www.palgrave-journals.com/jos/

AutoSimOA: a framework for automated analysis of simulation output

K Hoad*, S Robinson and R Davies

Operational Research and Management Sciences Group, Warwick Business School, University of Warwick, Coventry, UK

There are two key issues in assuring the accuracy of estimates of performance obtained from a simulation model. The first is the removal of any initialisation bias; the second is ensuring that enough output data are produced to obtain an accurate estimate of performance. Our aim is to produce an automated procedure for inclusion into commercial simulation software to address both of these issues. This paper describes the results of a 3-year project to produce such an analyser. Our Automated Simulation Output Analyser identifies the warm-up period, estimates the number of replications, and/or analyses output from a single run, with the aim of providing the user with accurate and precise measures of their chosen output statistics.

Journal of Simulation (2011) **5,** 9–24. doi:10.1057/jos.2010.22; published online 12 November 2010

Keywords: discrete-event simulation; simulation output analysis; initialisation bias; warm-up; multiple replications; batch means methods

1. Introduction

This paper describes a framework for automatically analysing the output from a simulation model. The aim of the framework is that it should be implemented in commercial simulation software with a view to improving the use of simulation, particularly by non-expert simulation users. The analyser, known as AutoSimOA (*Auto*mated *Sim*ulation *O*utput *A*nalyser), provides a sequential procedure that identifies the warm-up period, determines the number of replications, and/or analyses the output from a single (long) run, with the aim of providing accurate (low bias) and precise (low variance) measures for the output statistics that are of interest to the simulation users. The analyser is for use with terminating and non-terminating (steady-state) simulations, and deals with initialisation bias when it is present.

The paper starts by discussing the need for automated analysis of simulation output. Following this the problem of analysing the output from a simulation is stated mathematically. The overall framework is then described, followed by more detailed descriptions of the three components of AutoSimOA: the warm-up analyser, the replications calculator and the single run analyser. AutoSimOA is demonstrated by applying it to the output of two different models. The paper concludes with a discussion on the implementation of the AutoSimOA framework.

There is no attempt in this paper to justify the methods used as part of AutoSimOA through either empirical testing or mathematical proof. The testing of the recommended methods has been reported in detail elsewhere. References to these papers are provided when discussing the individual methods in AutoSimOA.

2. The requirement for automated analysis of simulation output

Visual interactive modelling systems, first seen in the late 1980s, placed simulation model development into the hands of non-experts by removing the need for a detailed knowledge of programming code. Today, discrete-event simulation is in widespread use, being applied in areas such as manufacturing design and control, service system management (eg call centres), business process design and management, and health applications.

The prevalence of simulation software and its adoption by non-experts has almost certainly led to significant problems with the use of the simulation models that are being developed. The appropriate analysis of simulation output requires specific skills in statistics that many non-experts do not possess. Decisions need to be made about initial transient problems, the length of a simulation run, the number of independent replications that need to be performed and the selection of scenarios (Robinson, 2004; Law, 2007). Appropriate methods also need to be adopted for reporting, comparing and ranking results. The majority of simulation packages only provide guidance over the selection of scenarios through simulation ‘optimisers’ (Law and McComas, 2002). Other decisions are left to the user with little or no help from the software. As a result, it is likely that many simulation models are being used poorly. Indeed,

**Correspondence: Operational Research and Management Sciences Group, Warwick Business School, University of Warwick, Coventry, CV4 7AL, UK.*

Hollocks (2001), in a survey of simulation users, provides evidence to support the view that simulations are not well used. The consequences are that incorrect conclusions might be drawn, at best causing organisations to forfeit the benefits that could be obtained, and at worst leading to significant losses with decisions being made based on faulty information.

Alongside developments in simulation software and simulation practice, theoretical developments in the field of simulation output analysis have continued. Many of these developments are reported at the annual Winter Simulation Conference, which has a stream dedicated to the subject (eg Mason *et al*, 2008). The focus of the work reported, however, is largely on theoretical developments rather than practical application. For instance, a survey of research into the initial transient problem and methods for selecting a warm-up period found some 44 methods (Hoad *et al*, 2009b). None of the methods, with the possible exception of simple time-series inspection and Welch's method (Welch, 1983), appear to be in common use.

Three problems seem to inhibit the use of output analysis methods:

- Most methods have been subject to only limited testing, giving little certainty as to their generality and effectiveness.
- Many of the methods require a detailed knowledge of statistics and therefore are difficult to use, especially for non-expert simulation users.
- Simulation software do not generally provide implementations of the methods.

One solution to these problems is to implement an automated output analysis procedure in simulation software. This would overcome to a significant degree the problem of the need for statistical skills. The development of such a procedure requires the thorough testing, and where necessary adaptation, of candidate methods. Apart from the work described here, we are only aware of one other research effort on automated analysis, the Akaroa project (see http://www.cose.canterbury.ac.nz/research/RG/net_sim/simulation_group/akaroa, accessed March 2009).

A 3-year research project has been undertaken by the authors to explore the potential to automate the analysis of simulation output and to ultimately create an automated simulation output analysis tool. The work focused on three specific areas: selecting a warm-up period, determining the number of replications and analysing the output from a single (long) run. In doing so, the work only focused on automating the analysis of output from a single scenario. In carrying out this project the authors searched the literature for analysis methods; where necessary, undertook thorough testing of candidate methods; adapted methods to make them suitable for automation; and proposed an overall procedure for automated analysis. The selection and testing of methods have been reported in detail elsewhere (Hoad *et al*, 2009a, b). This paper focuses on the overall framework that was devised as a result of this work.

3. Statement of the problem

Our aim is to develop an automated procedure that obtains unbiased estimators of the population mean and variance (μ and σ^2, respectively) for one or more simulation output statistics.

3.1. Output data from simulation

We start with the case of a single output statistic, y. From a single run of the model the simulation generates a data series for y as follows: $y_1 = y_{1,1},\ y_{1,2}, \ldots, y_{1,M}$, where the first index refers to the number of the run (ie 1 for a single run) and M is the number of observations (run-length) obtained from the simulation. We wish to estimate μ, the mean of y, where $\mu = \underset{j\to\infty}{Lim}\,E(y_{1,j})$ and σ^2, the variance of y, where $\sigma^2 = E(y_1^2) - \mu^2$. The early data in the series y_1 may not be in steady-state (if y_1 reaches steady-state at all) due to the starting condition of the simulation. Given that we wish to estimate the mean value of y from the data in y_1, then these early data will bias the estimate of the mean. This phenomenon is known as 'initialisation bias'. The initial data are considered to have a significant bias on the estimate of the mean if a confidence interval, with significance level α, constructed from the data in y_1 does not give the expected coverage (ie $1-\alpha$) of μ. The point at which the initial data cease to have a significant bias on the estimate of the mean is denoted as L. According to Law (2007, p 509) the estimate of the mean may differ significantly from μ if L and M are too small. The position of L depends in part on the run-length (M) of the simulation; a longer run-length implies that L can have a smaller value, since the greater remaining bias is subsumed into more steady-state data. In recognising the existence of initialisation bias we define the series y_1 as consisting of a series of data, $y_{1,1}, \ldots,\ y_{1,L}, y_{1,L+1}, \ldots, y_{1,M}$.

Observations for y_1 might be time based (eg hourly throughput) or by entity (eg time in system for each entity). The values of y_1 might be in steady-state throughout the length of the simulation run ($L = 0$); the series might be subject to significant initialisation bias prior to reaching steady-state ($0 < L < M$); or the series might be in a transient state throughout, either because it has not yet reached a steady-state ($L \geqslant M$), or because the output data do not have a steady-state. Assuming a steady-state is reached, the data series y_1 then follows some unknown distribution $F(y)$.

It is common practice in simulation to carry out multiple independent replications by running the model with different streams of random numbers. Given that N replications are performed with the model, then N data series for y (denoted y_n) are created. As such, the simulation generates a matrix of output data $\mathbf{Y}$ as follows:

$$\mathbf{Y} = \begin{pmatrix} y_{1,1}, & \cdots & ,y_{1,L_1}, & y_{1,L_1+1}, & \cdots & ,y_{1,M_1} \\ y_{2,1}, & \cdots & ,y_{2,L_2}, & y_{2,L_2+1}, & \cdots & ,y_{2,M_2} \\ \cdot & & \cdot & \cdot & & \cdot \\ \cdot & & \cdot & \cdot & & \cdot \\ \cdot & & \cdot & \cdot & & \cdot \\ y_{N,1}, & \cdots & ,y_{N,L_N}, & y_{N,L_N+1}, & \cdots & ,y_{N,M_N} \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \cdot \\ \cdot \\ \cdot \\ y_N \end{pmatrix}$$

where the rows denote the series of data generated for y from replication $n = 1, 2, \ldots, N$. Because L and M can be different for each individual replication, they are given an index n.

A final extension would be to add the case where we are interested in multiple outputs from the simulation model. This can be achieved by extending the notation to the form y_n^p, where $p = 1, 2, \ldots, P$ and denotes the index of the output statistic of interest. This would lead to P output matrices, $\mathbf{Y}^p$. For now we shall only consider the case of a single output statistic.

3.2. Warm-up estimation

Where the data series y_n are subject to initialisation bias it is useful to delete the biased data $(y_{n,1}, \ldots, y_{n,L_n})$ prior to further analysis. It should be noted that if the run-length (M_n) is sufficient to warrant that the initialisation bias in $(y_{n,1}, \ldots, y_{n,L_n})$ is insignificant, in other words $(y_{n,1}, \ldots, y_{n,M_n})$ provides an unbiased estimator of μ, then deletion is not necessary at all.

Initialisation bias often fades asymptotically into the steady-state, that is, the steady-state distribution of $F(y_n)$ is only obtained as $M_n \to \infty$. In practical terms, given the variance in the data, satisfactory estimates of μ and σ^2 can be obtained from data that are subject to negligible bias. Hence, the location of the point L_n can be quite early in the output data series, since the requirement is only for the data beyond L_n to have an insignificant bias on the estimates of μ and σ^2.

