

A COMPARISON OF THE ACCURACY OF DISCRETE EVENT AND DISCRETE TIME

Arnold Buss
Ahmed Al Rowaei

MOVES Institute
Naval Postgraduate School
700 Dyer Road
Monterey, CA 93943 USA

ABSTRACT

Many combat and agent-based models use time-step as their simulation time advance mechanism. Since time discretization is known to affect the results when numerically solving differential equations, it stands to reason that it might likewise affect the results of such simulations. This paper demonstrates that is indeed the case. Using simple queueing models, we demonstrate that the size of the time step can have a substantial impact on estimated measures of performance. While large time steps can execute faster than a corresponding discrete event model, there can be substantial errors in the estimates. Conversely, with small time steps the results match both the discrete event measures as well as the analytic values, but can take substantially longer to execute.

1 INTRODUCTION

Time advance mechanisms for simulations are divided between next-event, for discrete event simulation (DES) models (Law and Kelton 2000) and time-step, or discrete time simulation (DTS), commonly used in military combat models and agent-based models. Time-step is also the preferred mechanism for continuous simulations, that is those involving differential equations. The time advance method used for a particular model is often nearly invisible to the modeler and hardly ever questioned.

It turns out that the DTS approach introduces an additional “parameter” into a model, namely the size of the time step, that can have a substantial impact on the results of the simulation. In most applications using DTS, the model does not allow changing the size of the time step, and thus introduces the possibility that the results have unknowingly been affected. When a time step model is used, there is a possibility that the results are in fact in large part due to the size of the time step rather than the inherent characteristics of the simulation model. To our knowledge there has been no systematic investigation of the comparative effects of different size time steps in such simulations, nor have there been any extensive studies of the differences between models constructed with one or the other approach.

In this paper we will investigate these differences for a few simple cases. Specifically, we will study the impact of the size of the time step on the results as well on the execution time. The results of time step models will also be compared to corresponding DES models. It is important to note that the objectives of this paper are not to find optimal or even “good” values of time steps in simulation models; rather, as the results of a few simple examples suggest, the use of time step advance in simulation models is fraught with danger. The remedy is to use next-event time advance instead, since it appears to not have the difficulties associated with time step. The literature on numerical solution to differential equations is vast, and it should be noted that the use of a simple stiff equation with the Euler method is only intended as a motivating example. There are many superior approaches to solving differential equations; however, only the

Euler method has direct analogy with time step simulation models in practice. To the knowledge of the authors, neither time step simulation models nor agent-based models incorporate such more sophisticated methods. Nor do any appear to include such features as adaptive time steps, another common numerical method. Although potentially harmful effects of the size of the time step is well-known in the numerical methods literature, there has been little work in traditional simulation modeling domains, such as queueing systems. What has been done appears to confirm the fact that the use of time steps affects the accuracy of the results (e.g. Park and Fishwick 2008).

First we will discuss the two different approaches to time in the next section, followed by a motivating example from differential equations. After that, we empirically investigate different outcomes as a function of the time advance mechanism and, in the case of DTS, the size of the time step. We finish with conclusions and future work.

2 TIME STEP AND NEXT EVENT APPROACHES TO TIME ADVANCE

In this section we will briefly present and discuss each of the two time advance methods studied in this paper. Each approach models the system in question as a collection of state variables, possibly including “entities” used to represent individual customers, jobs, etc.

The time step approach, or discrete time simulation (DTS), is the most straightforward and simple way for a simulation model to advance time, which perhaps explains its popularity in many application areas such as agent-based models. There is a fixed quantity, Δt , that is the increment at which time is advanced. The first time is advanced by Δt , then every state variable (and entity if used) is updated according to the logic defined by the model. The time-step method is used in many agent-based frameworks (e.g. MANA, Pythagoras) as well as many combat simulations (e.g. JANUS, COMBAT XXI).

The next-event method starts by restricting state variables to have piecewise constant trajectories, and then identifying the state transitions with the (instantaneous) “jumps” made by them. Since these state transitions can occur at any time, and since all states are constant from any current simulation time until the time of the earliest state transition, time can advance to that earliest state transition time, that is to the “next event.” Thus, time advances in non-constant intervals, potentially jumping over relatively long periods when nothing “of interest” is occurring.

