World Transactions on Engineering and Technology Education
Vol.4, No.2, 2005

An experiment to show undergraduate students the impact of transient analysis decisions on parameter estimation for non-terminating simulated systems

Mary Court, Jennifer Pittman & Huong Pham

University of Oklahoma
Norman, United States of America

ABSTRACT: In this article, the authors present the results of experiments that have been performed to identify the pitfalls of performing *bad* transient analysis when estimating steady-state parameters via the method of independent replications. The intention was to demonstrate to students that failure to delete transient data might lead to confidence intervals that underestimate steady-state parameters. Two types of systems are analysed: M/M/1/GD/∞/∞ systems and an M/M/s/GD/∞/∞ optimisation problem. These systems have been chosen since they are typically taught in an undergraduate stochastic operations research course where a closed-form solution of the steady-state parameter exists. Surprisingly, the results support the opposite of the authors' original intention – regardless of run length, ignoring transient analysis often leads to the same level of coverage at greater precision, or provides no gain in coverage to justify the effort of performing transient analysis. Thus, the authors now pose the question: *should transient analysis be taught?*

INTRODUCTION

One of the most difficult topics for undergraduate students taking their first stochastic operations research or simulation course is non-terminating systems analysis [1]. At issue is the student's ability to understand that output data generated from a non-terminating system's simulation will have both transient and steady-state data, ie the data are not independently and identically distributed data (iid). Additionally, they are uncomfortable or inexperienced with utilising approximation tools (simulation) that rely on *ad-hoc* methodologies (eg graphical techniques to distinguish between transient and steady-state behaviour) and statistical laws (eg the central limit theorem) for parameter estimation. This is understandable since, more often than not, undergraduate students have only used mathematical modelling techniques that are *guaranteed* (as long as the underlying assumptions of the technique are not violated) to generate *one-and-only* (or hopefully, the optimal) solution to a problem. In contrast, simulation output analysis for non-terminating systems may generate several different *approximations* for the unknown parameter of interest.

Complicating the issue is the inability of students to validate their results. In general, these students are not comfortable with the amount of judgement/skill/experience required to evaluate their findings (eg confidence intervals about the parameters of interest). Simulation analysis (particularly simulation output analysis of non-terminating systems) tends to be *too ad-hoc* for the *typical* undergraduate student; while non-terminating output analysis is too important a topic to ignore when teaching simulation modelling courses. In the authors' opinion, ignoring this topic when teaching simulation modelling is equivalent to generating *inadequate simulation practitioners*.

The authors present a set of experiments to help identify the pitfalls of performing *bad* transient analysis when estimating steady-state parameters via the method of independent replications. The intention of the experiments is to demonstrate to industrial engineering undergraduate students that failure to delete transient data (or not enough transient data) will lead to *poorer* confidence intervals than confidence intervals generated when the student takes the time to more accurately identify and discard transient data.

In fact, the authors' goal is to demonstrate that these confidence intervals are more likely to not cover the true mean, or precision (half-width size) will be less (greater half-width size) than the confidence intervals generated when transient data are discarded. The authors' experiments are designed for two classes of non-terminating systems where the authors perform what they define as *perfect transient analysis*. While performing the experiments, the authors also explore the impact on confidence interval generation when the analyst has run the simulation long enough to generate what is defined as *perfect run length* versus not running the simulation long enough – *insufficient run length*. The results are surprising: for those cases where transient analysis is carried out *badly* (or even ignored), the confidence intervals provide coverage at greater precision than those cases where *perfect transient analysis* is performed.

The remaining sections of the paper are in the following order:

- The background section expands upon the motivations for this research;
- The definitions section delineates what the authors mean by performing transient analysis *badly* and they present their definitions of perfect transient point, worst-case transient analysis and perfect run length;
- The methodology, results and analysis section defines the experiments that have been performed, how definitions were put into practice and the resulting confidence intervals;

- The conclusions and future research section is a discussion of the results; some questions have been posed for educators and the avenues of future research that the authors wish to explore.