Given the asymptotic nature of the initialisation bias and the variance in the output data, it is extremely difficult to identify L_n exactly. Therefore, a warm-up analysis method should aim to identify a deletion, or truncation, point $(\widehat{L_n})$ such that any difference between $\widehat{L_n}$ and L_n has a minimal impact on the estimation of μ and σ^2. As noted above, the significance of the bias caused by any remaining data between $\widehat{L_n}$ and L_n depends on the run-length M_n. It is, of course, desirable to avoid the situation where $\widehat{L_n} \gg L_n$, since this wastes data and time for running the simulation. In any practical simulation, L_n will be unknown and therefore the warm-up method has to be tested on artificial data with known parameters.

3.3. Replications method

One means for obtaining estimators of μ and σ^2 is to analyse the output from multiple independent replications (referred to as the replications/deletion method) (Law, 2007). We can summarise each row of the matrix $\mathbf{Y}$ with the mean of the data series y_n as follows:

$$X_n = \frac{\sum_{m=\hat{L}_n+1}^{M_n} y_{n,m}}{M_n - \hat{L}_n}, \qquad \text{for } n = 1, \ldots, N \tag{1}$$

where X_n represents the mean of the output data series beyond the deletion point $(\widehat{L_n})$. It is expected that $\widehat{L_n}$ would normally be given the same value for all n, as would M_n.

Given that $X_1, X_2, \ldots, X_n$ are independent and identically distributed (IID) observations, then, following Law (2007), the sample mean of the X_ns:

$$\bar{X}(N) = \frac{\sum_{n=1}^{N} X_n}{N} \tag{2}$$

is an unbiased estimator for μ, that is $\mu = \underset{M_n \to \infty}{Lim} E[X_n]$. Similarly, the sample variance:

$$s^2(N) = \frac{\sum_{n=1}^{N} [X_n - \bar{X}(N)]^2}{N-1} \tag{3}$$

is an unbiased estimator of the population variance σ^2, for fixed M_n. Hence an approximate confidence interval for $\bar{X}(N)$ can be constructed using:

$$\bar{X}(N) \pm t_{N-1,\alpha/2} \sqrt{\frac{s^2(N)}{N}} \tag{4}$$

where $t_{N-1,\alpha/2}$ is the value from the Student's t-distribution with $N-1$ degrees of freedom and significance level $\alpha/2$. This assumes that N is large enough such that the distribution of $\bar{X}(N)$ can be assumed to be normal under the Central Limit Theorem, and that $(N-1)\frac{s^2}{\sigma^2} \sim \chi^2_{N-1}$ (Cochran, 1934).

3.4. Batch means method

As an alternative to multiple independent replications, the results from a single long run (eg y_1) of a simulation can be analysed. In this case the value of $X_1 \to \mu$ as $M_1 \to \infty$. However, estimating the variance is more problematic due to the likely autocorrelation in the data $(y_{1,1}, \ldots, y_{1,M})$. Autocorrelation violates the assumption of independence in the data, thus biasing the usual statistical estimates. Hence, other analysis techniques need to be employed. Various methods can be adopted for such an analysis including autoregressive methods (Fishman, 1971; Fishman, 1973), spectral analysis (Heidelberger and Welch, 1981) and regenerative methods (Crane and Iglehart, 1974a, b, 1975; Crane and Lemoine, 1977; Fishman, 1977; Lavenberg and Sauer, 1977). Here we use the batch means approach (Conway, 1963; Fishman, 1978) since this is deemed suitable for automation.

In the batch means method we calculate a series of batch means from the series y_1 (or indeed, any other single series, y_n) as follows:

$$Y_j(k) = \frac{\sum_{m=1+(j-1)k+\widehat{L}_1}^{jk+\widehat{L}_1} y_{1,m}}{k}, \quad \text{for } j = 1, 2, \ldots, b \quad \text{where } b = \left\lfloor \frac{M_1 - \widehat{L}_1}{k} \right\rfloor \tag{5}$$

and b is the number of batches and k is the size of (number of observations in) each batch.

Given a sufficient batch size (k) to assure the approximate independence of the $Y_j(k)$, then these may be treated as IID observations (assuming that no initial bias is present). It then follows that

$$\bar{Y}(b, k) = \sum_{j=1}^{b} \frac{Y_j(k)}{b} \tag{6}$$

is an unbiased estimator for μ, that is $E[\bar{Y}(b,k)] = \mu$. The sample variance of the $Y_j(k)$s and confidence interval for μ can then be calculated as for the replications method above, by substituting $Y_j(k)$ and $\bar{Y}(b,k)$ for X_n and $\bar{X}(N)$, respectively, in Equations (2)–(4).

However, assuring the independence of $Y_j(k)$ is not a trivial matter and as a result more complex procedures for determining the batch size (k) and estimating μ and σ^2 are required. These are discussed later in the section entitled 'The single run analyser'.

4. Overview of the AutoSimOA framework

The overall framework for AutoSimOA is shown in Figure 1. The analyser consists of three main components: the warm-up analyser, the replications calculator and the single run analyser. More detailed descriptions of each component are provided in turn in the next three sections.

The user is first asked to choose between running the model using multiple replications or one (long) run. The choice defines the path that is taken through AutoSimOA. Having chosen the run strategy, the user is asked whether a warm-up analysis should be performed or not. There is no attempt within AutoSimOA to make this choice, although the warm-up analysis (if selected) should indicate if, in fact, little or no warm-up is required.

The warm-up analysis may be performed on a single run or on the accumulated results from multiple replications. If multiple replications are available, then Hoad *et al* (2009b) show that this is the preferred approach. If, in the run strategy, the user indicates a preference for multiple replications, then the results in the warm-up analysis are carried out on data averaged over multiple replications (with a default of five replications). The replications calculator is then used on the truncated data to estimate how many replications ($\widehat{N}$) need to be run to achieve the desired precision for the estimate ($\bar{X}(N)$) of the output statistic of interest (X_n). The user specifies a fixed run-length (M) and the same truncation point ($\widehat{L}$) is used for all individual replications. Albeit that different truncation points could be used for every individual replication, following separate analyses of each replication, practically it is simpler to opt for a single point.

If the user chooses to perform just one run, this causes AutoSimOA to base its warm-up analysis on a single run. The user is then given the choice of running the model for a set time period (M is specified by the user) and analysing the data produced using the batch means calculator, or allowing the run-length calculator component to choose a run-length ($\widehat{M}$), with or without a precision requirement for the confidence interval of μ based on the overall sample mean $\bar{Y}(b,k)$.

AutoSimOA aims to automate as much of the process as possible. At present, however, certain decisions continue to be made by the user. These are: whether to perform one run or multiple replications, whether to analyse for warm-up and whether to use a set-run-length or not. These decisions require an understanding of the model, the system being modelled and the context within which the model is being used.

5. The warm-up analyser (Al in Figure 1)

The warm-up analyser uses the MSER-5 heuristic (White, 1997; White *et al*, 2000). This method was chosen following a thorough review of the literature on the initial transient problem in which 44 warm-up methods were identified. These methods were then evaluated for their fit with the requirement for an automated procedure and candidate methods were tested on example data. MSER-5 was tested on over 3000 data sets, from which it was concluded that the heuristic was the most robust automatable method. The selection and testing of warm-up methods is described in Hoad *et al* (2009b).

In order to implement MSER-5 as an automated procedure we have devised a heuristic framework around the method. This framework, which involves an iterative (sequential) procedure, is shown in Figure 2. The user specifies the output variables for which a warm-up analysis should be performed. There can be as many variables as the user wishes to specify. The warm-up analyser treats each variable independently,

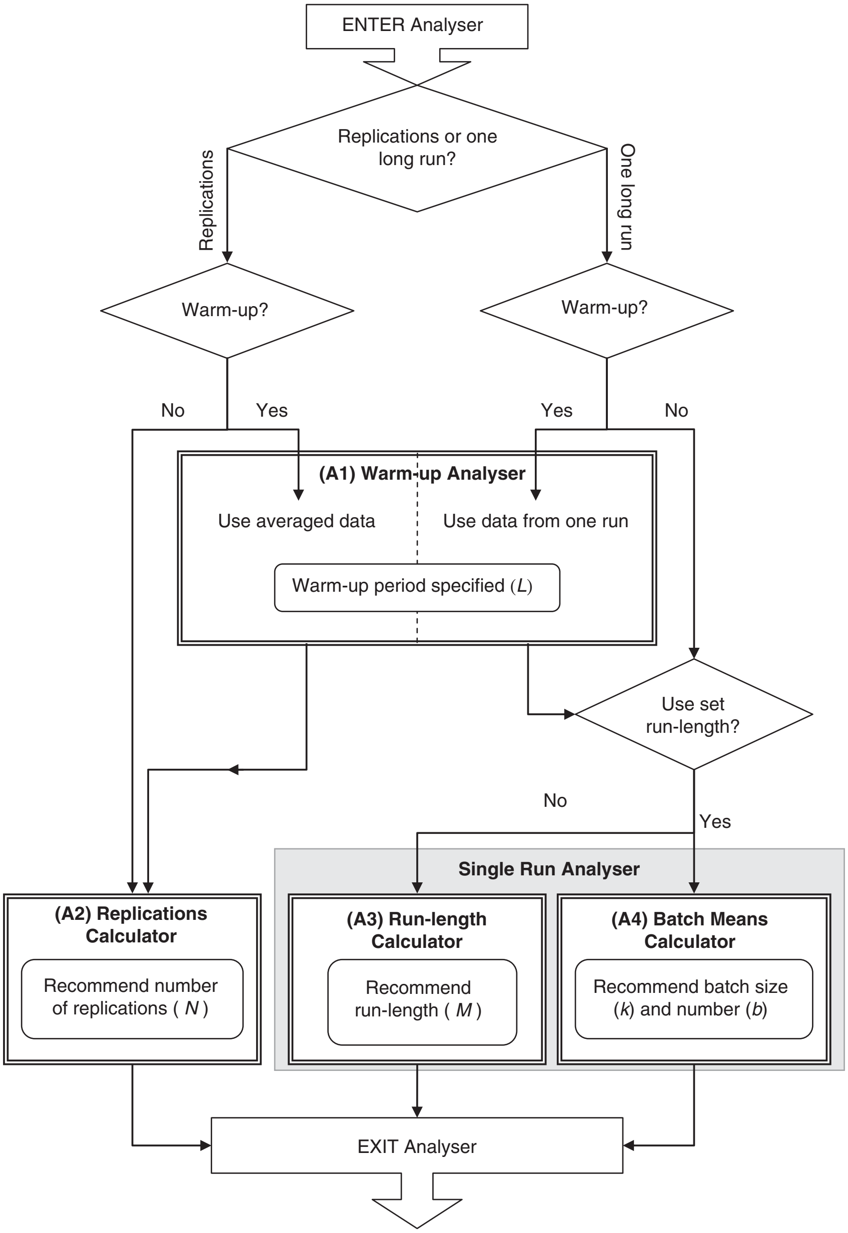
ENTER Analyser
Replications or one long run?
Replications
One long run
Warm-up?
Warm-up?
No
Yes
Yes
No
(A1) Warm-up Analyser
Use averaged data
Use data from one run
Warm-up period specified (*L*)
Use set run-length?
No
Yes
Single Run Analyser
(A2) Replications Calculator
Recommend number of replications (*N*)
(A3) Run-length Calculator
Recommend run-length (*M*)
(A4) Batch Means Calculator
Recommend batch size (*k*) and number (*b*)
EXIT Analyser

Figure 1 The overall framework for AutoSimOA.

recommending individual warm-up periods. For reasons of simplicity, however, the description that follows refers to only one output variable.