There are some advantages to using a time-step method for a particular simulation model. The first, as previously mentioned, is that it is extremely simple to understand. Also, it corresponds with numerical methods for solving differential equations, at least in the traditional sense (see the following section), so it provides some comfort to a modeler coming from that domain. There are few restrictions on the types of state trajectories that are possible, thus allowing for potentially greater latitude in defining state variables with such a framework. This possibly accounts for its popularity in agent-based modeling.

There are a number of disadvantages to the time-step approach as well. Perhaps the most salient is the fact that time is only defined as multiples of Δt and thus most state transitions occur simultaneously. This introduces a number of modeling difficulties (in addition to the ones we will discuss later in the paper).

Since time is only defined at multiples of Δt , the values of state variables in between time steps is essentially undefined. Furthermore, in many cases it is not natural for state changes to only occur at exact multiples. For example, consider a model that includes objects colliding with each other. In a time-step model, such collisions will typically occur in-between time steps, and indeed a collision is often signaled by the fact that at the time step, the objects are in fact embedded in each other. In the case of hard objects this is obviously impossible, so the “fix” is to back up and calculate when the collision actually did occur and compute where the objects actually are at the time increment. This is really an attempt to duplicate the continuous time property of next-event models but in an *ad hoc* manner. It implicitly recognizes the fact that important state transitions often do not occur at regular intervals in multiples of Δt .

Perhaps a more serious limitation is in dealing with the many simultaneous state transitions that are inevitable in a time-step model, since everything happens at the same time. In many situations the order of the state transitions can have a large effect on the behavior of the system. In a time-step simulation, this

occurs very frequently (i.e. at every time step), and whichever tiebreaking mechanism is used, it is bound to be inappropriate for at least some situations.

In most DES models this is not encountered frequently since events can occur at arbitrary times. Although it may be important in some situations to implement a tiebreaking priority scheme for events that happen to occur at exactly identical times, these are generally quite rare, and thus even when they exist will tend to not have a profound impact on the results.

Often time discretization is accompanied by a state space discretization, for example a checkerboard type of movement where each entity occupies a cell and moves from one cell to an adjacent one at each time step. In that case, there is an upper bound on the speed of any entity, namely $\Delta x / \Delta t$, where Δx is the size of the state discretization.

DES models tend to be more efficient than corresponding time step models. Much of this comes from the fact that “uninteresting” periods (i.e. ones for which there are few state changes) are quickly moved through in a DES model. In such circumstances, the typical time step size is too small. Correspondingly, when the model has many rapid state transitions, the DES model “slows down” accordingly, whereas the time step model tends to steps over these places and may not be able to capture the rate at which the changes are occurring. In these circumstances the time step is too big.

Finally, since the size of the time step must often be set very small to capture essential elements of the model, execution times can become much greater than for corresponding DES models. This will be explored further when we study a specific situation with both a time step and DES approach.

3 MOTIVATING EXAMPLE: SOLVING DIFFERENTIAL EQUATIONS

To motivate the concern over the accuracy of time-step models, we will first look at a classic approach to numerically solving differential equations, the Euler method (Golub and Ortega 1992). Although this is the simplest, and perhaps most naïve approach, it is appropriate to study in this context because it is the one that is most analogous to a standard time-step simulation. In one dimension a simple linear differential equation has the form.

$$\frac{dy}{dt} = f(y, t). \quad (1)$$

The Euler method turns the differential equation into a difference equation by discretizing time into intervals of size Δt , thus turning Equation (1) into Equation (2) below (Golub and Ortega 1992).

$$y(t + \Delta t) = y(t) + \Delta t \cdot f(y(t), t) \quad (2)$$

Equation (2) can be thought of as a first-order Taylor series approximation to Equation (1). Note how Equation (2) is analogous to a typical time-step simulation model. The new state value is the old state value plus an increment. The Euler method starts with an initial value $y(0)$ and then uses Equation (2) to update the value of the state y at successive time increments.

It is well-known that difficulties arise with the Euler method even with a very simple linear equation. For example, taking $f(y, t) = -100y + 100$ and $y(0) = 2$ (this is a so-called “stiff” equation) the resulting numerical solution is wildly different for different sizes of time step Δt (See Figure 1).

