Proceedings of the 2007 Winter Simulation Conference
S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, eds

AUTOMATING DES OUTPUT ANALYSIS: HOW MANY REPLICATIONS TO RUN

Kathryn Hoad
Stewart Robinson
Ruth Davies

Warwick Business School
University of Warwick
Coventry, CV4 7AL, U.K.

ABSTRACT

This paper describes the selection and automation of a method for estimating how many replications should be run to achieve a required accuracy in the output. The motivation is to provide an easy to use method that can be incorporated into existing simulation software that enables practitioners to obtain results of a specified accuracy. The processes and decisions involved in selecting and setting up a method for automation are explained. The extensive test results are outlined, including results from applying the algorithm to a collection of artificial and real models.

1 INTRODUCTION

Appropriate use of a simulation model requires accurate measures of model performance. This in turn, requires decisions concerning three key areas: warm-up, run-length and number of replications. These decisions require specific skills in statistics. Because of the variety and complexity of statistical methods used to make informed decisions on these matters, at the moment, most simulation software provides little or no guidance to users on making these important decisions. In surveys of simulation users both Hlupic (1999) and Hollocks (2001) identify better experimental support as being much needed. Hollocks notes that 'simulation is increasingly in the hands of non-specialists, so the quality of experimentation is at higher risk'. The purpose of this paper is to document the development of a methodology for automatically advising a simulation user on one of these three key decisions, how many replications should be run? The specific objectives are:

- To determine the most appropriate method/s for automating the selection of the number of replications to run.
- To determine the effectiveness of the analysis method/s.
- To revise the method/s where necessary in order to improve their effectiveness and capacity for automation.
- To propose a procedure for automated output analysis of the number of replications.

At present this work is only concerned with looking at analysis of a single scenario.

Multiple replications are performed by changing the random number streams that are used by the model and re-running the simulation. The aim is to produce multiple samples in order to obtain a better estimate of mean performance (Robinson 2004, Law 2007, Banks et al. 2005). The question therefore arises as to how many replications are needed. The limiting factors will be computing time and expense. Assuming that performing N replications achieves a satisfactory estimate of mean performance as required by the user, performing more than N replications may be an unnecessary use of computer time and considerable expense. However, performing fewer than N replications could lead to inaccurate results and thus to incorrect decisions being made.

There are three main methods found in the literature for choosing N: Rule of Thumb (Law and McComas 1990), a simple Graphical Method (Robinson 2004) and the Confidence Interval (with Specified Precision) Method (Robinson 2004, Law 2007, Banks et al. 2005). Law and McComas (1990) recommend running at least 3 to 5 replications. This rule of thumb is useful for telling users that relying upon the results of only one run is unwise. However, it makes no allowance for the characteristics of a model's output, so although running 3 to 5 replications may suffice for one model it may be woefully inadequate for another.

In the simple graphical method a user carries out a series of replications and plots the cumulative mean of a chosen output variable against n (the number of replications). The user can then visually select the point on the graph where the cumulative mean line becomes "flat" and use this as the number of replications. This method has the advantage of being simple to understand and perform, as well as utilizing the output of interest in the decision made. It is however subjective with no measured precision level. The third main approach, Confidence Interval (with Specified Precision) Method, asks the user to make a judgment as to

how large an error they can tolerate in their model's estimate of the true mean. Replications are then run, and Confidence Intervals constructed around the sequential cumulative means, until this desired precision in the output is achieved. The advantage of this approach is that it relies upon statistical inference to determine the number of replications required. The disadvantage is that many simulation users do not have the skills to apply such an approach.

The Confidence Interval (with Specified Precision) Method was the one chosen to be adapted into an algorithm for automation and was then tested using artificial and real models. This method runs increasing numbers of replications until the confidence intervals constructed around the chosen output variable (e.g. mean queue length) using the t-statistic, are within a (user) specified precision. This allows the user to tailor the accuracy of output results to their particular requirement or purpose for that model and result. This method assumes that the cumulative mean has a normal distribution, (which is true under the Central Limit Theorem when the number of replications is large). We introduce and describe this simple algorithm in the next section. Section 3 explains the methodology used to test this algorithm on a selection of artificial and real models and Section 4 sets out the test results. We suggest an addition to the algorithm in Section 5. A discussion and final conclusions are found in Sections 6 and 7.