If the user selects multiple replications as the run strategy for the simulation model (see the section entitled 'The replications calculator'), then the warm-up analyser will run multiple replications ($N=5$ is the default value) and generate a single series of data ($\bar{y}$) consisting of the average values across the replications:

$$\bar{y} = \frac{\sum_{n=1}^{N} y_{n,1}}{N}, \frac{\sum_{n=1}^{N} y_{n,2}}{N}, \ldots, \frac{\sum_{n=1}^{N} y_{n,M}}{N}, \quad \text{for } M \geqslant 100 \qquad (7)$$

Our tests with MSER-5 (Hoad *et al*, 2009b) demonstrated that the method is more robust when applied to data from multiple replications. However, if the user selects a single run as the run strategy, then MSER-5 only uses the data from that single run, that is y_1.

The data set $\bar{y}$ or y_1 is batched into the maximum possible number of batches, b, of length k (with default $k=5$ as per MSER-5), that is $\lfloor M/k \rfloor = b$. A data series consisting of b batch means is therefore generated. It is these data that are input to the MSER heuristic.

An initial run-length (M) of at least 100 is required in order to provide the analyser with sufficient data to start the procedure (ie this provides at least 20 batches of length 5 to

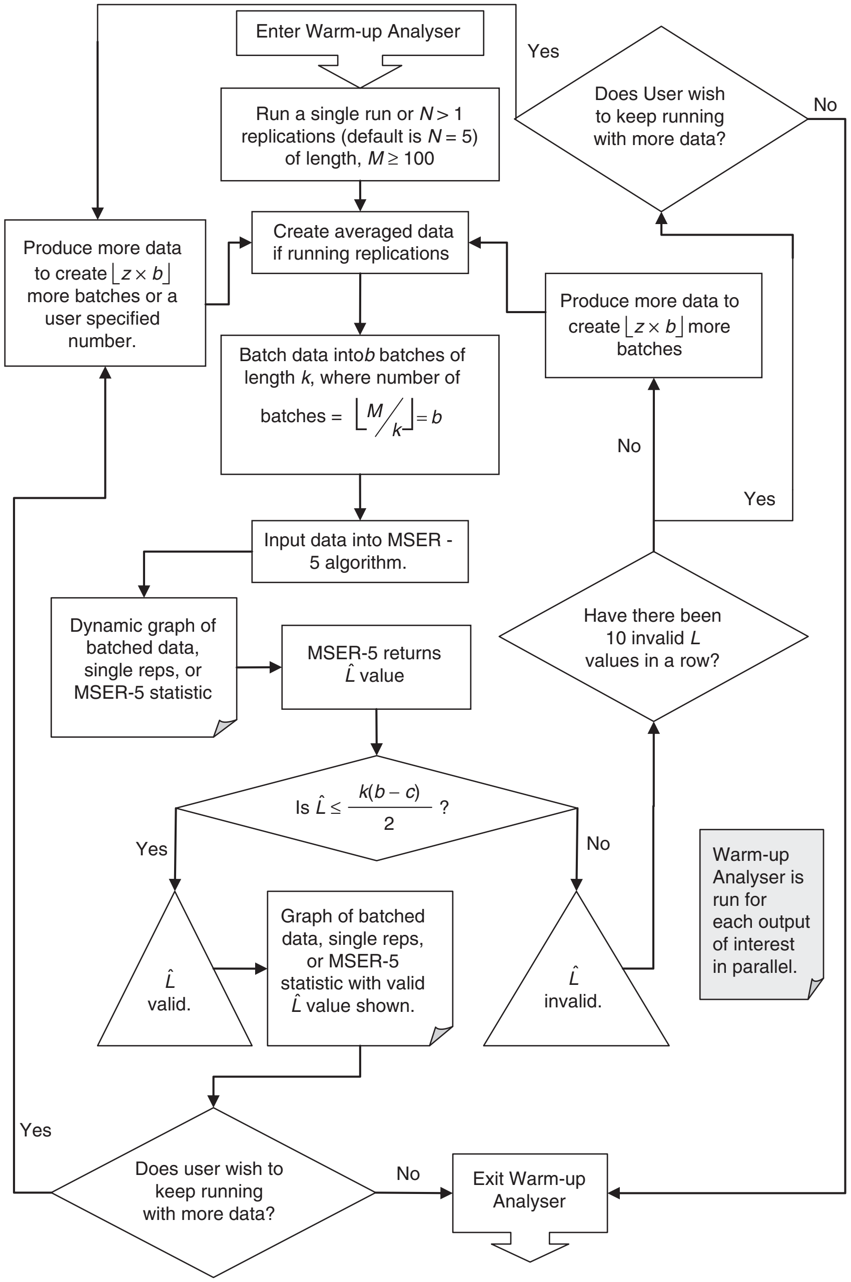
Enter Warm-up Analyser
Run a single run or $N > 1$ replications (default is $N = 5$) of length, $M \geq 100$
Create averaged data if running replications
Batch data into$b$ batches of length k, where number of batches = $\lfloor M/k \rfloor = b$
Input data into MSER - 5 algorithm.
Dynamic graph of batched data, single reps, or MSER-5 statistic
MSER-5 returns $\hat{L}$ value
Is $\hat{L} \leq \frac{k(b-c)}{2}$?
Yes
No
$\hat{L}$ valid.
Graph of batched data, single reps, or MSER-5 statistic with valid $\hat{L}$ value shown.
$\hat{L}$ invalid.
Have there been 10 invalid L values in a row?
No
Yes
Produce more data to create $\lfloor z \times b \rfloor$ more batches
Does User wish to keep running with more data?
Yes
No
Produce more data to create $\lfloor z \times b \rfloor$ more batches or a user specified number.
Warm-up Analyser is run for each output of interest in parallel.
Does user wish to keep running with more data?
Yes
No
Exit Warm-up Analyser

Figure 2 The warm-up analyser heuristic framework based on MSER-5.

the MSER heuristic). On a related note, it is advised that the run-length used for the warm-up analysis should not be much greater than the run-length the user expects to deploy in the subsequent runs of the simulation model, as the position of the truncation point determined using MSER-5 is dependent in part on the run-length.

MSER-5 provides a suggested truncation point ($\widehat{L}$) based on the data given to it; the calculation of the MSER-5 statistic and determination of $\widehat{L}$ are described in White (1997) and White *et al* (2000). The suggested truncation point, however, may not be valid. In particular, if $\widehat{L}$ falls in the second half of the data series, then there is a concern that there is insufficient apparent steady-state data to be certain that the model has reached a steady-state. Further to this, MSER-5 sometimes identifies a truncation point close to the end of the data series simply because by chance there is a low standard error in the final data values (the series is smooth). In order to address these issues, the analyser does not calculate the MSER-5 statistic for the last c values (default $c = 5$). As a consequence of these two issues, $\widehat{L}$ is only considered valid if $\widehat{L} \leqslant \lfloor k(b-c)/2 \rfloor$, that is, if it falls in the first half of the series for which the MSER-5 statistic has been calculated. If an invalid $\widehat{L}$ value is returned, further data are requested from the simulation model. Enough data to create a further $\lfloor z \times b \rfloor$ batches are produced (where the default for z is 10%).

While the algorithm is running, the user has the option of viewing a graph of the batched averaged data, the individual replications or single run, and the MSER-5 test statistic. When a valid $\widehat{L}$ value is returned by MSER-5, the proposed truncation point is shown on the graphs (see, for example, Figure 8). This provides an opportunity for face validation of the suggested truncation point. The user then has the opportunity to stop, accepting the suggested $\widehat{L}$, or to continue with more data until a truncation point that is deemed suitable is found.

If 10 invalid estimates of $\widehat{L}$ occur in a row the analyser pauses, informs the user and asks if the user wishes to continue. If the user does wish to continue the analyser asks for the number of extra batches to be input. As such, the user has the opportunity to jump further ahead by specifying how much more data should be generated. We believe that this facility is particularly important in two specific circumstances:

- If the user gave the method insufficient data in the first place for the model to have reached a steady-state.
- The output data are very highly autocorrelated, thus requiring more data to be able to identify the steady-state.

This facility acts as a fail-safe, preventing the analyser from wasting time when it has far too little data on which to base a decision. Once the user has specified the number of extra batches, the analyser continues as before, returning again to ask for more batches if 10 additional invalid estimates of $\widehat{L}$ are obtained.

6. The replications calculator (A2 in Figure 1)

The replications calculator aims to identify the number of replications ($\widehat{N}$) required to achieve a confidence interval of a specified precision ($d_{required}$). The precision (d_N) is defined as the half-width of the interval expressed as a percentage of the sample mean, that is:

$$d_N = \frac{100 t_{N-1,\alpha/2}\sqrt{s^2(N)/N}}{\bar{X}(N)} \tag{8}$$

where N is the current number of replications run. We also include the ability to define precision in absolute terms (ie the value of the half width), which is appropriate when $E[\bar{X}(N)] = 0$.