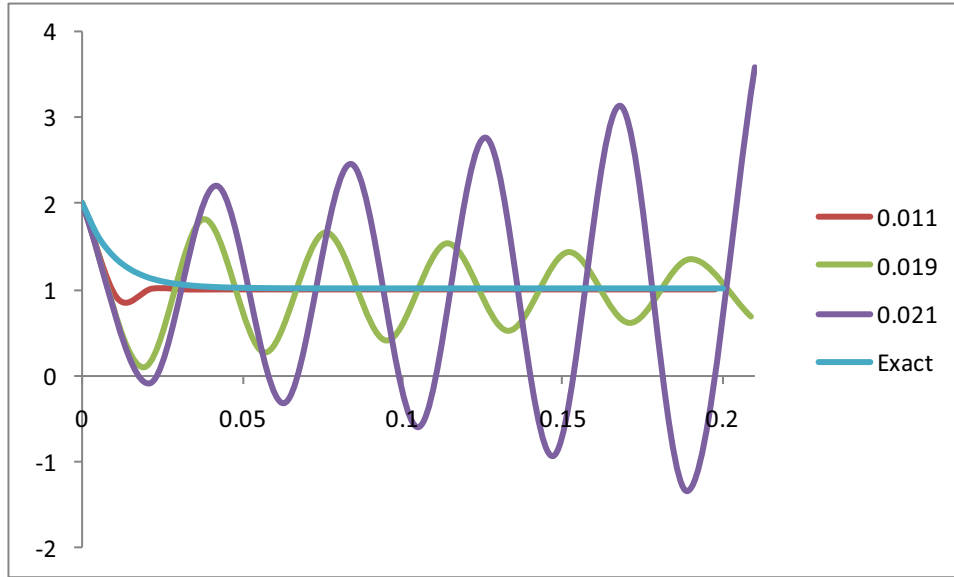


Figure 1: Euler Solutions to Stiff Equation

For values of Δt less than 0.02, the steady-state value of 1.0 is approached, although for $\Delta t = 0.019$ it oscillates about the value, gradually converging. However, for $\Delta t = 0.021$ there is a qualitative difference in the “solution” and it winds up oscillating ever more wildly, ultimately diverging instead of converging.

From a certain perspective, such a disparity between results based only on the size of the time step should be considered a red warning flag for all time-step models. If there can be such an effect for this simple, one-state linear model, what unknown effects might the size of time steps be having on more complicated models? How confident can a modeler be that the results obtained from a simulation (or agent-based) model are qualities inherent in the model itself versus anomalies introduced by the discretization of time and a “bad” size of time step chosen?

3.1 Quantization Method

An elegant and efficient approach to solving differential equations with a DES approach was introduced by Ziegler, quantization (Zeigler, Praehofer, and Kim 2000; Nutaro 2005). A simplified version involves discretizing the state space instead of the time domain. For a simple differential equation model such as discussed in the previous section, the simplest approach is to quantize the state space to be multiples of some fixed quantity D . Depending on the sign of $f(y, t)$, the next level of state will be one higher or one lower than the current state. The time for that transition to occur can be approximated by solving Equation (2) for Δt . Although this approach was originally formulated in terms of DEVS (Zeigler, Praehofer, and Kim 2000), our simplified version can be compactly expressed as an Event Graph (Schruben 1983), as indicated in Figure 2.

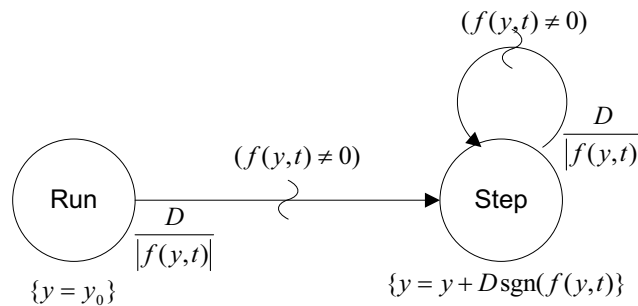


Figure 2: Event Graph for Quantized Model

The parameters for the model in Figure 2 are the initial value, y_0 , and the state space quantum, D ; the state variable is y .

When this approach was applied to the Stiff linear equation of the previous section, the results were much more satisfactory than the Euler method (See Figure 3). For one thing, there was little deviation in behavior for vastly different sized quanta D . All solutions converged quickly to the neighborhood of the steady-state value of 1.0. Those for which the steady-state value was in fact a multiple wound up staying at the correct value, whereas those which were not a multiple of the quantum ended up oscillating around the steady-state value. Even then, the state values were never more than D from the “true” value.