2 THE REPLICATION ALGORITHM

2.1 Overview and Definitions of Algorithm

This section sets out the algorithm, and describes and explains the definitions and mechanisms used. Figure 1 shows how the replication algorithm interacts with the model in a sequential way in order to ascertain the number of replications required. Any and all input data and parameter values are fed into the simulation model and the chosen output results produced. These model output results are input into the replication algorithm which calculates whether the required precision criteria set by the user have been met. If these have not, one more replication is made with the simulation model, and the new output value fed into the replication algorithm. This cycle continues until the algorithm's precision criteria are met.

We define the precision, d_n, as the ½ width of the Confidence Interval expressed as a percentage of the cumulative mean (Robinson 2004):

$$d_n = \frac{100 t_{n-1,\alpha/2} \, S_n / \sqrt{n}}{\overline{X}_n},$$

where n is the current number of replications carried out, $t_{n-1,\alpha/2}$ is the student t value for *n-1* degrees of freedom and a significance of *1-α*, $\overline{X}_n$ is the cumulative mean and S_n is the estimate of the standard deviation, both calculated using results X_i (i = 1 to n) of the n current replications.

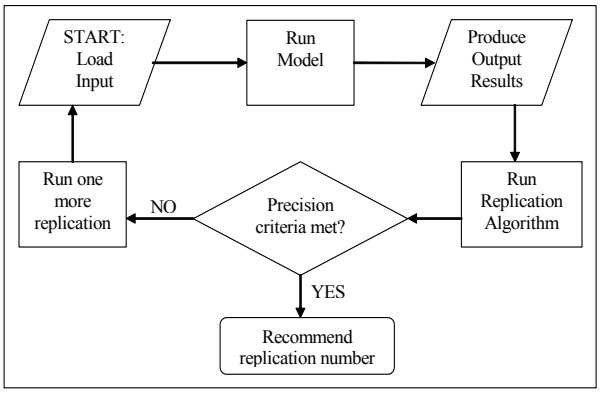

Figure 1: Flow diagram of the sequential procedure.

2.2 Stopping Criteria

The simplest method (Robinson 2004) is to stop as soon as d_n is first found to be less than or equal to the set desired precision, $d_{required}$ (user defined), and to recommend that number of replications to the user (Law 2007). But it is possible that the data series could prematurely converge to an incorrect estimate of the mean with precision $d_{required}$ by chance and then diverge again. In order to account for this possibility, when d_n is first found to be less than or equal to the set desired precision, $d_{required}$, the algorithm is designed to look ahead, performing a set number of extra replications, to check that the precision remains $\leq d_{required}$. We call this 'look ahead' value *kLimit*. The actual number of replications checked ahead is a function of this user defined value, *f(kLimit)*:

$$f(kLimit) = \begin{cases} kLimit, & n \leq 100 \\ \left\lfloor n \times kLimit / 100 \right\rfloor & n > 100 \end{cases}.$$

The function is constructed to allow the number of replications in the 'look ahead' to be in proportion with the current number of replications, n. Hence, when $n \leq 100$ the total number of replications checked ahead is simply *kLimit*. But when $n > 100$, the number of replications is equal to the proportion $\left\lfloor \frac{n \times kLimit}{100} \right\rfloor$. This has the effect of relating how far the algorithm checks ahead with the current value of n, while keeping the order of size consistent.

The number of replications taken to reach the desired precision is referred to as *Nsol* in the algorithm.

2.3 The Algorithm

The Replication Algorithm is as follows:

Let $n = 3$
Set $d_{required}$
Set *kLimit*

Run n replications of model.