BACKGROUND

Figure 1 depicts a high-level concept map for teaching the confidence interval generation of terminating and non-terminating systems [2]. This map is geared towards an upper-level undergraduate course since several of the more mathematically complicated techniques available for generating confidence intervals are omitted from the map. But even with those techniques omitted, one can see the complexity of the topics and their *links*. This may be why some researchers have suggested that simulation not be taught at the undergraduate level or that the topics of modelling and analysis be split between two required courses [3]. However, several topics could be covered as prerequisite knowledge, such as those topics found in engineering statistics and stochastic operations research or queuing courses. This is the case at the University of Oklahoma, Norman, USA. The undergraduate programme in industrial engineering has been revamped so that the simulation course required for the industrial engineering students is incorporated into the fall semester of the senior year [1].

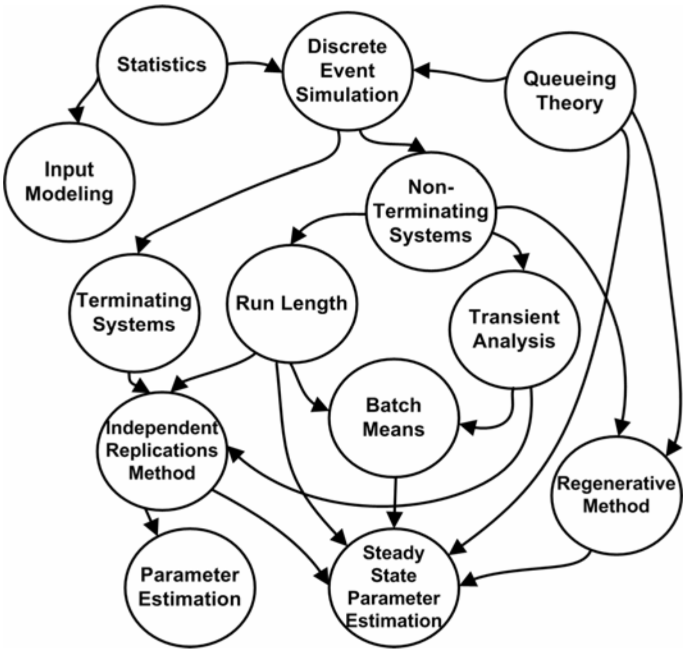

Figure 1: High-level concept map for generating confidence intervals of terminating and non-terminating simulations systems ([2], modified).

The goal of this undergraduate course in simulation is to produce a graduate *who is capable of applying a simulation language for the purpose of analysing, designing and comparing systems* [1]. Consequently, the curriculum supports the concept map in Figure 1 where an engineering statistics course is taken in the spring semester of the sophomore year, an experimental design course is taken in the fall semester of the junior year, a stochastic operations research course and a quality engineering course are taken in the spring semester of the junior year, the required simulation course is taken in the fall semester of the senior year, and a graduate-level statistical analysis course in simulation is offered as an elective in the spring semester of the senior year.

In this course sequence, the first introduction of simulation analysis is incorporated into the stochastic operations research course. Here, Markov chain analysis and queuing theory are followed by Monte Carlo simulation, discrete-event simulation logic and then, output analysis of non-terminating systems. The idea is to be able to have the student perform experiments so that queuing theory results of steady-state parameters can be contrasted against the results generated via a simulation experiment. At this point in the stochastic operations research course, the student is familiar with the concept of steady state and has enough of a statistics background to understand the importance of placing confidence intervals around results generated via statistical (simulation) experiments.

A *learn-by-doing* assignment is then imposed upon the student. The student is required to take a spreadsheet approach for invoking discrete-event simulation logic and then, calculate an unknown parameter of interest (eg the average waiting time in queue) for an M/M/1/GD/∞/∞ system. The assignment also requires the student to perform transient analysis in Microsoft® *Excel©*. The student must calculate *enough observations* (address the issue of run length) of the unknown parameter so that a graph of the cumulative average for that unknown parameter is generated to assist the student in determining where in the output data the system *exhibits steady-state behaviour*. Based on the student's *judgement* of where steady-state begins, the student must then generate 20 independent simulation runs with the transient phase deleted from each of the 20 runs (the method of independent replications).