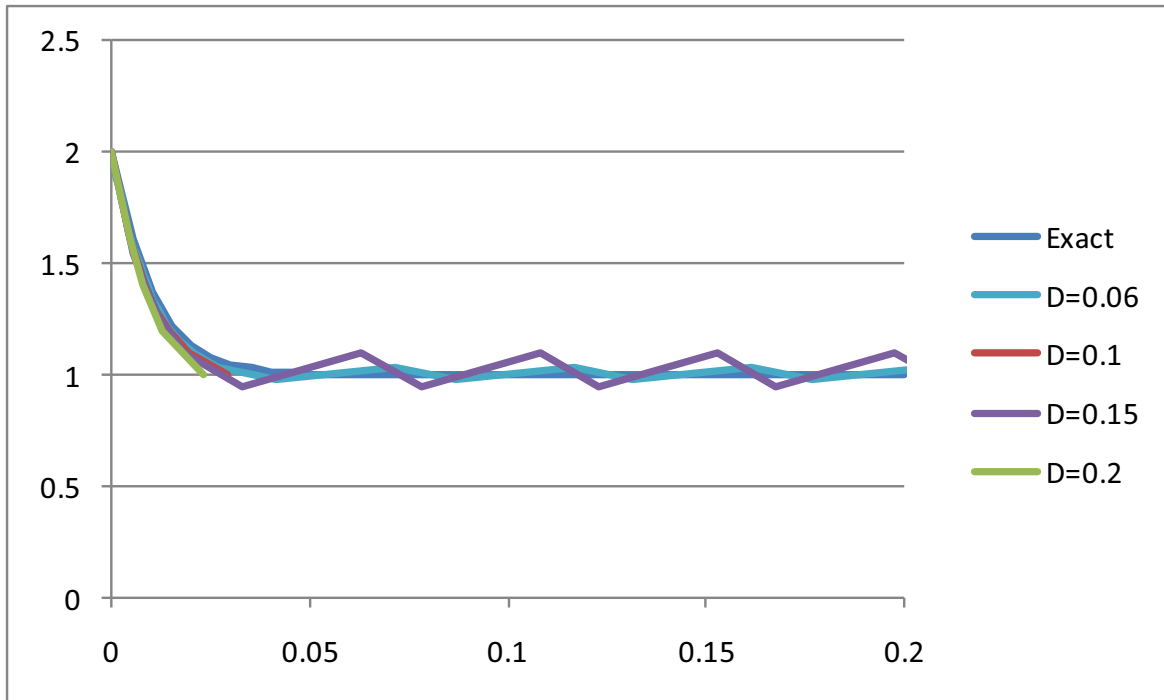


Figure 3: Quantized Solution for Various State Quantizations

The relative ease with which the quantization approach can be implemented along with its superior robustness with respect to the Euler method suggests strongly that a DES approach may be generally preferable to a time-step approach in many more areas than previously thought.

We now turn to simple queueing models to demonstrate the contrast between time-step and DES approaches.

4 QUEUEING MODEL

For our purposes, an M/M/k model is ideal since there are closed-form equations for steady-state measures. This allows the simulation results to be compared with the known exact results.

4.1 DES Model

The DES version of the model was implemented in Simkit (Buss 2010), a package written in Java™. It is based on the Event Graph model shown in Figure 4.