While *Convergence Criteria* not met Do
- Calculate cumulative mean, ($\overline{X}_n$), confidence limits, and precision, (d_n).
- If $d_n \leq d_{required}$
  - Let *Nsol* be equal to n, the number of replications required to reach $d_{required.}$
  - If *kLimit* > 0
    - Calculate *f*(*kLimit*).
    - If the *Convergence Criteria* is met i.e. $d_{n+1}, \ldots, d_{n+f(kLimit)} \leq d_{required.}$
      Then
        Recommend *Nsol* replications to user
      Else
        Perform one more replication.

Loop

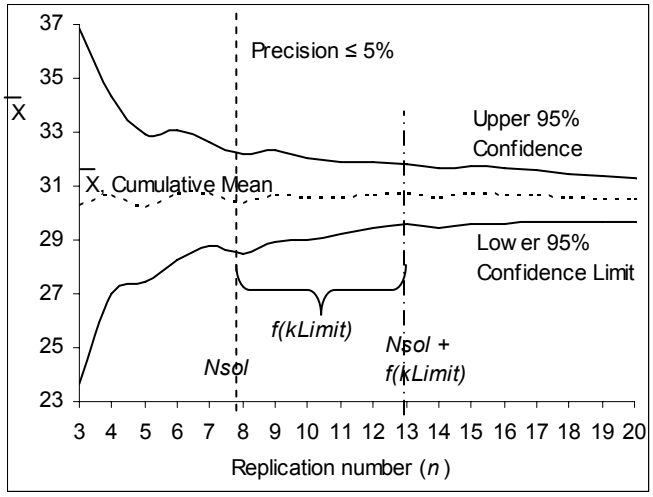

Figure 2: Graphical example of the replication algorithm with a kLimit 'look ahead' value of 5 replications.

Figure 2 is a visual example of the Replication Algorithm. In this example the *kLimit* 'look ahead' value is set at 5 and the required precision is set at 5% (i.e. $d_{required} = 5$). The first time the precision d_n is less than or equal to 5 occurs at $n = 8$ replications. *Nsol* is therefore set to 8 replications. Then, as $n < 100$, the algorithm continues for 5 more replications, calculating d_n at $n = 9$, 10, 11, 12 and 13 replications. If at any point d_n becomes greater than 5%, the 'look ahead' procedure is stopped. *Nsol* then becomes set to the next number of replications where $d_n \leq 5$ and the 'look ahead' procedure is repeated. In this visual example, the cumulative mean ($\overline{X}_n$) and Confidence Intervals are well behaved and d_n stays below 5% for the entire 5 'look ahead' replications. The algorithm, in this case, stops running after 13 replications and advises the user that 8 replications is sufficient (with these particular random streams) to achieve the required precision of 5%.

3 TESTING METHODOLOGY

In this section we will present the models and performance measures used to test the algorithm.

3.1 The Test Models

A large number of real simulation models (programmed in SIMUL8) were gathered together along with a collection of artificial models found in the literature. Real models with transient characteristics for specific output variables were identified. For each model an output measure was identified and multiple replications (2000) were performed. The mean of the identified output for each replication was calculated in order to create sets of mean values. These sets were then analysed for shape (symmetric, left or right skewed) and to see which statistical distribution would best fit (e.g. normal, beta,…). Artificial data was then created to mimic the general types of output found in the real models, the advantage being that the true mean and variance were known for these artificial data sets. For each artificial data set we created 100 sequences of 2000 data values. These 2000 data values were treated as a sequence of summary output values (e.g. mean through-put per hour) from 2000 replications of a simulation model.

The relative standard deviation (stdev/mean) of an output data set has a large impact on the number of replications required. The data sets created were therefore designed to cover a wide range of values for the relative standard deviation. The algorithm was first tested on this group of 24 artificial data sets, (13 non-Normal, 11 Normal), a representative sample of which are described in Table 1.

The algorithm was also tested with real model output. The models were selected to cover a variety of different output distributions and relative standard deviation values. A representative sample of these models is described in Table 2. For each real simulation model, we ran between 3000 and 11000 replications. For each replication we calculated the mean value of the chosen output variable for that model and then divided this large set of values into 100 separate sequences.