Then, by the central limit theorem, the student may use the average from each of the 20 runs to calculate a 95% confidence interval of the unknown parameter. Armed with their confidence interval, students are to contrast the results against the closed-form solution (from queuing theory) for an M/M/1/GD/∞/∞ system. Hopefully, the closed-form solution is contained within their confidence interval, but, on average, 5% (or more) of the students in the class will generate confidence intervals that do not contain the closed-form solution. Those cases are used to illustrate several important points to the class:

- A failure rate of at least 5% is to be expected since a 95% confidence interval implies that the true parameter will lie outside the confidence interval 5% of the time.
- Failure to delete enough transient or insufficient run lengths may lead to *poorly* constructed confidence intervals. Thus a greater than 5% failure rate may be detected since some students:

  a. May have performed their transient analysis poorly (so they do not have iid observations);
  b. May not have generated *enough observations* in steady state (insufficient run lengths);
  c. Some students may have done both (a) and (b).

- A greater than 5% failure rate could also be attributed to the confidence interval method invoked: the method of independent replications. Here, the transient deletion is treated as a constant across all runs. So, if the first run of the simulation just happens to have a shorter transient phase than any of the other runs, the sample means from each of the other runs will tend to underestimate the parameter of interest (initialisation bias in the method of independent replications).
- The previous points bring one to the next point of interest. Attaining the stated level of coverage (95%) requires the

confidence interval to rely upon normally distributed means, or at least approximately normally distributed means, from each replication. Thus, if the parameter of interest's underlying distribution is not normally distributed, and the run length is not long enough for the central limit theorem to *kick-in*, less than 95% coverage is to be expected (see ref. [4]). Thus, increasing the run length and the number of runs may yield confidence intervals with coverage closer to the stated level of confidence (in this case, 95%).

- If the error rate is greater than expected, the student should also question the algorithm used to generate the random numbers. The algorithm must be robust enough to yield uniformly distributed [0,1] random variates that do not repeat (long random number streams). This ensures that the exponentially distributed inter-arrival and service rates are iid.
- For the M/M/1/GD/∞/∞ system simulated, the value of the steady-state parameter can be obtained from well-known queuing theory results; in practice, the true value is frequently unknown, which is the justification for performing simulation analysis. So, in practice, students will not be able to judge the validity of their results (ie the validity of their confidence intervals) against known parameters. They must either be confident in the output analysis techniques that they followed to obtain the estimate, or they must use some other reasoning mechanism to validate their results. One common reasoning approach applied in practice is to reduce the complexity of the system down to a system where well-known results exist. Then, a comparison may be made between the simplified model's well-known results and the output analysis results for the more complicated system. For example, if the *more-complicated* system (the system of study) has a machine that is subjected to breakdowns while the well-known system does not, the practitioner should expect the estimate for the average waiting time in queue of the more complicated system (the one that is simulated) to be *greater* than the well-known result.

In short, the confidence intervals for non-terminating parameters generated via simulation output analysis techniques will tend to be closer to the stated level of confidence if the student deletes enough transient, runs the simulation long enough, utilises more than 20 replications and has a parameter of interest that tends to be normally distributed.

DEFINITIONS

The goal of the experiments is to demonstrate the impact on confidence intervals (generated via the method of independent replications) when the analyst performs transient analysis *badly*. First, there needs to be agreement on how *bad transient analysis* should be defined; to offset this definition, there needs to be agreement on how *perfect transient analysis* should be defined. Additionally, definitions have to be developed that can be understood at the undergraduate level.

It is proposed that a student is able to achieve *perfect transient analysis* when he/she identifies the point in the output data, such that from that point to the end of the simulation run, the average of the remaining data equals the true mean of the unknown parameter of interest. For example, if the student wishes to obtain an estimate of the true average waiting time in queue, *perfect transient analysis* equates to deleting enough of the initial data, such that the remaining data, when averaged (the sample mean from the data), equals the true mean.