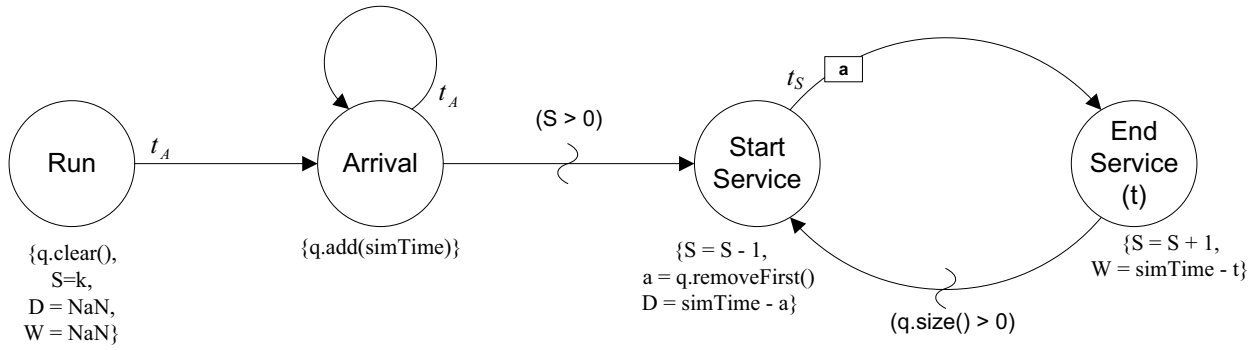


Figure 4: Event Graph for Multiple Server Queue

4.2 DTS Model

Although queueing models are traditionally modeled using DES, occasionally a time-step approach has been used (for example, Lozano and Macias 2004). To create a time-step version of the M/M/k queue requires the observation that the exponential distribution used for inter-arrival times and service times in the DES model can be approximated by the appropriate geometric random variables, which in turn can be generated by successive Bernoulli trials. Thus, for example, at each time step an arrival will occur with probability p_A . For a time step of Δt , the interarrival distribution is $\Delta t \cdot N$ where $N \sim \text{Geom}(p_A)$. The mean is $\Delta t / p_A$ so a mean interarrival time of λ would require $p_A = \Delta t / \lambda$. The service times are handled in a similar manner.

4.3 Analytical Results

For the M/M/k queueing model, there are well-known closed-form solutions for steady-state measures (Gross and Harris 1998). Specifically, if P_0 is the probability the system is empty, L the expected number in the system, and W the expected time in the system,

$$P_0 = \frac{1}{\sum_{i=0}^{k-1} \frac{\rho^i}{i!} + \frac{\rho^k}{k!(1-\rho)}}; L = \rho + \frac{\rho^{k+1} k^k}{k!(1-\rho)^2} \cdot P_0; \text{ and } W = \frac{L}{\lambda}, \quad (3)$$

where λ is the arrival rate, μ the service rate, and $\rho = \lambda / k\mu$.

4.4 Simulation Results and Comparison with Analytic Results

The simulation models were executed starting in an empty and idle state. For the baseline system to study, we chose $\lambda = 7$, $\mu = 4$, and $k = 2$, resulting in a traffic intensity of $\rho = 0.875$. Later higher intensities will be considered. Although normal analysis in this context dictates having a “warm-up” period, for this study we wanted to examine the differences in how steady state was achieved, and so collected output for all customers starting from the first. For 400 independent replications, the results were averaged and as shown in Figure 5, show that the steady state is approached fairly rapidly. By time 1000, the modeler can be fairly certain that steady-state has been approximately reached.

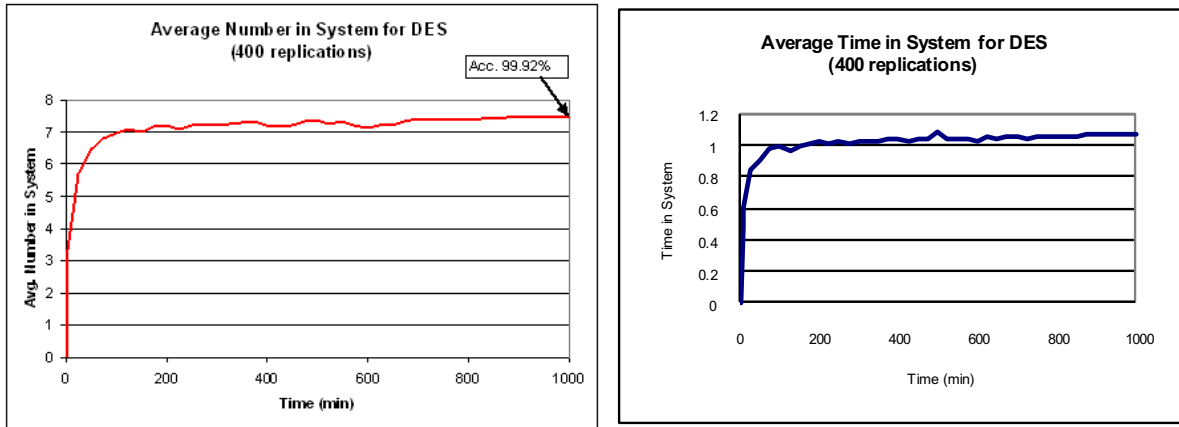


Figure 5: Output Measures for DES Model

The DTS model was then run in a similar manner for three different sizes of time step shown in Figure 6.