The algorithm was run with 100 different data sequences for each artificial and real output data set in order to produce a range of results that could be statistically analysed. The length of each data sequence was purposely set

Table 1: Description of a sample of the artificial models used in the testing of the replication algorithm's performance.

Model ID	Statistical Distribution	μ	σ/μ	Skewness	Theoretical *Nsol*
A2	Beta(20, 21, 1.44, 2.82)	20.3000	0.01015	0.502	2.62
A9	Beta(4, 2)	0.6667	0.26726	-0.468	112.18
A13	Bimodal{50% of data from Beta(1.5, 8), 50% of data from Beta(8, 1.5)}	0.5000	0.72000	0	798.99
A20	Normal(30, 3)	30.0000	0.10000		17.84
A22	Normal(30, 6)	30.0000	0.20000		63.90
A24	Normal(30, 21)	30.0000	0.70000		755.35

Table 2: Description of a sample of the real models used in the testing of the replication algorithm's performance.

Model ID	Model Description	Fitted statistical distribution	$\overline{X}$	$s/\overline{X}$	Skew-ness	Output variable	Theoretical *Nsol*
R2	An emergency call centre	Normal(88.6, 1.2696) or Beta(41.9094, 98.6797, 237.333, 51.2511)	88.6095	0.0144	-0.20	Percentage of 999 calls answered within 10 seconds per hour	2.85
R4	A university canteen at lunch period	Erlang(167.667, 13, 2.78095)	203.7564	0.0474	0.72	Time spent in system (secs) per customer	5.97
R5	A high street fast food restaurant	Pearson5(83.7852, 6.9201, 72.5332)	96.1651	0.0592	2.14	Time spent in queue for ordering, paying and collecting order (secs) per customer	7.91
R6	A call centre	Normal(30.717, 2.0083)	30.7282	0.0629		Through-put (completed calls) per hour	8.61
R8	A pub with kitchen & restaurant facilities	Bimodal	17.9544	0.1663		Time spent waiting for food to arrive (mins) per table of customers	44.93

to be far greater than the estimated *Nsol* value so that the algorithm did not have to end prematurely. The precision, d_n, was set at 5%.

3.2 Criteria for Assessing the Replication Algorithm

Five main performance measures were used to assess the algorithm results for each artificial data set:

1. *Coverage of the True Mean*: For each of the 100 times that the algorithm recommended a replication number (*Nsol*), the estimated mean value of the data set was also recorded. 95% confidence intervals were constructed around these mean values. The mean proportion of these 100 CIs that included the true mean value of the data set was calculated and a 95% confidence interval constructed around this mean proportion. We would wish that 95% of the time the true mean value does indeed fall inside our estimated 95% confidence intervals. It is therefore desirable that the value 95% be included in the confidence interval around the mean proportion.
2. *Bias*: The difference between the estimated mean and the true mean was calculated for each of the 100 algorithm runs. The mean of these values was then found and a 95% confidence interval constructed around the mean. For a 'good' estimate of the true mean value a small bias is required. If the estimated mean was equal to the true mean then the bias would be zero. It is therefore desirable that our constructed confidence interval contains the value zero.
3. *Absolute Bias*: The absolute difference between the estimated mean and the true mean was calculated for each of the 100 algorithm runs. The mean of these values was then found and a 95% confidence interval constructed around the mean. For the same reasons detailed above it is desirable that the absolute bias is as close to zero as possible.