Since it is possible to select an incorrect (inaccurate) confidence interval that has achieved the desired precision by chance (ie through a series of similar X_n values), the calculator includes a look-ahead procedure. This procedure looks ahead by adding data from further replications to determine if the confidence interval remains within the desired precision. If the interval diverges from the desired precision, then the calculator will continue adding more replications until the confidence interval falls within the desired precision again. The length of the look-ahead is defined by the value $\lambda Limit$ such that the look-ahead $f(\lambda Limit)$ is calculated as follows:

$$f(\lambda Limit) = \begin{cases} \lambda Limit, & N \leqslant 100 \\ \left\lfloor N \times \frac{\lambda Limit}{100} \right\rfloor, & N > 100 \end{cases} \tag{9}$$

The replications calculator and the tests performed on it are described in more detail in Hoad *et al* (2009a).

The framework for the replications calculator is shown in Figure 3. As for the warm-up analyser, the user specifies the output variables of interest, which are likely to be substantially the same as those for which the warm-up analysis is performed. The replications calculator treats each variable independently, recommending individual numbers of replications for each variable. Again, for reasons of simplicity, the description below outlines the use of the calculator with only a single output.

Default values for $d_{required}$ and $\lambda Limit$ are 5% and 5, respectively. The significance level (α) is set to 5%, but may be changed by the user. The initial number of replications ($N_{initial}$) is set by default to 3. The calculator then runs the initial replications and calculates the sample mean, confidence limits and precision. If the precision is more than the required precision (ie $d_N > d_{required}$), an additional replication is performed and the results are recalculated. This loop continues until the precision is such that $d_N \leqslant d_{required}$. The look-ahead procedure is then invoked. This iteratively adds the results from an additional replication up to the number of additional replications as defined by $f(\lambda Limit)$. If on any of these replications the precision diverges to be outside the desired precision, the calculator reverts to adding additional replications until the confidence interval falls within the desired precision again. If the confidence interval remains within the desired precision throughout the look-ahead, the number of replications ($\widehat{N}$) is recommended as the value of N at the point where the confidence interval first met the desired precision criterion (ie at the start of the look-ahead).

One concern with the replications algorithm is the time that it might take to reach the desired precision. This could be a problem when the model runs slowly, or when a very fine level of precision is required. In order to address these issues, we recommend a fail-safe mechanism in which the procedure continuously monitors the number of replications that might be required and reports this to the user through a graph. The required number of replications can be estimated after each replication as follows (Banks *et al*, 2005):

$$\hat{N}^* = \left[\frac{100 t_{N-1,\alpha/2^{s(N)}}}{d_{required}\bar{X}(N)}\right]^2 \tag{10}$$

Our tests showed that this value is only stable for large values of N. As a result, it should only be used as an

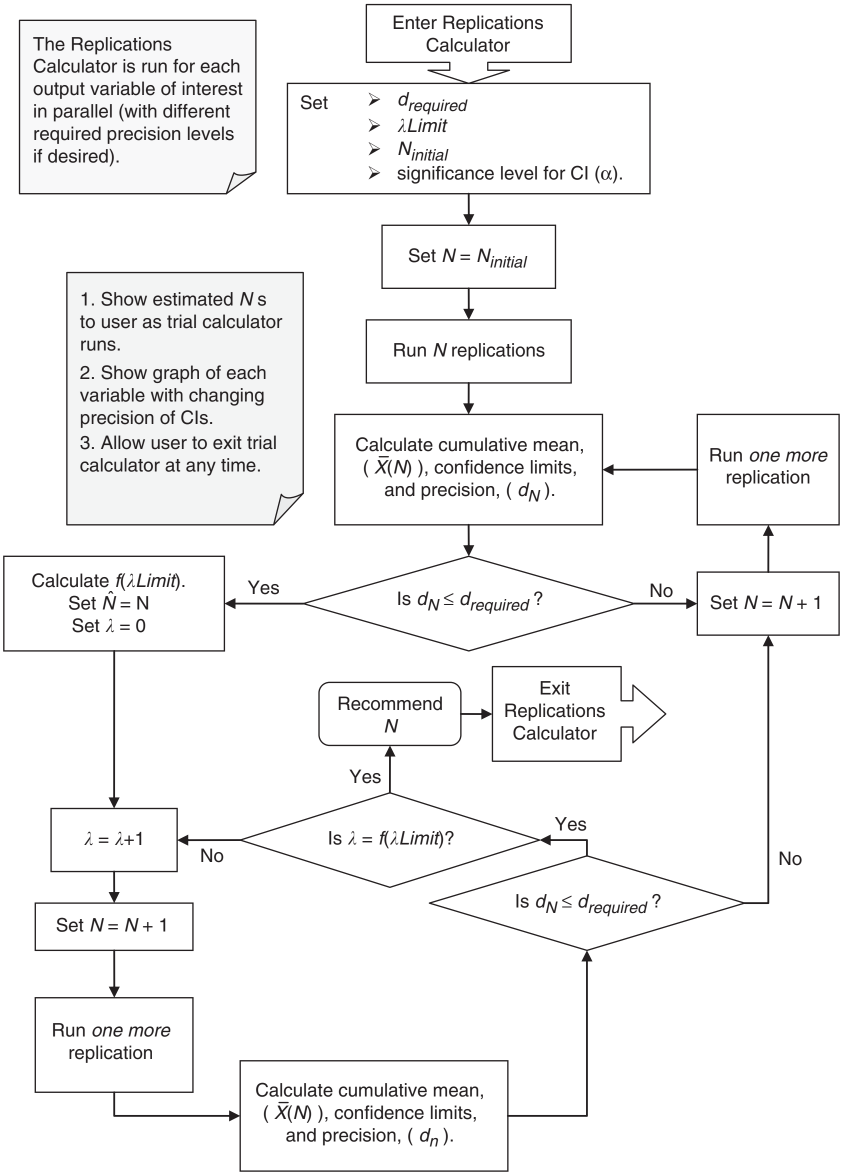
The Replications Calculator is run for each output variable of interest in parallel (with different required precision levels if desired).
Enter Replications Calculator
Set
$d_{required}$
$\lambda Limit$
$N_{initial}$
significance level for CI (α).
Set $N = N_{initial}$
1. Show estimated Ns to user as trial calculator runs.
2. Show graph of each variable with changing precision of CIs.
3. Allow user to exit trial calculator at any time.
Run N replications
Calculate cumulative mean, ($\bar{X}(N)$), confidence limits, and precision, (d_N).
Run one more replication
Is $d_N \le d_{required}$?
Yes
No
Calculate $f(\lambda Limit)$. Set $\hat{N} = N$ Set $\lambda = 0$
Set $N = N + 1$
Recommend N
Exit Replications Calculator
Yes
Is $\lambda = f(\lambda Limit)$?
Yes
No
$\lambda = \lambda+1$
No
Is $d_N \le d_{required}$?
Set $N = N + 1$
Run one more replication
Calculate cumulative mean, ($\bar{X}(N)$), confidence limits, and precision, (d_n).

Figure 3 The replications calculator heuristic framework.

indicator from which a user can decide whether to continue trying to run the model to the desired precision.

7. The single run analyser (A3 and A4 in Figure 1)

If the user opts to perform a single (long) run of the simulation, there are three possible options for running the model:

1. *Fixed Run-Length*: The user has a specific run-length (M) in mind (eg 1 month) and wants a mean value with a valid confidence interval at the end of this set time. In this case, the single run analyser should calculate the sample mean and confidence interval from the given data and report the precision achieved. If there are insufficient data to produce a valid confidence interval, the algorithm should advise the user accordingly.
2. *Confidence Interval with a Specific Precision*: The user desires a mean estimate with a confidence interval of a specific precision. The output analyser should run the model until there are enough data to achieve this ($\widehat{M}$). However, the user must be given the ability to abort the procedure if the analyser is taking too long (ie too much data are required for the specified precision). The algorithm should then form a valid confidence interval, if possible, using the data created thus far, and report the precision achieved.
3. *Valid Confidence Interval*: The user neither requires a specific precision nor does the user have a set run-length

in mind. The output analyser should run the model until enough data are collected to achieve a valid confidence interval ($\widehat{M}$). Again, the user must be given the option to abort the procedure if the method is taking too long.

The first option is addressed by the batch means calculator (A4 in Figure 1) and the second and third options by the run-length calculator (A3 in Figure 1). When the user chooses to abort the run-length calculator, this returns the analyser to running option 1 (the batch means calculator).

As stated previously, the batch means approach has been adopted for AutoSimOA. A search of the literature on batch means methods revealed a core group of researchers working on this approach. As a result, the majority of the methods are extensions or adaptations of previous methods. Figure 4 shows the 'family trees' of the methods, depicting which methods were developments from previous work.

Batch means methods generally fall into two categories: sequential or fixed sample size. The first are methods that sequentially request more data until some stopping criterion (eg a precision requirement) is fulfilled; these are suitable for application to the second and third options above. The second category acts upon a fixed amount of data, producing results, if possible, with only the data given; these are suitable for application to the first option above.