In order to investigate cases of performing transient analysis *badly*, it is agreed that the *worst-case-transient analysis* is to have the student/practitioner ignore (either intentionally or unintentionally) transient analysis altogether. In other words, one needs to investigate cases where the student/practitioner does not delete any initial data from the simulation runs and hence, the method of independent replications would see its worst case of initialisation bias.

However, one also wishes to explore cases where the student/practitioner *just happens to* overcome the initialisation bias of the worst-case-transient analysis by generating *enough* steady-state data to offset the initialisation bias. So, cases will be explored where, even though the practitioner intentionally or unintentionally chose to ignore transient analysis, he/she still generated what is called the *perfect run length*. Then, the *perfect run length* is defined as the ability of the analyst to intentionally or unintentionally ignore transient analysis and yet, generate a simulation run whose data, when averaged across the entire run (the replication's sample mean), equals that of the true mean.

Note that while several transient analysis methods exist (see ref. [4]), a pilot run is used to determine transient analysis that will apply that warm up period (perfect transient point) across all runs. Then, for the method of independent replications, transient deletion in the remaining runs (replications) is treated as a constant (perfect transient point of the pilot run).

Utilising a pilot run is also the approach to be taken for analysing perfect run length. In practice, the run length is highly influenced by the objectives of the simulation study and is typically determined via a pilot run of the simulation (and usually after transient analysis has been concluded). So again, as with the *ad-hoc* methodologies of transient analysis, run length selection is dependent upon the analyst's ability to *judge* how long the simulation should be run. However, once run length is determined, it is fixed to uphold the method of independent replications requirement of fixed run lengths for all runs. Thus, for these perfect run length cases, a pilot run of the model is generated to determine the time of perfect run length and then, run the remaining simulations with the perfect run length time invoked as the stopping rule for all runs.

METHODOLOGY, RESULTS AND ANALYSIS

Two queuing systems cases are analysed, namely:

- *Case 1*: M/M/1/GD/∞/∞ systems at three levels of ρ (=0.50, 0.75, 0.90);
- *Case 2*: An M/M/s/GD/∞/∞ optimisation problem with λ=2/minute, μ=0.5/minute, a per server cost of $9/hour and a delay cost to the customer of $0.05/minute, at s=5 and s=6, see [5].

The cases are chosen since well-known queuing theory results exist, and these systems are typically introduced in an undergraduate stochastic operations research course. The parameter of interest for Case 1 is the average waiting time in the queue. The Case 2 system types are characteristically used to introduce the formulation required to solve queuing optimisation problems. Here, the objective is to minimise the total expected cost in terms of the expected delay cost to the customer as a function of the service level (the number of servers, s). In the queuing analysis, the formulation is straight-forward (see ref. [5]), and a closed-form solution exists for the minimum expected total cost.

For the given values of the Case 2 system, the minimum total expected cost occurs when s=5. Since the waiting cost to the customer and the per server costs are fixed, to solve the problem through simulation analysis is equivalent to determining a parameter estimate for the average waiting time in queue (W_q – the only unknown parameter). All cases were simulated via *Arena* 7.01 software (see ref. [6]), and analysed using *Arena* 7.01 and Microsoft® *Excel*©.

For the systems of Case 1, the methodology is as follows:

1. A pilot run of the model is made for the waiting time in the queue where the stopping rule for the pilot run is invoked when the average waiting time across the run equals the theoretical value. At this point, the simulation is terminated and the simulation's run length is noted as the *perfect run length*. Then, 20 independent replications of the simulation model are generated with the perfect transient's run length time utilised as the stopping condition for each of the 20 runs. The mean from each of the replications is then utilised to generate a 95% confidence interval about the average waiting time in queue. Table 1 lists the perfect run lengths identified in the experiment.
2. Run lengths of 6,000, 20,000, 50,000 and 100,000 time units are utilised for all cases in order to identify the perfect transient point at various run lengths, and to allow the *worst-case transient analysis* (no deletion of transient) at various run lengths to be explored. Since the perfect run length determined for ρ=0.90 was found to be greater than 100,000 time units, an additional case is utilised at a run length of 1,000,000 time units (see Table 2).
3. For all run lengths of 2 above, (except perfect run length), perfect transient analysis is invoked to determine where in that particular run length, the average of the data equals the theoretical value (*perfect transient point*). The process is to export the output data of the simulation's pilot run into an *Excel* spreadsheet and perform what one might call a *reverse cumulative average*.
   For example, with a run length of 20,000 time units, the first average obtained is of all of the output responses generated over the 20,000 time units. If this average equals the true mean, the entire run is considered to be in steady-state. If not, the first observation is dropped and the remaining data are averaged. If the remaining data's sample mean equals that of the true mean, then the simulation time for the second observation is noted as the *perfect transient point*. If not, the second observation is dropped from the data set and the process of dropping each successive observation is repeated until the perfect transient point is found.
   Table 2 lists the perfect transient points. It should be noted that for some run lengths, perfect transient cannot be found. For the run lengths of 20,000 time units and 50,000 time units, the perfect transient points for ρ=0.50 are considerably greater than the perfect transient points for ρ=0.75. It should also be noted that for the run length of 100,000 time units, the perfect transient point is 0.00 for ρ=0.50 and 81,009.49 for ρ=0.75. At first, this may seem to contradict the trend expected for longer run lengths within a particular ρ or for the same run length across different ρ's. However, an explanation can be found given that only a pilot run of the simulation is used to determine the perfect transient point; transient itself is stochastic and thus, the perfect transient point is also stochastic. So while one would expect to see the perfect transient point increase as run lengths increase, since it is stochastic, it will *move* within a ρ at various run lengths and *move* for run lengths at various ρ's.
4. If a perfect transient point can be found for a particular run length, two more confidence intervals are generated via the method of independent replications, namely:

   - Firstly, when the replications have the perfect transient deleted but the total run length is terminated at the original run length's time unit. For example, if the run length is 20,000 time units and the perfect transient point is found to occur at 12,210 time units, each replication will have a total of 7,790 time units worth of data available to calculate each replication's sample mean.
   - Secondly, when perfect transient is deleted from each of the replications and the total run length is modified to equal that of the perfect transient's time units plus the original run length's time units. Following the previous example with an initial run length of 20,000 time units, the new run length is 32,210 time units for each of the replications, where the first 12,210 time units are specified as *warm-up* (amount of simulation time deleted as transient) and the remaining 20,000 time units of data are available for calculating each replication's sample mean.

Table 1: Worst-case transient analysis run lengths and perfect run lengths for Case 1 systems.

Run Length	ρ=0.50	ρ=0.75	ρ=0.90
Perfect run length	62,314.95	16,709.11	965,052.03
Worst-case transient analysis	6,000	6,000	6,000
	20,000	20,000	20,000
	50,000	50,000	50,000
	100,000	100,000	100,000
			1,000,000

Table 2: Perfect transient point of run lengths for Case 1 systems (-- indicates not found).

Run Length	ρ=0.50	ρ=0.75	ρ=0.90
6,000	--	--	--
20,000	12,210.40	868.93	--
50,000	28,675.29	672.31	--
100,000	0.00	81,009.49	--
1,000,000			250,537.13

For Case 2, the methodology for Case 1 is invoked for two realisations of s=5 and s=6. Table 3 contains the prefect run length and worst-case transient analysis run lengths for the Case 2 systems. It should be noted that an additional run length of 500,000 time units is required when s=6, since perfect run length is found at 310,498.33 time units.

Table 4 contains the perfect transient point of the run lengths and, as with some of the Case 1 systems, some run lengths for the Case 2 systems contain no perfect transient point.

Table 3: Worst-case transient analysis run lengths and perfect run lengths for Case 2 systems.

Run Length	s=5	s=6
Perfect run Length	11,218.90	310,498.33
Worst-case transient analysis	6,000	6,000
	20,000	20,000
	50,000	50,000
	100,000	100,000
		500,000

Table 4: Perfect transient point of run lengths for Case 2 systems (-- indicates not found).