Table 3: Results of testing the Algorithm with varying 'look ahead' periods on a representative sample of artificial models. Only the results that changed for each model as *kLimit* increased are included in the table.

kLimit	Model	Mean Bias with 95% CIs	Mean Absolute Bias with 95% CIs	95% CIs for coverage of true mean	Average *Nsol* with 95% CIs - Does the CI contain theoretical *Nsol*?
None	A2	**-0.0390 ± 0.0230**	0.0987 ± 0.0142	0.95 ± 0.0432	3.01 ± 0.0198 No
	A9	-0.0008 ± 0.0036	0.0139 ± 0.0023	0.95 ± 0.0432	112.91 ± 3.9229 Yes
	A13	0.0002 ± 0.0022	0.0101 ± 0.0008	1 ± 0.0000	799.32 ± 7.5839 Yes
	A20	-0.0323 ± 0.1600	0.6390 ± 0.0970	0.93 ± 0.0506	16.24 ± 1.1919 No
	A22	-0.1592 ± 0.1834	0.6523 ± 0.1331	0.93 ± 0.0506	62.52 ± 3.0770 Yes
	A24	-0.2062 ± 0.4055	0.7897 ± 0.3760	0.93 ± 0.0506	747.05 ± 18.2462 Yes
5	A9	0.0005 ± 0.0032	0.0129 ± 0.0018	0.98 ± 0.0278	114.73 ± 3.2478 Yes
	A13	0.0005 ± 0.0022	0.0103 ± 0.0008	No change	800.64 ± 7.6900 Yes
	A20	-0.0163 ± 0.1375	0.5539 ± 0.0819	0.95 ± 0.0432	17.82 ± 1.0024 Yes
	A22	-0.0333 ± 0.1403	0.5624 ± 0.0845	0.97 ± 0.0338	64.98 ± 2.4821 Yes
	A24	0.0088 ± 0.1542	0.6053 ± 0.0959	No change	757.39 ± 10.4073 Yes
10	A9	0.0008 ± 0.0032	0.0129 ± 0.0019	No change	115.07 ± 3.2798 Yes
	A22	-0.0335 ± 0.1402	0.5622 ± 0.0845	No change	65.28 ± 2.4645 Yes
25	A20	-0.0178 ± 0.1370	0.5524 ± 0.0815	No change	17.83 ± 1.0003 Yes

4. *Comparison to Theoretical Nsol Value*: The number of replications that our algorithm would have recommended assuming that the true mean and standard deviation were known throughout, was calculated for each different artificial data set. These theoretical *Nsol* values were produced in such a way as to provide a fair comparison with the algorithm estimates of *Nsol*. To achieve this they were calculated iteratively using the same methodology as employed by the replication algorithm. The only difference is that the theoretical values were calculated using the true values for the mean and standard deviation rather than estimated values. Because the number of replications, *n*, used in the calculations had to be an integer the theoretical value of *Nsol* often fell between two values of *n*. Linear interpolation was therefore used to refine these values. These theoretical *Nsol* values could then be compared with the *Nsol* values produced by the algorithm where the mean and standard deviation had to be estimated from the data available.
5. *Average Nsol*: The number of replications recommended (*Nsol*) by the algorithm for each of the 100 runs was recorded and the average value calculated. A 95% confidence interval was then constructed around this mean value. It is desirable that the theoretical value of *Nsol* for that particular artificial data set lies inside this confidence interval.

The same performance criteria were used when the algorithm was applied to the real models but the 'true' mean and standard deviation values were estimated from the whole sets of output data (3000 to 11000 data points). This, we believe, is a sufficiently accurate measure of the true mean for the purposes of comparison with the algorithm results.

4 TEST RESULTS

In this section we will present the results of testing the replication algorithm on a variety of artificial and real models. It will answer the questions: How well does this algorithm perform (for large and small *n*)? Is the 'look ahead' procedure required? And if so, which value of *kLimit* is 'best'?

The algorithm performs as expected. Although *Nsol* values for individual algorithm runs are very variable (as is to be expected using random streams), the average values for 100 runs per model were close to the theoretical values of *Nsol*. When using a 'look ahead' of 5 or more, the coverage of the true mean was 95% or greater for all artificial and real models. The bias (difference between true mean and estimated mean values) was small on average and the 95% confidence intervals included zero for all models when a 'look ahead' was used. Results for a sample of artificial models are found in Table 3.