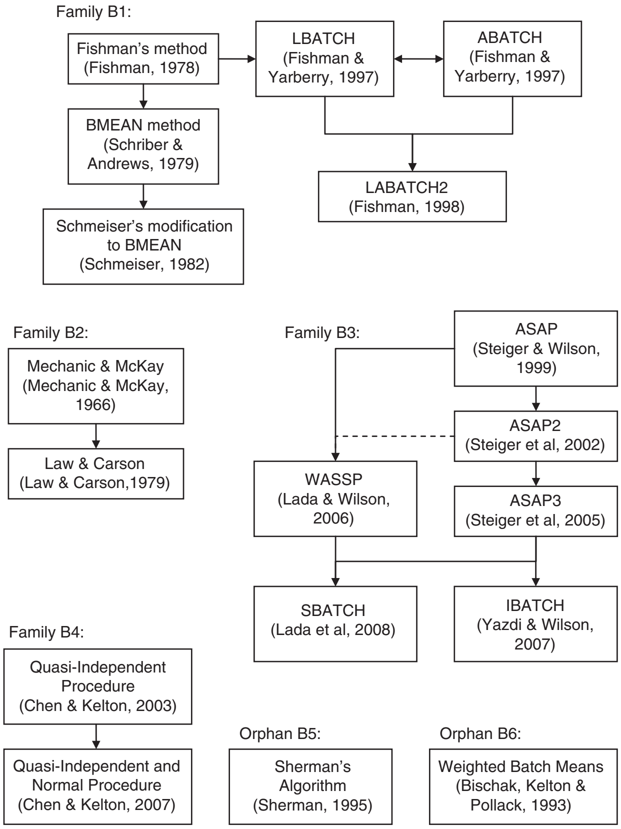
Family B1:
Fishman's method (Fishman, 1978)
LBATCH (Fishman & Yarberry, 1997)
ABATCH (Fishman & Yarberry, 1997)
BMEAN method (Schriber & Andrews, 1979)
LABATCH2 (Fishman, 1998)
Schmeiser's modification to BMEAN (Schmeiser, 1982)
Family B2:
Mechanic & McKay (Mechanic & McKay, 1966)
Law & Carson (Law & Carson,1979)
Family B3:
ASAP (Steiger & Wilson, 1999)
ASAP2 (Steiger et al, 2002)
WASSP (Lada & Wilson, 2006)
ASAP3 (Steiger et al, 2005)
SBATCH (Lada et al, 2008)
IBATCH (Yazdi & Wilson, 2007)
Family B4:
Quasi-Independent Procedure (Chen & Kelton, 2003)
Quasi-Independent and Normal Procedure (Chen & Kelton, 2007)
Orphan B5:
Sherman's Algorithm (Sherman, 1995)
Orphan B6:
Weighted Batch Means (Bischak, Kelton & Pollack, 1993)

Figure 4 'Family trees' of batch means methods in the simulation literature.

Bearing this in mind we selected a method from each category, LABATCH2 (Fishman, 1998) and ASAP3 (Steiger *et al*, 2005), for implementation within AutoSimOA. Both of these methods are fairly recent and received good recommendations and test results in the literature (Fishman, 1998; Steiger *et al*, 2005; Alexopoulos, 2006, Lada *et al*, 2008). The LABATCH2 algorithm works on a fixed quantity of data, providing a confidence interval without set precision. It was, therefore, deemed suitable for option 1. Meanwhile, ASAP3 is a sequential procedure that attempts to create a valid confidence interval around the mean estimate to a set precision (described in either absolute or relative terms). It can also run with no precision requirement. It is, therefore, suitable for options 2 and 3. Both procedures are easily automated and therefore suitable for AutoSimOA.

A brief overview of LABATCH2 and ASAP3 is now provided. For further details see Fishman (1998) and Steiger *et al* (2005), respectively.

7.1. The LABATCH2 procedure (Option 1: A4 in Figure 1)

LABATCH2 takes in a set amount of data (specified by M) and performs a series of 'interim reviews' on increasing amounts of the data provided. Each interim review consists of batching the data, using either the square root (SQRT) or fixed number of batches (FNB) rules (Alexopoulos *et al*, 1997), carrying out the Von Neumann test for independence (Von Neumann, 1941), and producing an estimate of the mean and variance (of the batched data), and a confidence interval for the mean. These interim results (both displayed in graphic and tabular form) can then be surveyed by the user in order to conclude whether the variance has stabilised and the final confidence interval is valid. LABATCH2 is essentially made up of two separate methods, ABatch and LBatch (Fishman and Yarberry, 1997). The user is required to choose which one of these should be used in the procedure. The only difference between these two methods is the decision rule used to switch between the SQRT and FNB batching rules.

7.2. The ASAP3 procedure (Options 2 and 3: A3 in Figure 1)

ASAP3 is a sequential procedure that progressively increases the batch size (and correspondingly the run-length, $\widehat{M}$) until the batch means pass the Shapiro–Wilk multivariate normality test (Shapiro and Wilk, 1965). The batch size is then further increased until a first-order autoregressive time series model (AR(1)), with a parameter not significantly greater than 0.8, can be fitted to the batched data. The terms of an inverted Cornish-Fisher expansion (Stuart and Ord, 1994) are then computed for the classical batch means t-ratio based on the AR(1) parameter estimators. ASAP3 then produces a correlation-adjusted confidence interval based on this expansion (Steiger *et al*, 2005). Although ASAP3 attempts to deal with any initial bias in the data by discarding the first two batches of data, it was found not to cope well with substantial bias in that it required extremely (and prohibitively) large amounts of data. This is not such a problem for AutoSimOA, since much of the initial bias should have been removed by the warm-up analyser.

8. Examples of the implementation of AutoSimOA

Having described AutoSimOA, we now demonstrate the results of using the complete analyser on two case studies. We generated output data from two discrete-event simulation models. The first is a steady-state data set with an initial transient (generated from a non-terminating simulation) and the second is a transient data set (generated from a terminating simulation) in which customer arrivals vary over a day (the run-length is 1 day). Using these two data sets we are able to explore the different paths through AutoSimOA.

The first model ('user support model') simulates calls received, processed and actioned at an IT support help desk (Robinson, 2001). The output of interest is the average time the calls spend in the system (Figure 5). This is a steady-state output with a substantial initial bias. The true steady-state mean is believed to be around 2269 min. This was estimated using a long run with 54 000 data points. The second model is of a cinema call centre (Robinson, 2004). The output of interest is the average waiting time of the calls in the centre (Figure 6). This is a transient output as the model has varying call arrival rates throughout the working day and the model terminates after a 1-day run. There is no initialisation bias since the simulation starts from a realistic empty state. The true mean average waiting time is estimated as 1.4399 min based on running the model for many replications to obtain results for 113 600 entities.

Table 1 sets out the details of each data set with the paths that each set could take through AutoSimOA. Figures 7 (a)–(d) illustrate these different paths.

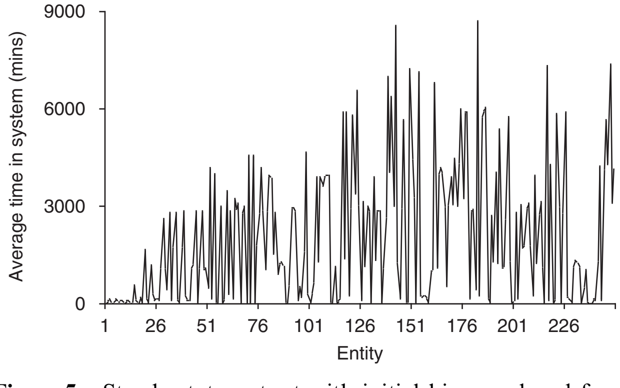

Figure 5 Steady-state output with initial bias produced from the 'user support model': output = average time calls spend in the system (in min).

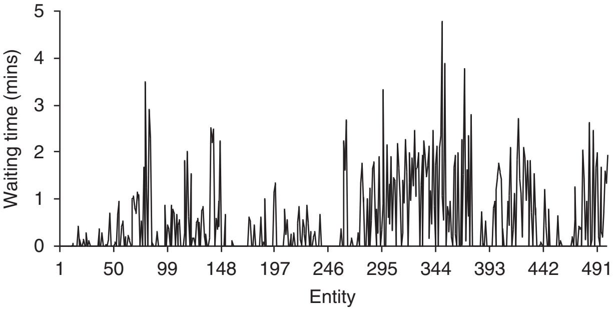

Figure 6 Transient output produced from the 'cinema call centre' model: output = waiting time per entity (in min).

8.1. Results: steady-state output data from the user support model

Path A: warm-up analyser and replications calculator. Five replications of length 500 (user specified) are run and averaged across the replications to give values for $\bar{y}$. These are input into the warm-up analyser that recommends a warm-up period of 70 data points. From the MSER-5 graph (Figure 8) it appears that the data have stabilised and a truncation point of 70 is reasonable. The recommended warm-up period is therefore accepted and the current data (5 replications) truncated accordingly. This leaves a truncated run-length of 430.

AutoSimOA then moves on to determine the number of replications that are required. On entering the replications calculator the user is asked if the existing run length of 430 is sufficient. This is not deemed to be sufficient and an amended run-length of 1000 is requested. More data are then created from the simulation model to bring each of the 5 truncated replications to a length of 1000.

The input parameters for the replications calculator are kept at their default levels. The calculator recommends that 10 replications are required to achieve the desired precision, giving a mean estimate of 2211.3 and 95% confidence interval of (2104.5, 2318.1). This interval covers the estimated true value of 2269.

As the calculator runs it generates two graphs. The first shows the mean, confidence intervals and precision as more replications are run (Figure 9). The second graph shows how the estimate of the number of replications that might be required ($\widehat{N}^*$) changes as more replications are run (Figure 10). It is this graph that provides a fail-safe for the user. If $\widehat{N}^*$ appears to be so large that the running time for the calculator is going to be excessive, the user has the option to interrupt the calculator and change the parameters; most likely the required precision. Note how poor the estimates of $\widehat{N}^*$ are with only a few replications.

Path B: warm-up analyser and run-length calculator. Data from a single run of length 350 is entered into the warm-up analyser, which then recommends a warm-up period of 25 data points. The graph of the data and test statistic

Table 1 Data sets used to demonstrate AutoSimOA

Data set	*Initial bias*	*Steady-state/transient*	*Path*
User support model output: time in system (min)	Yes	Steady state	A, B, C
Cinema call centre model output: waiting time (min)	No	Transient	D

A: *Replications or one run?*—'Replications'—*Warm-up?*—'Yes'—Warm-up Analyser—Replications Calculator—Exit Analyser
B: *Replications or one run?*—'One run'—*Warm-up?*—'Yes'—Warm-up Analyser—*Set run-length?*—'No'—Run-length Calculator—Exit Analyser
C: *Replications or one run?*—'One run'—*Warm-up?*—'Yes'—Warm-up Analyser—*Set run-length?*—'Yes'—Batch Means Calculator—Exit Analyser
D: *Replications or one run?*—'Replications'—*Warm-up?*—'No'—Replications Calculator—Exit Analyser.