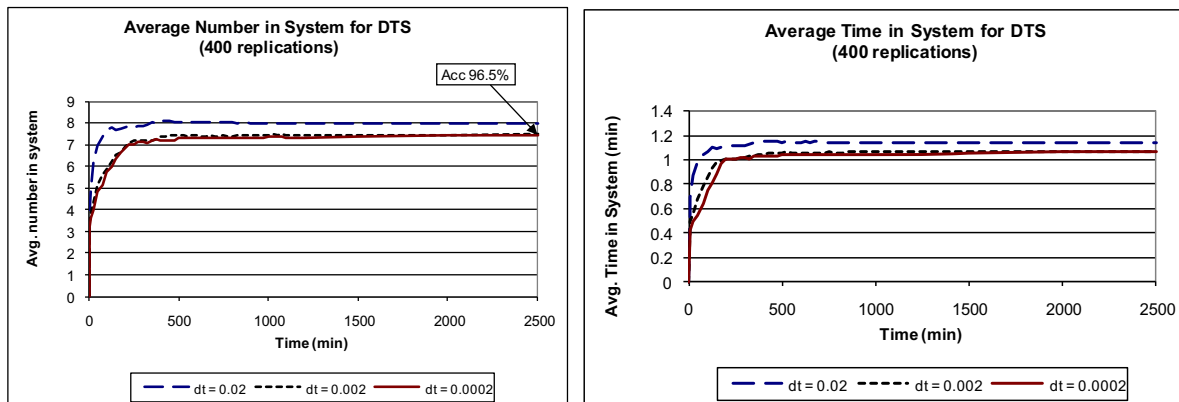


Figure 6: Output Measures for DTS Model with Different Time Steps

These results show that, even though the qualitative behavior is similar to the DES output for a very smaller time steps ($\Delta t = 0.0002$ and 0.002), for the larger time step there is a substantial gap, even when the simulation was run for substantially longer time.

The DTS systems were then run even longer, for 3000 time units, with 1000 replications and confidence intervals computed for the two larger time steps. The results are summarized in Table 1.

Table 1. DES, DTS, and Exact Solutions for the M/M/2 Queue

Results of 1000 Runs	Exact	DES		DTS			
		Mean	95% CI	$\Delta t = 0.02$		$\Delta t = 0.002$	
				Mean	95% CI	Mean	95% CI
Avg. # Cust. in System	7.466	7.449	(7.374,7.524)	8.088	(8.037,8.139)	7.47	(7.426,7.514)
Avg. Time in System	1.066	1.065	(1.064,1.067)	1.157	(1.150,1.164)	1.068	(1.062,1.074)

As can be seen from Table1, the DES model produced highly accurate estimates, whereas for the DTS models, the smaller size resulted in somewhat less accurate estimates (although the confidence in-

tervals did include the true means), whereas the larger time step produced estimates that were substantially off, with confidence intervals that did not cover the true means.

The situation deteriorates even more as the time step is increased. Figure 7 shows the percent error for the average number in the system as the time step is increased. As expected, the error grows considerably as time step as large as 0.1 is used.

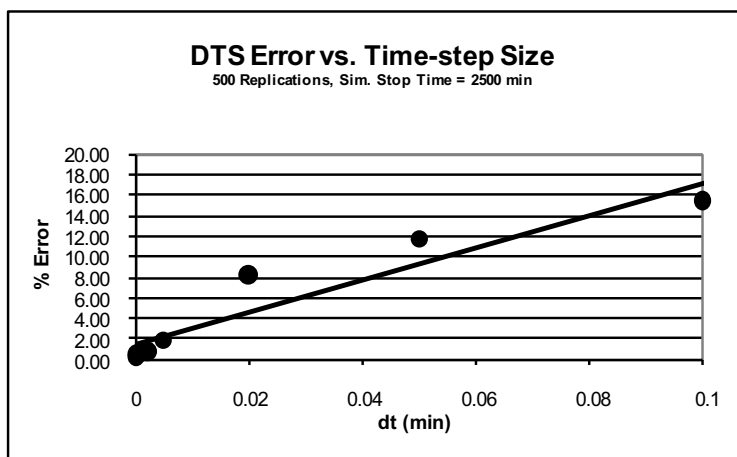


Figure 7. Percent Error as a Function of Time Step for DTS Models

A final set of experiments was run varying the number of servers and the traffic intensity. For these, the service time was fixed at $\mu = 4$, with the number of servers k varied between 3 and 5 and traffic intensities running from 0.1 to 0.95. The results are shown in Table 2.