Using a 'look ahead' period improves the estimate of mean model performance. Without a 'look ahead' period, 4 of the 24 artificial models tested produced a mean bias that was significantly different to zero, as did one of the real models. Two of the artificial models failed in the coverage of the true mean. There is also one specific example in the real models, (model R8 with a bimodal output), where the coverage of the true mean does not include the proportion 95. This is rectified by using a look ahead period of 5 replications. For the artifical models, half the average estimated *Nsol* values were significantly different to the theoretical *Nsol* values (for those models with theoretical Nsol values > 3). This fell to just one model when a

'look ahead' was used. Similar results were found using the real models. Using a 'look ahead' value of 5 ensured that only one artificial model had a bias that was significantly different to zero and that no model (real or artificial) failed in the coverage of the true mean. The percentage decrease in absolute mean bias from using no 'look ahead' period to one with a length of 5 was 8.76% for the artificial models. The percentage decrease in absolute mean bias from using a 'look ahead' period of 5 to one with a length of 10 was just 0.07% and for 10 to 25 there was a decrease of 0.26%. There was a similar result for the real models. Therefore, looking ahead just 5 replications can significantly improve bias and the coverage of the true mean. Specific examples of this occurrence are shown in Table 4 for a variety of artificial and real models.

Table 4: Specific examples of changes in *Nsol* values and improvement in coverage of the true mean, for increasing values of 'look ahead' period (*kLimit*)

Model	kLimit	Nsol	Theoretical Nsol	CI contains true mean value?
A22	0	4	64	no
	5	54		yes
A9	0	4	112	no
	5	120		yes
A24	0	3	755	no
	5	718		yes
A20	0	3	18	no
	5	22		yes
R4	0	3	6	no
	5	7		yes
R6	0	3	9	no
	5	11		yes
R8	0	3	45	no
	5	46		yes

When counting the number of times the *Nsol* value changes by using a look ahead period of 5 replications compared with none, it was found that only 5 out of all 32 real and artificial models showed no changes at all. The greatest number of changes out of the 100 runs per model was 37. In contrast, when looking similarly at the difference between using a 'look ahead' of 5 and 10, the number of models not showing any change rises to 21. The greatest number of changes out of 100 runs per model in this case is only 8. Similarly, when looking at the difference between using a 'look ahead' of 10 and 25, the number of models not showing any change is 20 and the greatest number of changes out of 100 runs per model is just 3. All these results indicate that a *kLimit* of 5 has the largest impact and is the obvious choice as a default value for this parameter.

It was found that for all models the theoretical *Nsol* values all lie within the 95% confidence limits for the mean estimated *Nsol* values when a 'look ahead' value of 5 replications was used. The only exception was where both estimated and theoretical *Nsol* values are close to 3 (e.g. Model A2). This is due to the initial n value in the algorithm being artificially set to 3. This stops the algorithm from choosing *Nsol* values of 2 and hence biases the average results.

In validating the algorithm it was confirmed that the estimated *Nsol* results did not violate the underlying mathematical correlations between the various formulas used to calculate precision d_n and *Nsol* in the algorithm. There is a quadratic correlation between the average *Nsol* values for each model and the true standard deviation of the output in relation to the true mean (i.e. relative $\sigma = \sigma/\mu$).

The statistics used to construct the confidence intervals around the cumulative mean assume that the cumulative mean has a normal distribution, (which is true under the Central Limit Theorem when the number of replications is large). For small n and non-normal data it is therefore possible that the confidence intervals around the cumulative mean produced by the algorithm are not accurate. However, in practice there did not seem to be an obvious problem with the algorithm results for small n. But should this prove to be a problem it may be possible to use other techniques to construct the confidence intervals when n is small, such as Bootstrapping (Cheng 2006) and Jack Knife processes (Shao and Tu 2006), (although it is thought that these procedures would still struggle with very small n, e.g. $n<10$).