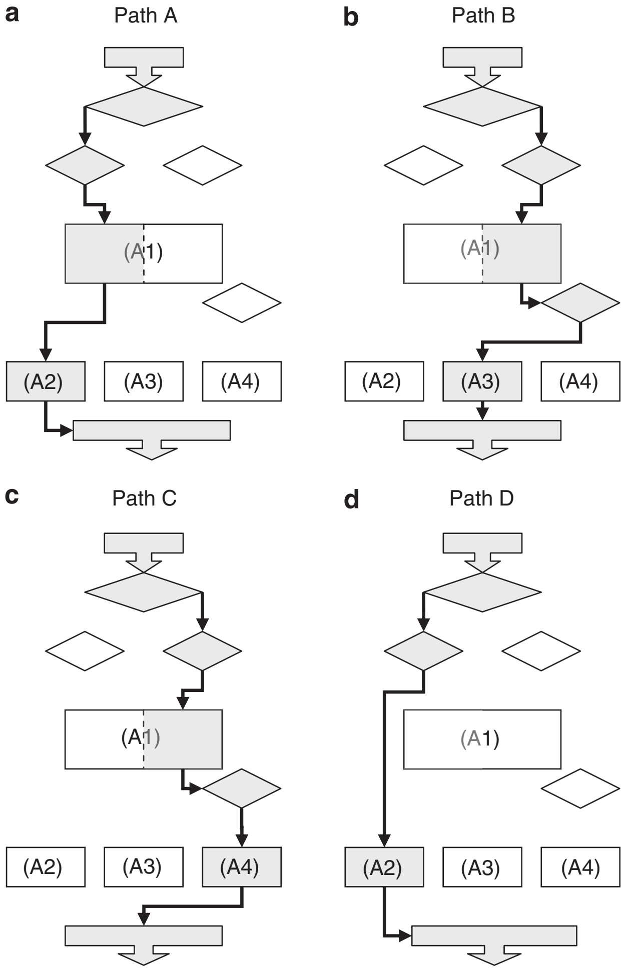

Figure 7 Paths through AutoSimOA, for example models shown on an outline of Figure 1.

(Figure 11(a)) shows a relatively large peak in the batched data values at the end of the current data set. It is therefore unclear whether a run length of 350 is sufficient since it appears the data may not have reached steady-state. A further 250 data points (50 batches) are added and again the analyser recommends a warm-up of 25 data points. After examining the new graph (Figure 11(b)) this truncation point is accepted.

The run-length calculator is then invoked to produce a confidence interval with a required absolute half width of ±60 min. ASAP3 requires 10 368 data points to achieve this, giving a mean estimate of 2297.7 and a half width of 58.1, hence the confidence interval is (2239.6, 2355.8). This interval does cover the estimated true mean.

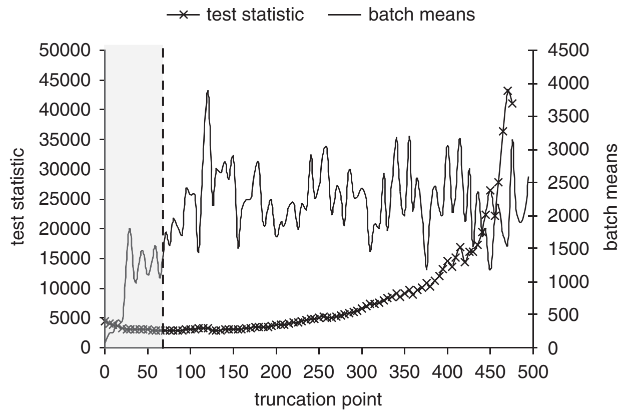

Figure 8 Path A: MSER-5 warm-up analysis for multiple replications with the user support model (recommended warm-up denoted by vertical dashed line).

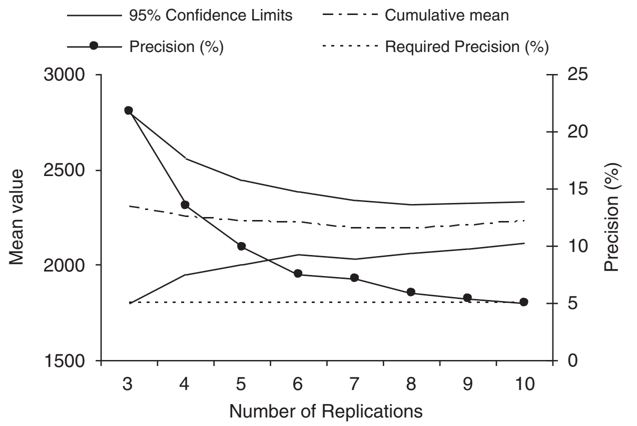

Figure 9 Path A: replications calculator applied to the user support model with the Graph showing the mean, 95% confidence intervals and precision.

Path C: warm-up analyser and batch means calculator. The warm-up analysis is the same as for path B and a warm-up period of 25 data points is used. The run-length is set to 6500 data points. The LABATCH2 method is implemented using the LBatch procedure, and requiring 95% confidence intervals. Table 2 shows the results obtained from the sequence of

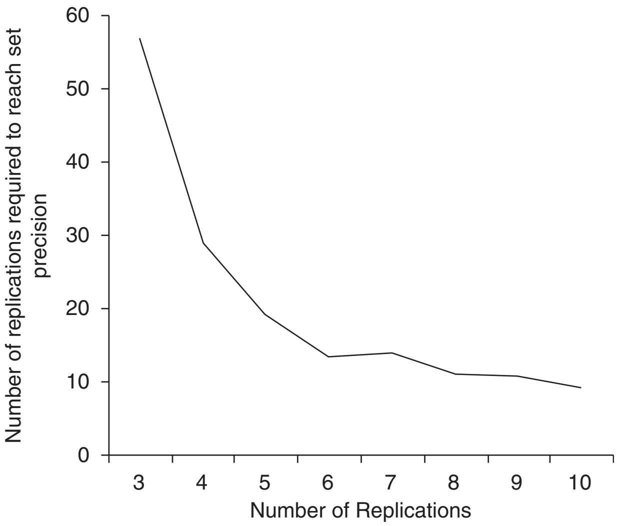

Figure 10 Path A: replications calculator fail-safe (applied to the user support model) with the graph showing the estimated number of replications ($\widehat{N^*}$).

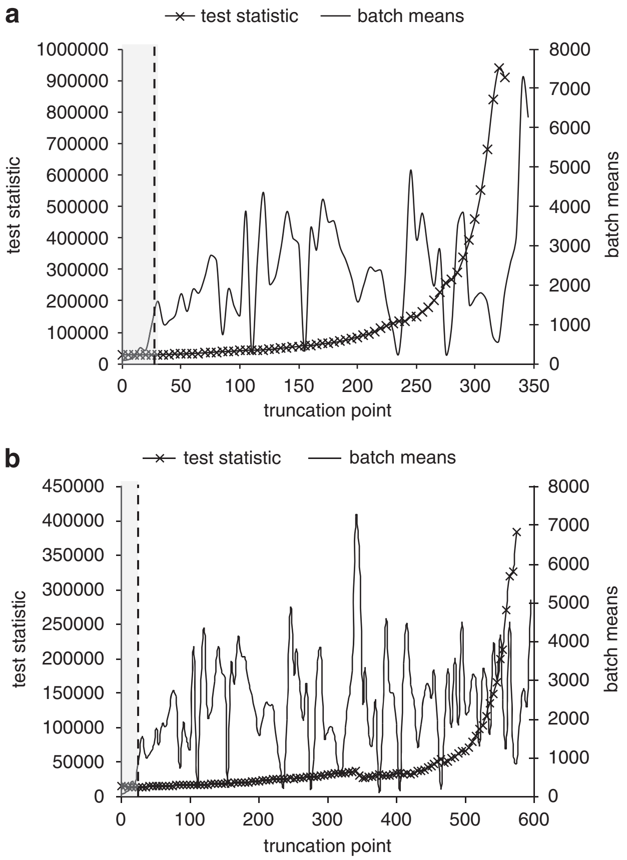

Figure 11 (a) Path B: MSER-5 warm-up analysis for a single replication (run length of 350) with the user support model (recommended warm-up denoted by vertical dashed line); (b) Path B: MSER-5 warm-up analysis for a single replication (run length of 600) with the user support model (recommended warm-up denoted by vertical dashed line).

reviews performed by LABATCH2 as it increases the amount of data in the calculation of the confidence interval.

Figures 12 and 13 display the standard deviation and the mean and confidence intervals (respectively) at each review. On the final review, 94.5% of the data are used (the maximum amount possible with the specific starting batch size (1) and number of batches (3) selected). Looking at the tabular results it is noted that the batched data passes the independence test for 9 out of the 12 reviews with no particular pattern for the failures (cells shaded in grey). Looking at the graphs it appears that the standard deviation of the batched data begins to stabilise and the confidence intervals narrow after the eighth review. We therefore choose to accept the final mean estimate and confidence interval displayed at the top of the results tableau, noting that this confidence interval has a precision of 3.2%. The confidence interval (2251.5, 2401.7) covers the estimated true mean, as for paths A and B.

8.2. Results: transient output data from the cinema call centre model

Path D: replications calculator. When applied to the cinema call centre model the replications calculator recommends 5 replications to achieve the specified precision of 5%. The mean estimate is 1.4441 with a confidence interval of (1.3824, 1.5058). This interval includes the estimated true mean of the process (1.4399). Figure 14 shows the mean, confidence limits and precision reported by the replications calculator.

9. Implementation of AutoSimOA

AutoSimOA is intended to be an automated output analysis system that can be implemented with commercial simulation software. Indeed, some elements of the framework have already been implemented in the SIMUL8 software. Our experience in developing, testing and implementing AutoSimOA has raised a number of practical issues. These are discussed here.