Table 2. Percent Error with Different Values of k and ρ (“ExeT” = execution time)

k	ρ	DES		DTS							
		Err%	ExeT	$\Delta t = 0.05$		$\Delta t = 0.02$		$\Delta t = 0.002$		$\Delta t = 0.0002$	
				Err%	ExeT	Err%	ExeT	Err%	ExeT	Err%	ExeT
3	0.1	0.0	0.166	67.7	0.015	6.0	0.35	0.3	2.85	0.3	3.5
	0.3	0.4	0.51	43.8	0.02	1.2	0.4	0.3	4.55	0.5	5.77
	0.5	0.1	0.81	22.8	0.03	0.1	0.6	0.2	5.15	0.1	8.54
	0.7	0.2	1.19	5.3	0.033	4.0	0.7	0.6	6.67	0.3	13.6
	0.9	0.9	1.51	14.0	0.04	1.4	0.8	3.7	7.72	1.6	18.75
	0.95	0.5	1.58	19.2	0.05	3.7	1.2	8.2	9.22	3.2	24.5
5	0.1	0.0	0.28	97.4	0.025	5.4	0.38	0.6	3.24	0.2	4.8
	0.3	0.4	0.83	96.2	0.03	3.2	0.53	0.1	5.7	0.5	8.88
	0.5	0.0	1.42	55.4	0.042	2.0	0.71	0.0	8.61	0.1	14.65
	0.7	0.1	1.97	22.5	0.053	1.1	0.93	1.4	11.9	1.6	19.43
	0.9	0.1	2.58	3.8	0.06	11.9	1.12	8.1	13.5	0.7	26.1
	0.95	0.7	2.66	10.3	0.085	7.0	1.2	6.8	14.94	2.0	38.36

These results confirm the previous ones that with very small time steps relatively accurate results can be obtained, but with larger ones results can be substantially off. This disparity tends to increase with increased traffic intensity.

The execution times are also shown in Table 2. With the rapidly changing environment for computing, any such performance measures must be taken with a grain of salt as absolute measures. However, since all simulations were executed on the same computer, the relative performance gives useful informa-

tion. In particular, Table 2 shows that for large time steps, the DTS model executes quite rapidly, much faster than the DES model. However, the errors in the output estimates for these time steps certainly outweigh any performance benefits. Conversely, even though the smallest time step gives relatively accurate results comparable to the DES model, there is a substantial performance penalty more than an order of magnitude in some cases.

5 CONCLUSIONS AND FUTURE WORK

This paper has shown that the time step approach in simulation models is analogous to the Euler method for the numerical solution to differential equations. It has further demonstrated that, at least for some simple queueing models, many of the same problems demonstrated by the Euler method for differential equations arise for simulation models as well. The choice of time step can have a substantial impact on the results of the simulation as well as on the execution time. Briefly, with relatively large time steps, the incorrect answer is obtained very rapidly, whereas for the small time steps required to obtain accurate results, there is a corresponding substantial performance penalty.

Perhaps more serious is the fact that the modeler may be completely unaware of the fact that the size of the time step is impacting the measures obtained from the simulation model. Whether aware or not, since the time step magnitude can affect the results, there is always some uncertainty as to whether it in fact is influencing the results. The examples in this paper were chosen because there were analytic results available to verify the accuracy of the simulations. In general, simulation models are used precisely in situations for which there are no analytic results known. In such circumstances, the use of naïve time step is not advised. Based on the few examples presented here, as well as more models analyzed in follow-up work, the next-event time advance used by DES models appears to be uniformly superior to the standard time-step approach commonly used.

As mentioned previously, this research is not intended to address optimal solutions of differential equations or other numerical methods. In those situations there are many well-known approaches that are superior to the Euler method. Nor is this work attempting to find ideal or optimal values of time step intervals. Rather, the conclusion is that in simulation modeling applications (including most agent-based applications) the use of next-event time advance provides a more reliable and accurate methodology than the use of time-step as it is typically employed.

This has implications for agent-based models, which are almost universally implemented in a time-step manner. The time advance method for such models is almost never questioned. Yet the results shown here suggest that there should be some concern about the efficacy of any results obtained using time step models. The “large” time steps in the previous section were certainly not “large” in any absolute sense, and not even relative to the parameters of the model. Yet the disparity between the known analytic solutions and the errors in the estimates produced in those runs ought to give any modeler pause.