5 INCORPORATING A FAIL SAFE INTO THE ALGORITHM

If the model runs 'slowly' the algorithm could take an unacceptable amount of time to reach the set precision. We therefore recommend that a 'fail safe' is incorporated into the replication algorithm to warn the user when a model may require a 'long time' to reach $d_{required}$. At each iteration of the algorithm an estimate of *Nsol* can be calculated using the formula (Banks et al. 2005):

$$Nsol^* = \left\lceil \frac{100 t_{n-1,\alpha/2} S_n}{\overline{X}_n d_{required}} \right\rceil^2 .$$

However, this estimate is only as accurate as the current estimate of the standard deviation and mean. It has been found that this estimate can be very inaccurate for small n (Law 2007, Banks et al. 2005). Figure 3 shows a range of typical behaviour of $Nsol^*$ values. From investigating the values of $Nsol^*$ generated using different artificial models (where *Nsol* is 'large') it seems unlikely that this estimate of *Nsol* is valid for values of $n < 30$.

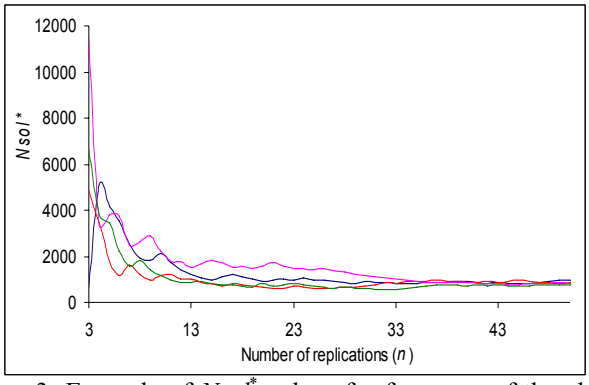

Figure 3: Example of $Nsol^*$ values for four runs of the algorithm on the artificial Bimodal data set (A13).

It is therefore proposed that a useful aid to the user would be a graph of the changing value of $Nsol^*$ continually updated as the algorithm progresses (similar to Figure 3). The user could then make a judgment as to whether to let the algorithm progress naturally or to terminate prematurely.

6 DISCUSSION

We advise that the default value for the algorithm parameter *kLimit* is best set at five. This is due to the fact that in testing, the majority of premature convergence problems were solved by looking ahead just five extra replications. Further experimentation also confirmed that this was still the case when using different values of required precision. Five is also thought to not be a prohibitively large number of extra replications to perform. We advise though that the user should be given the opportunity to alter the *kLimit* value if they wish.

The initial number of replications was set to 3 due mainly to the fact that the SIMUL8 simulation software used in this research has this value as their minimum number of replications. It was shown in testing using the real models that small numbers for *Nsol* were often found and therefore it is deemed sensible to keep the initial number of replications as small as 3.

It has also been suggested that "stability" of the cumulative output variable should be considered as well as precision in a stopping criteria (Robinson 2004). It has been proposed that it is not only important that the confidence interval is sufficiently narrow but also that the cumulative mean line is reasonably 'flat'. We therefore incorporated this extra stability criteria into our algorithm to produce an 'extended replication algorithm'. In order to check for 'flatness' of the cumulative mean line the extended algorithm 'draws' two parallel lines called Inner Precision Limits (IPLs) around the cumulative mean line. These are defined as a percentage of the $d_{required}$ value. If the cumulative mean crosses either IPL within the 'look ahead' period, then the stability criteria is violated. This new algorithm was tested on the real and artificial models to see what effect this extra stability criteria would have on replication number estimation. It was found that the added stability criteria did not significantly enhance the accuracy of the estimate of mean performance. It did not significantly reduce bias in the mean estimate. It was found that the extra stability procedure caused the replication algorithm to be unnecessarily complicated and could cause confusion in the user. When the stability criteria is invoked it causes the final *Nsol* recommendation to be associated with a much smaller precision than the user requested because n is greater, but does not significantly reduce bias in the mean estimate. Equivalent results can be produced by simply setting a smaller $d_{required}$, which is much more easily understood by a user. The extra stability criteria was therefore dropped from the replication algorithm.

The relative standard deviation value (i.e. standard deviation (σ) / mean (μ)) is the predominant cause for the size of the *Nsol* value. The distribution and skewness of the data do not appear to have a large effect in their own right but of course do effect the value of the relative standard deviation itself and therefore may be important in that respect.