9.1. Output data type

One issue with applying AutoSimOA to real models is which output data should, and should not, be analysed. There are two specific data types that present a problem in this respect: cumulative values (eg utilisations and average time in a queue) and extreme values (eg maxima and minima). Both of these present a particular problem for time-series-based analyses, that is, warm-up and single run analysis. For cumulative results the variance of the data reduces as the quantity of data is increased. For extreme values the same is likely to be the case with changes to maxima and minima becoming less frequent as the simulation runs. Indeed, at an

Table 2 Path C: LABATCH2 results for the user support model

Final results table: mean estimate

Series	*Mean Estimate**	*Standard error*[†]	*95% confidence interval*		*Standardised half width*	*Standardised*
			Lower	*Upper*		
1	2326.59	38.3162	2251.481	2401.705	0.032284	

Total number of data points supplied: 6500

Interim review table

LBatch data analysis for output series 1

Review	*Data total*	*Batch number*	*Batch size*	*Mean estimate*	*95% confidence interval*		*Standard deviation estimate*	*Independence test p-value*[#]
					Lower	*Upper*		
1	3	3	1	440.617	−950.473	1831.707	559.989	0.299200
2	8	4	2	1040.324	−681.945	2762.592	1082.355	0.176052
3	12	6	2	1080.458	−289.356	2450.272	1305.287	0.260779
4	24	8	3	1179.750	447.001	1912.500	876.473	0.338124
5	48	12	4	1508.040	865.712	2150.368	1010.952	0.017736
6	96	16	6	1715.260	681.946	2748.574	1939.176	0.264867
7	192	24	8	2268.810	1181.611	3356.008	2574.695	0.016617
8	384	32	12	2305.105	1000.770	3609.440	3617.742	0.131456
9	768	48	16	2445.910	1385.700	3506.119	3651.240	0.144988
10	1536	64	24	2448.422	1634.686	3262.158	3257.647	0.011680
11	3072	96	32	2431.859	1781.332	3082.386	3210.595	0.221122
12[§]	6144	128	48	2334.874	1794.568	2875.181	3089.150	0.437934
If original data are independent:								
	6500	6500	1	2326.593	2270.261	2382.925	2316.76449	1.57E–28

*The estimate of the mean is based on all 6500 observations.

[†] The estimate of the variance is based on the first 94.52% of the 6500 observations.

[#] Significance level for independence testing = 0.1

[§] Review 12 used the first 94.52% of the 6500 observations.

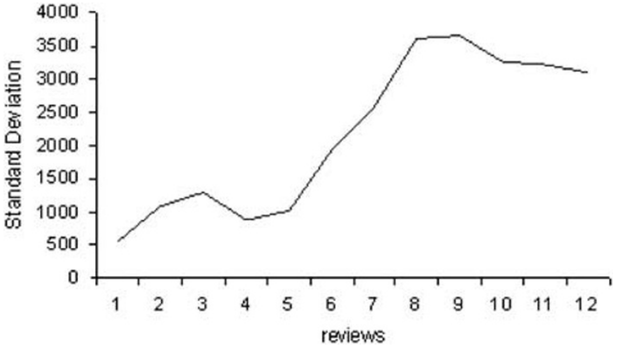

Figure 12 Path C: LABATCH2 standard deviation results for the user support model.

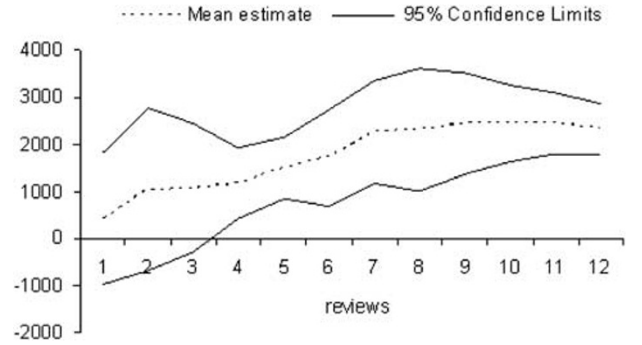

Figure 13 Path C: LABATCH2 mean estimate and confidence interval results for the user support model.

extreme, the minimum value of entities in a queue does not change from the initial state of empty. To address these issues we recommend that cumulative data are disaggregated, for instance, percentage utilisation is disaggregated into a time-series of hourly utilisation. Furthermore, AutoSimOA should not be applied to extreme values. For time-series data there is also an issue of whether the data should be recorded and analysed based on time (eg hourly) or on entities (eg time in queue for individual entities as they leave). It seems that, in general, simulation software only directly record output data by time (eg a time-series of throughput) and that to record data by entity requires some additional coding

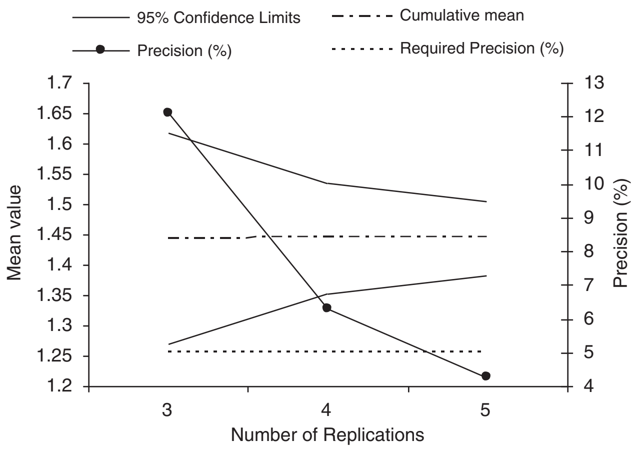

Figure 14 Path D: replications calculator applied to the cinema call centre model with the graph showing the mean, 95% confidence intervals and precision.

with the data being written to an external file. It also seems that the software define warm-up periods and run-lengths in terms of time and not the number of entities processed. Interestingly, many research papers record data by entity and there are clearly cases where it is preferable to do so. It would be useful if simulation software gave a facility to set up runs based on either time (eg 10 080 min, 1 week) or entities (eg once 1000 have left the system), and also gave a facility for recording the output data in either way.

Given that the software provides time-based data and requires the set-up to be in terms of time, then disaggregation should also be based on time. This is not a trivial task as the analyser needs to record the cumulative statistics and the number of entities involved in the aggregate values for each time period in order to disaggregate the data properly.

9.2. Multiple outputs

It was noted previously that we would normally be interested in more than one output. Indeed, it is possible that a user may have quite a number of key output statistics of interest. In practical terms this is relatively straightforward to address. We would recommend that AutoSimOA analyses all the outputs of interest (the full matrix $\mathbf{Y}^p$) and that the warm-up period, run-length and/or number of replications is selected based on the output that requires longest to stabilise.

9.3. Multiple scenarios

AutoSimOA performs an analysis on an individual simulation scenario, but frequently users need to analyse multiple scenarios. We believe the ideas underlying AutoSimOA can be extended to the case of comparing multiple scenarios. For instance, when comparing two scenarios a confidence interval on the differences between the mean of an output statistic from the two scenarios can be used for comparison purposes. The replications calculator can be used to obtain a confidence interval on the differences of a specified precision.

Where more than two scenarios are involved then multiple confidence statements are required. As is standard practice in statistical inference, the significance level (α) associated with multiple confidence intervals should be adjusted in order to provide an overall confidence level (see Law (2007) on the Bonferroni inequality). This becomes problematic if a very large number of output statistics are of interest. It is then necessary to resort to more sophisticated means, for example ranking and selection (Kim and Nelson, 2007).

In terms of assuring the accuracy of the results obtained from each scenario, then AutoSimOA could be run for every scenario. The problem with this is that the sample sizes obtained from each scenario might (necessarily) be quite different. For reasons of fair comparison it might be preferable to use AutoSimOA on a base case and then reuse the warm-up period, run-length and/or number of replications for all other cases. In this instance, we would recommend re-running AutoSimOA from time-to-time, and particularly for critical scenarios, in order to affirm that an appropriate run set-up is still being used. The results would be accurate in themselves but might not be sufficiently accurate to compare with each other.

If the run-length calculator is used on the base case (using ASAP3) then this would present an issue with reusing that run-length on further scenarios. Since ASAP3 relies on a correlation-adjusted confidence interval in its calculations, the resulting confidence interval is not just dependent on the batch size and number of batches. In other words, the same batch size and batch number cannot be applied to a different set of data and a valid confidence interval necessarily be produced. A pragmatic solution to this might be to use the run-length suggested by ASAP3 for the base case on all future runs, but to apply the LABATCH2 procedure for generating the confidence intervals. It should be noted that this would not assure the precision of the confidence intervals, but it would provide a more consistent basis for generating the results.

9.4. Issues with automation

AutoSimOA is a sequential procedure for making decisions about warm-up, run-length and number of replications. As a result, at each iteration the analyser requires more data to be generated from the simulation model. To do this efficiently the simulation should be able to run on from its present termination point. In general this is not an issue for most simulation software when running a single replication. This requires some more effort, however, when multiple replications are being performed and further data are required, as is the case when using averaged data in the warm-up analyser.

AutoSimOA cannot claim to be fully automated. Indeed, this is in part deliberate since we believe that output analysis requires some intervention from the user of the model who is knowledgeable about the real-world system being modelled. There are two key areas where users are involved in the analysis with AutoSimOA.

Deciding what to do. On the basis of a knowledge of the model, the user must determine whether a warm-up analysis is needed and whether multiple replications or a long run is required. The user must also decide the length of run for the multiple replications case. These decisions depend on the nature of the model and the output, for instance, whether the model is terminating or non-terminating, or whether the output data are transient or steady-state. There may be ways of automating these decisions by inspecting the details of the model, but this would require further research into the characteristics of models.

Determining if the recommendations are reasonable. For each element of AutoSimOA the user is presented with graphical output showing the nature of the recommendations being made. For the warm-up analyser we recommend that the user can view graphs of the MSER-5 test statistic (Figure 8), the batched data (Figure 8) and the individual time-series if multiple replications are being used. For the replications calculator the user should be able to see a graph of the mean, confidence intervals and precision with increasing numbers of replications (Figure 9). We also recommend a graph showing the expected number of replications ($\widehat{N}^*$) (Figure 10) as a fail-safe mechanism to prevent excessive running time. Finally, for the single run analyser we would suggest graphs of the standard deviation (Figure 12) and mean and confidence intervals (Figure 13) are presented to the user for the LABATCH2 procedure. These graphs allow the user to gain some understanding of the reason for the recommendation and thereby some confidence in the decision. The user is also given the option to override the recommendations of AutoSimOA.

9.5. Limitations of AutoSimOA

There are some output analysis requirements that are not covered by AutoSimOA in its current form. First, AutoSimOA is not directly able to handle cyclic data. The user is required to batch the data to remove cycles prior to analysis by AutoSimOA. Second, if transient output data are subject to initialisation bias, the procedures used by AutoSimOA will not be able to detect the difference between the initial transient and further transient data. We do not know of such a procedure, and it is likely that for the foreseeable future users will need to identify such initial transients based on their knowledge of the model.