Run Length	s=5	s=6
6,000	--	1251.18
20,000	--	14,400.08
50,000	47,917.48	48,129.71
100,000	76,131.37	33,915.82
500,000		315,533.26

Table 5 reveals the 95% confidence intervals generated for the worst-case transient analysis and perfect run lengths (as defined in Table 1) of the Case 1 systems. A shaded box indicates a run length that does not apply for that ρ. For each ρ, the confidence interval generated at perfect run length is indicated by a double-border cell. Of the 16 confidence intervals generated, only one failed to contain the true mean (W_q) and occurred at the 6,000 run length for ρ=0.75. It should be noted that this is the only confidence interval generated by a run length below perfect run length that does not contain W_q.

Table 5: 95% confidence intervals of worst-case transient analysis run lengths and perfect run lengths for Case 1 systems.

M/M/1 / W_q / Run Length	ρ=0.50	ρ=0.75	ρ=0.90
	0.500	2.250	8.100
6,000	0.507+/- 0.019	2.363+/- 0.095	8.136+/- 0.810
16,709.11		2.280+/- 0.056	
20,000	0.499+/- 0.008	2.262+/- 0.050	8.229+/- 0.446
50,000	0.500+/- 0.007	2.260+/- 0.042	7.962+/- 0.230
62,314.95	0.500+/- 0.004		
100,000	0.501+/- 0.003	2.277+/- 0.028	8.109+/- 0.176
965,052.03			8.077+/- 0.070
1,000,000			8.083+/- 0.069

Readers should recall that, for all of the run lengths, no transient is deleted, yet 15 of the 16 confidence intervals contain the true mean. For ρ=0.50 and ρ=0.75, the best precision for the confidence interval is attained at the 100,000 run length, while for ρ=0.90, it is attained at the 1,000,000 run length.

These results indicate that there is no advantage to attaining perfect run length in terms of generating a confidence interval that contains W_q. However, the analysis does show that as the run length increases, the precision of the confidence interval improves.

Table 6 contains the 95% confidence intervals generated from the perfect transient analysis runs of the Case 1 systems. Readers should recall from the methodology section that two types of run lengths are performed after the perfect transient point is identified.

The first, (a), is run at the original run length with a warm-up period set at the perfect transient point. Thus, the total simulated time is the original run length, but the output data collected from each replication have the transient data deleted from the run. The second, (b), is run with the total simulated time equal to the original run length plus the perfect transient point. The net effect is that (b) will have *steady-state* data collected for the original run length's time units while (a) will have less *steady-state* data collected. A '--' symbol indicates that a perfect transient point could not be found for the run length of (a) (see Table 2). A shaded box indicates a run length that does not apply for that ρ.

All confidence intervals generated contain W_q, and, except for two sets of confidence intervals (ie ρ=0.75 with original run length at 100,000 and ρ=0.90 with original run length at 1,000,000), the precision of the (a) confidence intervals is the same or better than the precision of the confidence intervals of (b). Then, there seems to be no advantage to increasing the run length for *steady-state* data collection once perfect transient is deleted. As with the previous results of Table 5, precision improves as run length increases.

Table 6: 95% confidence intervals generated via perfect transient analysis at two run lengths for Case 1 systems: (a) original run length and (b) original run length + perfect transient point.

M/M/1 / W_q / Original Run Length		ρ=0.50	ρ=0.75	ρ=0.90
		0.500	2.250	8.100
6,000	(a)	--	--	--
	(b)	--	--	--
20,000	(a)	0.488+/- 0.012	2.264+/- 0.054	--
	(b)	0.496+/- 0.013	2.266+/- 0.057	--
50,000	(a)	0.500+/- 0.005	2.261+/- 0.044	--
	(b)	0.511+/- 0.005	2.257+/- 0.044	--
100,000	(a)	0.501+/- 0.003	2.289+/- 0.050	--
	(b)	0.501+/- 0.003	2.269+/- 0.023	--
1,000,000	(a)			8.087+/- 0.094
	(b)			8.112+/- 0.076

Tables 7 and 8 reveal the results of the analysis for the Case 2 systems. As with the Case 1 systems, the same trends exist for the Case 2 systems' confidence intervals:

- For the worst-case scenario and perfect run lengths (see Table 7), there is no advantage to attaining perfect run length in terms of generating a confidence interval that contains Wq; also, as the run length increases, the precision of the confidence interval improves;
- For the perfect transient runs (see Table 8), there seems to be no obvious advantage to increasing the run length for *steady-state* data collection once perfect transient is deleted; as with all previous results, precision improves as the run length increases.