All of the results of this research (the models presented here as well as those studied in other related work) suggest that whenever there is a choice between using a discrete event simulation (DES) and a time-step approach to building a model, that the former should be chosen. In situations for which there may not be an existing standard DES approach, the modeler should also be aware that DES has considerable flexibility and modeling power, especially when used with a compelling modeling methodology such as Event Graphs (e.g. Allore and Schruben 2000; Buss and Sanchez 2005).

This work is continuing by examining more complex models still within the queueing realm (for example, adding balking, reneging, and time-varying parameters) as well as in military combat models as well as models of behavior and social interactions. The area of agent-based modeling is a rich one that is gaining much attention. Although there have been some attempts to incorporate DES concepts in agent-based models, there has not been a full-fledged DES world-view incorporated into an agent-based approach. All agent-based implementations of which we are aware and explicitly or implicitly time-stepped in their world view. Specifically, the logic is generally to advance the clock by a time interval, then update the state of every agent. A DES world view would model each agent as staying in whichever state it was in, and only changing when some event occurred. For example, an agent that was moving towards a

destination would continue in that state until the destination was reached (an event) or some intervening event caused a change to its (internal) state. This approach will be fleshed out in future work.

ACKNOWLEDGMENTS

This paper has benefitted from discussion in the Agent Modeling Focus Group in the MOVES Institute; conversations with Chris Darken and Loren Pietso are particularly appreciated. The paper has also benefitted from the careful and useful feedback from two anonymous reviewers, for which the authors are grateful.

REFERENCES

- Allore, H.G. and L.W. Schruben. 2000. Disease Management Research Using Event Graphs. *Computers and Biomedical Research*, 33, August 2000, 245-259.
- Buss, A. 2010. Discrete Event Simulation Modeling. Working Paper.
- Buss, A. and P. J. Sanchez. 2005. Simple Movement and Detection in Discrete Event Simulation. In *Proceedings of the 2005 Winter Simulation Conference*, ed. M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, , 992-1000 . Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Park, H. and P.A. Fishwick. 2008. A Fast Hybrid Time-Synchronous/Event Approach To Parallel Discrete Event Simulation Of Queuing Networks. In *Proceedings of the 2008 Winter Simulation Conference*, ed. S. J. Mason, R. R. Hill, L. Monch, O. Rose, T. Jefferson, and J. W. Fowler, 795-803. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Golub, G.H. and J.M. Ortega. 1992. *Scientific Computing and Differential Equations*. San Diego, CA: Academic Press.
- Gross, D. and C.M. Harris. 1998. *Fundamentals of Queueing Theory*. Third Edition. Wiley-Interscience. New York, NY.
- Law, A and D. Kelton. 2000. *Simulation Modeling and Analysis*. Third Edition. New York: McGraw Hill.
- Lozano E., Laita L., and Macias E., 2004. An Accelerated-time Simulation of Departing Passengers' Flow in Airport Terminals. *Mathematics and Computers in Simulation*, 67, September 2004, 163–172.
- Nutaro, J. 2005. Constructing Multi-Point Discrete Event Integration Schemes. *Proceedings of the 2005 Winter Simulation Conference*, ed. M. E. Kuhl, N.M. Steiger, F. B. Armstrong, and J. A. Joines, 267-273. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Schruben, L. 1983. Simulation Modeling with Event Graphs. *Communications of the ACM* 26: 957-963.
- Zeigler, B.P., H. Praehofer, and T.G. Kim. 2000. *Theory of Modeling and Simulation*, Second Edition. San Diego, CA: Academic Press.

AUTHOR BIOGRAPHIES

ARNOLD BUSS is a Research Associate Professor in the MOVES Institute at the Naval Postgraduate School. His research is in the area of Discrete Event Simulation modeling, particularly in developing effective and reusable frameworks for developing models. His e-mail address is [<abuss@nps.edu>](mailto:abuss@nps.edu).

AHMED “ALI” AL ROWAEI is a Ph.D. candidate in the MOVES Institute at the Naval Postgraduate School. He received his Master of Science in Applied Physics and Mechanical Engineering from the Naval Postgraduate School. The work presented here is part of his Ph.D. dissertation in the application of discrete event simulation. His e-mail address is [<aalrowae@nps.edu>](mailto:aalrowae@nps.edu).