DEVS simulation models.* In: Proc. Symposium on Theory of Modeling & Simulation: DEVS Integrative M&S Symposium, San Diego, USA, 249–256 (2014).

[9] **Junglas, P.**: *NSA-DEVS: Combining Mealy Behaviour and Causality.* SNE Simulation News Europe, 2, **31**, 73–80 (2021).

[10] **Goldblatt, R.**: *Lectures on the Hyperreals.* Springer New York (1998).

Author

Prof. Dr. rer. nat. Peter Junglas
Private Hochschule für Wirtschaft und Technik Vechta/Diepholz
Am Campus 2
D-49356 Diepholz
E-Mail: peter@peter-junglas.de