Table 7: 95% confidence intervals of worst-case transient analysis run lengths and perfect run lengths for Case 2 systems.

M/M/s / W_q / Run Length	s=5	s=6
	1.108	0.285
6,000	1.140+/-0.092	0.294+/-0.026
11,218.90	1.154+/-0.054	
20,000	1.138+/-0.041	0.298+/-0.013
50,000	1.117+/-0.026	0.290+/-0.007
100,000	1.114+/-0.018	0.286+/-0.005
310,498.33		0.286+/-0.002
500,000		0.286+/-0.002

Table 8: 95% confidence intervals generated via perfect transient analysis at two run lengths for Case 2 systems: (a) original run length and (b) original run length + perfect transient point.

M/M/1 / W_q / Original Run Length		s=5	s=6
		1.108	0.285
6,000	(a)	--	0.299+/-0.028
	(b)	--	0.299+/-0.026
20,000	(a)	--	0.288+/-0.012
	(b)	--	0.288+/-0.015
50,000	(a)	1.102+/-0.095	0.283+/-0.007
	(b)	1.101+/-0.103	0.282+/-0.008
100,000	(a)	1.104+/-0.016	0.282+/-0.004
	(b)	1.103+/-0.018	0.285+/-0.004
500,000	(a)		0.284+/-0.002
	(b)		0.284+/-0.002

CONCLUSIONS AND FUTURE RESEARCH

Granted, the confidence intervals generated are for only one set of 20 replications each, while coverage analysis requires several sets of 20 replications to predict the accuracy of confidence intervals' coverage (see ref. [4]), but this is not the intention of the experiments. The objective of the experiments is to convey to undergraduate students the danger of not performing transient analysis when estimating an unknown parameter for a non-terminating system.

The method of independent replications was chosen since it tends to be more readily understood by undergraduate students than, say, the batch means method [2]. Additionally, the method (independent replications) tends to suffer from initialisation bias and thus, performing transient analysis should help to reduce the bias.

Since transient analysis relies on *ad-hoc* techniques, students tend to have difficulty in applying the techniques and justifying/validating the results [2]. Two types of queuing systems have been selected that are typically taught in a stochastic operations research course to illustrate transient analysis issues. Surprisingly, very little evidence was found to support the fact that performing transient analysis would lead to *better* confidence intervals. That is, even when transient analysis is ignored, the true mean is contained within the confidence interval and usually with greater precision.

Table 9 also supports this finding. Here, the *equivalent* run lengths for the Case 1 systems are compared by restating the results found in Tables 5 and 6. For the (b) confidence intervals of Table 6, the total run length is the original run length plus the time unit of the perfect transient point. So, the (b) confidence intervals have the same amount of data collected (in terms of time) over the run as the worst-case transient analysis run length times of Table 5.

Table 9: 95% half-widths of worst-case transient analysis versus perfect transient analysis at *equivalent* run lengths for Case 1 systems.

Equivalent Run Length	Worst-Case Transient Analysis (see Table 5)	(b) Run Length (see Table 6)
ρ=0.50		
6,000	0.019	--
20,000	0.008	0.013
50,000	0.007	0.005
100,000	0.003	0.003
ρ=0.75		
6,000	0.095*	--
20,000	0.050	0.057
50,000	0.042	0.044
100,000	0.028	0.023
ρ=0.90		
6,000	0.810	--
20,000	0.446	--
50,000	0.230	--
100,000	0.176	--
1,000,000	0.069	0.076

However, readers should recall that the worst-case transient analysis occurs when no transient is deleted. So, in theory, the (b) generated confidence intervals should be *better* than the confidence intervals generated under the *equivalent* worst-case transient analysis run length, since they contain *steady-state* data. Readers should also recall, as displayed in Table 9, that a '--' symbol indicates that a perfect transient point could not be found for the run length and only the (*) confidence interval did not contain W_q. Then, in general, the *best* half-width occurs when no transient is deleted.