A third limitation of AutoSimOA is that it only performs an analysis on the mean and variance of the output statistics of interest. There are occasions on which other statistics need to be measured, for instance, the mode, median and quantiles. It would be useful to extend AutoSimOA to provide decisions for these measures.

Finally, AutoSimOA provides no facilities for scenario analysis. This might include, for instance, comparison of scenarios (eg ranking and selection (Kim and Nelson, 2007)), and simulation optimisation (Fu, 2002). The addition of such facilities is an area for further research.

10. Conclusion

The automated analyser, AutoSimOA, provides recommendations on the warm-up period, number of replications and/or run-length for a discrete-event simulation. It is for use with terminating and non-terminating (steady-state) simulations. The aim of the analyser is to provide accurate and precise measures for the output statistics that are of interest from a simulation model. AutoSimOA has been designed for implementation in commercial simulation software with a view to improving the use of, and results obtained from, commercial simulation models. In its current form, AutoSimOA provides facilities that can help improve the use of simulation. We believe that with additional research and development the analyser can provide further capabilities to aid simulation output analysis.

Acknowledgements—This work was part of the Automating Simulation Output Analysis (AutoSimOA) project (http://www.wbs.ac.uk/go/autosimoa, accessed March 2009) that was funded by the UK Engineering and Physical Sciences Research Council (EP/D033640/1). The work was carried out in collaboration with SIMUL8 Corporation, who also provided sponsorship for the project. We would like to thank Luna Fadayel for her help with the work on batch means methods.

References

Alexopoulos C (2006). A comprehensive review of methods for simulation output analysis. In: LF Perrone *et al* (eds). *Proceedings of the 2006 Winter Simulation Conference*. IEEE: Piscataway, NJ, pp 168–178.

Alexopoulos C, Fishman GS and Seila AF (1997). Computational experience with the batch means method? In: S Andradottir, KJ Healy, DH Withers and BL Nelson (eds). *Proceedings of the 1997 Winter Simulation Conference*. IEEE: Piscataway, NJ, pp 194–201.

Banks J, Carson II JS, Nelson BL and Nicol DM (2005). *Discrete-event System Simulation*, 4th edn. Prentice Hall: Upper Saddle River, NJ.

Bischak DP, Kelton WD and Pollock SM (1993). Weighted batch means for confidence intervals in steady-state simulations. *Mngt Sci* **39**(8): 1002–1019.

Chen EJ and Kelton WD (2003). Determining simulation run length with the runs test. *Simulat Model Pract Theor* **11**: 237–250.

Chen EJ and Kelton WD (2007). A procedure for generating batch-means confidence intervals for simulation: Checking independence and normality. *Simulation* **83**: 683–694.

Cochran WG (1934). The distribution of quadratic forms in a normal system, with applications to the analysis of covariance. *P Camb Philol Soc* **30**: 178–191.

Conway RW (1963). Some technical problems in digital simulation. *Mngt Sci* **10**(1): 47–61.

Crane MA and Iglehart DL (1974a). Simulating stable stochastic systems, I: General multiserver queues. *J Assoc Comput Mach* **21**(1): 103–113.

Crane MA and Iglehart DL (1974b). Simulating stable stochastic systems, II: Markov chains. *J Assoc Comput Mach* **21**(1): 114–123.

Crane MA and Iglehart DL (1975). Simulating stable stochastic systems, III: Regenerative processes and discrete-event simulations. *Opns Res* **23**: 33–45.

Crane MA and Lemoine AJ (1977). *An Introduction to the Regenerative Method or Simulation Analysis*. Springer-Verlag: New York.

Fishman GS (1971). Estimating sample size in computing simulation experiments. *Mngt Sci* **18**(1): 21–38.

Fishman GS (1973). *Concepts and Methods in Discrete Event Simulation*. Wiley: New York.

Fishman GS (1977). Achieving specific accuracy in simulation output analysis. *Commun ACM* **20**: 310–315.

Fishman GS (1978). Grouping observation in digital simulation. *Mngt Sci* **24**(5): 510–521.

Fishman GS (1998). LABATCH2: Software for statistical analysis of simulation sample path data. In: DJ Medeiros, EF Watson, M Manivannan and J Carson (eds). *Proceedings of the 1998 Winter Simulation Conference*. IEEE: Piscataway, NJ, pp 131–139.

Fishman GS and Yarberry LS (1997). An implementation of the batch means method. *INFORMS J Comput* **9**: 296–310.

Fu MC (2002). Optimization for simulation: Theory vs. practice. *INFORMS J Comput* **14**(3): 192–215.

Heidelberger P and Welch PD (1981). A spectral method for confidence interval generation and run length control in simulation. *Commun ACM* **25**: 233–245.

Hoad K, Robinson S and Davies R (2009a). Automated selection of the number of replications for a discrete event simulation. *J Opl Res Soc*, doi:10.1057/jors.2009.121.

Hoad K, Robinson S and Davies R (2009b). Automating warm-up length estimation. *J Opl Res Soc*, doi:10.1057/jors.2009.121.

Hollocks BW (2001). Discrete-event simulation: An inquiry into user practice. *Simulat Pract Theory* **8**: 451–471.

Kim S-H and Nelson BL (2007). Recent advances in ranking and selection. In: SG Henderson *et al* (eds). *Proceedings of the 2007 Winter Simulation Conference*. IEEE: Piscataway, NJ, pp 162–172.

Lada EK, Steiger NM and Wilson JR (2008). SBatch: A spaced batch means procedure for steady-state simulation analysis. *J Simulation* **2**(3): 170–185.

Lada EK and Wilson JR (2006). A wavelet-based spectral procedure for steady-state simulation analysis. *Eur J Opl Res* **174**: 1769–1801.

Lavenberg SS and Sauer CH (1977). Sequential stopping rules for the regenerative method of a simulation. *IBM J Res Dev* **21**: 545–558.

Law AM (2007). *Simulation Modeling and Analysis*, 4th edn. McGraw Hill: New York.

Law AM and Carson JS (1979). A sequential procedure for determining the length of a steady-state simulation. *Opns Res* **27**(5): 1011–1025.

Law AM and Kelton WD (1982). Confidence intervals for steady-state simulations, II: A survey of sequential procedures. *Mngt Sci* **28**(5): 550–562.

Law AM and Kelton WD (1984). Confidence intervals for steady-state simulations, I: A survey of fixed sample size procedures. *Opns Res* **32**(6): 1221–1239.

Law AM and McComas MG (2002). Simulation-based optimization. In: E Yücesan, C-H Chen, SL Snowden and JM Charnes (eds). *Proceedings of the 2002 Winter Simulation Conference*. IEEE: Piscataway, NJ, pp 41–44.

Mason SJ, Hill RR, Mönch L, Rose O, Jefferson T and Fowler JW (2008). *Proceedings of the 2008 Winter Simulation Conference*. IEEE: Piscataway, NJ.

Mechanic H and McKay W (1966). *Confidence intervals for averages of dependent data in simulations II*. Technical Report ASDD 17-202, IBM Corporation, Yorktown Heights, New York.

Robinson S (2001). Soft with a hard centre: Discrete-event simulation in facilitation. *J Opl Res Soc* **52**(8): 905–915.

Robinson S (2004). *Simulation: The Practice of Model Development and Use*. Wiley: Chichester, UK.

Schmeiser B (1982). Batch size effects in the analysis of simulation output. *Opns Res* **30**(3): 556–568.

Schriber TJ and Andrews RW (1979). Interactive analysis of simulation output by the method of batch means. In: HJ Highland (ed). *Proceedings of the 1979 Winter Simulation Conference*. IEEE: Piscataway, NJ, pp 513–525.

Shapiro SS and Wilk MB (1965). An analysis of variance test for normality (complete samples). *Biometrika* **52**(3/4): 591–611.

Sherman M (1995). On batch means in the simulation and statistics communities. In: C Alexopoulos, K Kang, WR Lilegdon and D Goldsman (eds). *Proceedings of the 1995 Winter Simulation Conference*. IEEE: Piscataway, NJ, pp 297–302.

Steiger NM, Lada EK, Wilson JR, Joines JA, Alexopoulos C and Goldsman D (2005). ASAP3: A batch means procedure for steady-state simulation analysis. *ACM T Model Comput Simul* **15**(1): 39–73.

Steiger NM and Wilson JR (1999). Improved batching for confidence interval construction in steady-state simulation. In: PA Farrington, HB Nembhard, DT Sturrock and GW Evans (eds). *Proceedings of the 1999 Winter Simulation Conference*. IEEE: Piscataway, NJ, pp 442–451.

Steiger NM, Wilson JR, Lada EK, Alexopoulos C, Goldsman D and Zouaoui F (2002). ASAP2: An improved batch means procedure for simulation output analysis. In: E Yucesan, C-H Chen, JL Snowden and JM Charnes (eds). *Proceedings of the 2002 Winter Simulation Conference*. IEEE: Piscataway, NJ, pp 336–344.

Stuart A and Ord JK (1994). *Kendall's Advanced Theory of Statistics, Volume 1: Distribution theory*, 6th edn. Edward Arnold: London.

Von Neumann J (1941). Distribution of the ratio of the mean square successive difference to the variance. *Ann Math Stat* **12**: 367–395.

Welch P (1983). The statistical analysis of simulation results. In: S Lavenberg (ed). *The Computer Performance Modeling Handbook*. Academic Press: New York, pp 268–328.

White Jr KP (1997). An effective heuristic for bias reduction in simulation output. *Simulation* **69**(6): 323–334.

White Jr KP, Cobb MJ and Spratt SC (2000). Comparison of contemporary truncation heuristics for steady-state simulation. In: JA Joines, RR Barton, K Kang and PA Fishwick (eds). *Proceedings of the 2000 Winter Simulation Conference*. IEEE: Piscataway, NJ, pp 755–760.

Yazdi AT and Wilson JR (2007). IBatch: An autoregressive-batch means procedure for steady-state simulation output analysis. INFORMS07 Conference, Seattle.

Received 2 April 2009;
accepted 24 August 2010 after one revision