In practice it is likely that a user will be interested in more than one response variable. In this case the algorithm should ideally be used with each response and the model run using the maximum estimated value for *Nsol*. Likewise, in theory, when running different scenarios with a model, all output analysis including replication number estimation should be repeated for each scenario. This is however fairly impractical and can cause problems in using different n when comparing scenarios. But it is advisable to repeat the algorithm every few scenarios to check that the precision has not degraded significantly.

7 CONCLUSION

This paper describes the selection and automation of a replication algorithm for estimating the number of replications to be run in a simulation. The Simple Algorithm created is efficient and has been shown to perform well on a wide selection of artificial and real model output. It is 'black box' in that it is fully automated and does not require user intervention. We recommend though that as an added feature a graph of the changing estimate of *Nsol* with each iteration of the algorithm (*Nsol**) be shown to the user so that they can decide to prematurely end the algorithm if it is going to take a prohibitively long amount of time to reach the required precision. It is also advised that if this algorithm is included into a simulation package that full explanations of the algorithm and reported results (e.g. Figure 2) are included for the user to view if desired.

ACKNOWLEDGMENTS

This work is part of the Automating Simulation Output Analysis (AutoSimOA) project <http://www.wbs.ac.uk/go/autosimoa> that is funded by the UK Engineering and Physical Sciences Research Council (EP/D033640/1). The work is being carried out in collaboration with SIMUL8 Corporation, who are also providing sponsorship for the project.

REFERENCES

Banks, J., J. S. Carson II, B. L. Nelson, and D. M. Nicol. 2005. *Discrete-event system simulation.* 4th ed. Pearson Prentice Hall.

Cheng, R. C. H. 2006. Validating and comparing simulation models using resampling. *Journal Of Simulation* 1(1).

Hlupic, V. 1999. Discrete-event simulation software: What the users want. *Simulation* 73 (6):362-370.

Hollocks, B. W. 2001. Discrete-event simulation: An inquiry into user practice. *Simulation Practice and Theory* 8:451-471.

Law, A. M. 2007. *Simulation modeling and analysis.* 4th ed. McGraw-Hill.

Law, A M. and M. McComas. 1990. Secrets of successful simulation studies. *Industrial Engineering* 22(5):47-72.

Robinson, S. 2004. *Simulation; The practice of model development and use*. John Wiley & Sons Ltd..

Shao, J and D. Tu. 2006. *The Jackknife and Bootstrap (Springer Series in Statistics)*. Springer-Verlag.

AUTHOR BIOGRAPHIES

KATHRYN HOAD is a research fellow in the Operational Research and Information Systems Group at Warwick Business School. She holds a BSc in Mathematics and its Applications from Portsmouth University, an MSc in Statistics and a PhD in Operational Research from Southampton University. Her e-mail address is <kathryn.hoad@wbs.ac.uk>

STEWART ROBINSON is a Professor of Operational Research and the Associate Dean for Specialist Masters Programmes at Warwick Business School. He holds a BSc and PhD in Management Science from Lancaster University. Previously employed in simulation consultancy, he supported the use of simulation in companies throughout Europe and the rest of the world. He is author/co-author of three books on simulation. His research focuses on the practice of simulation model development and use. Key areas of interest are conceptual modelling, model validation, output analysis and modelling human factors in simulation models. His email address is <stewart.robinson@warwick.ac.uk> .

RUTH DAVIES is a Professor of Operational Research in Warwick Business School, University of Warwick and is head of the Operational Research and Information Systems Group. She was previously at the University of Southampton. Her expertise is in modeling health systems, using simulation to describe the interaction between the parts in order to evaluate current and potential future policies. Over the past few years she has run several substantial projects funded by the Department of Health, in order to advise on policy on: the prevention, treatment and need for resources for coronary heart disease, gastric cancer, end-stage renal failure and diabetes. Her email address is <ruth.davies@wbs.ac.uk> .