In fact, the worst-case transient analysis runs generated more *valid* (W_q is within the confidence interval) confidence intervals at shorter run lengths than (b) (since perfect transient could not be found at the shorter run lengths). So, the runs with no transient deleted generated 12 valid confidence intervals, while the *steady-state* runs only generated seven valid confidence intervals.

For the seven valid confidence intervals generated via the *steady-state* runs, the runs with no transient deleted generated confidence intervals with equal or better precision five out of those seven times. The same comparison (see Table 10), when performed for the Case 2 systems, yields similar, if not better, results.

Table 10: 95% half-widths of worst-case transient analysis versus perfect transient analysis at *equivalent* run lengths for Case 2 systems.

Equivalent Run Length	Worst-Case Transient Analysis (see Table 7)	(b) Run Length (see Table 8)
s=5		
6,000	0.092	--
20,000	0.041	--
50,000	0.026	0.103
100,000	0.018	0.018
s=6		
6,000	0.026	0.026
20,000	0.013	0.015
50,000	0.007	0.008
100,000	0.002	0.004
500,000	0.002	0.002

Thus, the question is posed: at the undergraduate level, should transient analysis be taught when teaching the method of independent replications? The authors' conclusion is *yes*, but with the following emphases:

- Transient analysis is an *ad-hoc* methodology and, as such, there is no guarantee that the confidence interval generated will be *better* (greater precision) than a confidence interval generated when transient data are present;
- Run length seems to be the most important factor when trying to obtain confidence intervals containing the true mean.

It is the authors' opinion that strong emphasis should be placed on the run length. It seems that, at least, for the systems studied, transient data, whether present or not, have very little impact on the validity of the confidence interval. However, run length does seem to impact significantly. So, the student should come away with the realisation that running the simulation *long enough* is much more important than identifying steady-state behaviour.

The confidence intervals were generated via the method of independent replications. Future research will be aimed at the impact of transient analysis when utilising the batch means method. The authors' prediction is that, since the batch means method is a single-replication method (see refs [4] or [6]), determining the perfect transient point will assist in providing a better confidence interval than if transient analysis is performed *badly*.

However, the authors are not comfortable with the same prediction for perfect run length analysis since, if independent batches can be generated, at least one of the batches will contain transient data in the sample (or batch) mean.

ACKNOWLEDGEMENT

The authors would like to acknowledge that the initial results of this transient analysis research have been published in the *2005 Winter Simulation Conference Proceedings* CD-ROM, in the paper entitled, *Should transient analysis be taught?*

REFERENCES

1. Court, M.C., A case study on the impact of Web based technology in a simulation analysis course. *Simulation, Special Issue: Simulation in Education and Education in Simulation*, 74, **4**, 207-213 (2001).
2. Court, M.C., The impact of using *Excel* macros for teaching simulation input and output analysis. *Inter. J. of Engng. Educ.*, 20, **6**, 966-973 (2004).
3. Altiok, T., Kelton, W.D., L'Eucyer, P., Nelson, B., Schmeiser, B., Schriber, T., Schruben, L. and Wilson, J., Various ways academics teach simulation: are they all appropriate? *Proc. 2001 Winter Simulation Conf.*, Arlington, USA, 401-407 (2001).
4. Law, A. and Kelton, W.D., *Simulation Modeling and Analysis* (3rd edn). Boston: McGraw Hill (2000).
5. Winston, W., *Operations Research: Applications and Algorithms* (4th edn). Belmont: Brooks/Cole-Thomas Learning (2004).
6. Kelton, W.D., Sadowski, R. and Sturrock, D., *Simulation with Arena* (3rd edn). Boston: McGraw Hill (2004).