Proceedings of the 2016 Winter Simulation Conference
T. M. K. Roeder, P. I. Frazier, R. Szechtman, E. Zhou, T. Huschka, and S. E. Chick, eds.

A PRACTICAL INTRODUCTION TO ANALYSIS OF SIMULATION OUTPUT DATA

Christine S.M. Currie
Russell C.H. Cheng

Mathematical Sciences
University of Southampton
Highfield, Southampton, SO17 1BJ, UK

ABSTRACT

The tutorial will be used to introduce some basic techniques for analysing the output of stochastic simulation models. Using examples, we will describe methods for determining the optimal warm-up length and number of replications as well as introducing ways of using simulation to compare different systems.

1 INTRODUCTION

We only consider the analysis of stochastic, dynamic simulation models in this tutorial. By this we mean that the inputs to the simulation models are random or stochastic. A quick look at the conference proceedings for a typical Winter Simulation Conference will show how diverse the applications of such models are, e.g. factory production lines; the performance of call centers; transmission of information over the Internet; etc.

Techniques will be introduced using worked examples, which have been chosen to represent a range of different types of simulation models. The examples are:

1. Simple queueing system (M/M/1 queue)
2. Model of server usage for online sales
3. Infectious disease model

A more complete description of the examples will be given as we meet them in the text, and analyse their output.

Our aim in this tutorial is that a reader will understand the nature of simulation output data better and will learn a few basic techniques for analysing it effectively. We do not intend to give a complete description of the theoretical underpinnings of output analysis nor to provide full details of all of the methods that we describe. Those who wish to find out more can consult the myriad of excellent books that discuss this topic, e.g. (Banks et al. 2009, Law 2014, Robinson 2014).

The tutorial begins by describing simulation output data, before moving on to provide two methods for estimating the warm up duration in non-terminating simulations. In Section 4 we describe methods for analyzing terminating simulations, before going onto discuss performance measures in Section 5. The penultimate section describes methods for comparing two or more systems, before a brief conclusion.

2 SIMULATION OUTPUT DATA

A simulation user will define a set of performance measures or outputs of interest that they wish to record when running their simulation model. Let us use y to describe the output that we are interested in. For example, y could be the queue length in a grocery store or the throughput of a factory production line. Assume that the simulation model is run n times, with a different set of random numbers in each of the n replications, and that in each replication, the simulation outputs m values for y, where m can take on any

value but commonly will either be equal to 1 or T, the number of observations in each replication. Then, the values

$$\mathbf{y} = \begin{pmatrix} y_{11} & y_{12} & \cdots & y_{1m} \\ y_{21} & y_{22} & \cdots & y_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ y_{n1} & y_{n2} & \cdots & y_{nm} \end{pmatrix},$$

constitute the simulation output data. Simulation output data are random, as we are considering stochastic simulation models. In this sense, running the model is just like observing reality in that each replication of the model will generate a new set of results.

2.1 Autocorrelation

Due to the kinds of systems that we model using simulation, the data often exibit a high degree of **autocorrelation**. If a data series is autocorrelated, there is a dependence between different data points in the series $y_{i1}, y_{i2}, \ldots, y_{im}$, and consequently the output data cannot be regarded as a set of independent data.

Example 1: M/M/1 Queue: For those unfamiliar with queuing theory, the M/M/1 queue is a system in which there is one server or activity that each individual in the system needs to visit or complete, and there is a queue of unlimited size in front of that server or activity. The arrival of individuals into the model follows a Poisson process (i.e. the number of arrivals per unit time follows a Poisson distribution and the inter-arrival times follow an exponential distribution); and the service or activity time also follows an exponential distribution. Individuals leave the system as soon as they have completed service. Analytical methods can be used to analyse this system but here, we simulate the process to give us an understanding of autocorrelation in simulation data.

We consider the queue length in an M/M/1 queue where the arrival rate is equal to 1 per second and the service rate is equal to 1.5 per second. Figure 1 shows a trace of the number of people in the queue and, we can see that the points are arranged into a series of wide peaks, separated by long periods of zero-length queues. For example, between $t = 79$ and $t = 88$, the queue builds up and then declines, and the number in the queue in a given time period is very dependent on the number in the queue in the previous time period. These data show a high degree of positive autocorrelation.

In general, data output in different replications of the model can be regarded as independent and consequently, this gives us one mechanism for dealing with the autocorrelation. Alternatively, we can consider results from batches of data providing that the batches are sufficiently large for the correlation between different batch results to be below some threshold level. This is known as the technique of Batch Means and the principle of the method is described in Section 3.2.

2.2 Terminating vs Non-Terminating

Simulations can be described as terminating or non-terminating and Table 1 gives examples of both. Terminating or finite simulations have a definite end, e.g. the end of the working day or the occurrence of some random event. The time of the end event need not be deterministic and consequently the length of the output data is not necessarily the same for each run of the simuation. Conversely, non-terminating simulations have no defined end event and in such situations we are usually interested in the steady-state behavior of the system. The difficulty with analysing non-terminating simulations is determining when the steady-state has been reached.

3 NON-TERMINATING SIMULATIONS: WARM UP AND RUN LENGTH

Non-terminating simulation models can be used to describe systems which keep going in the same way indefinitely. For example, models of factory production lines are often non-terminating models because the behavior of the line once the initial transient behavior associated with starting up the line has subsided

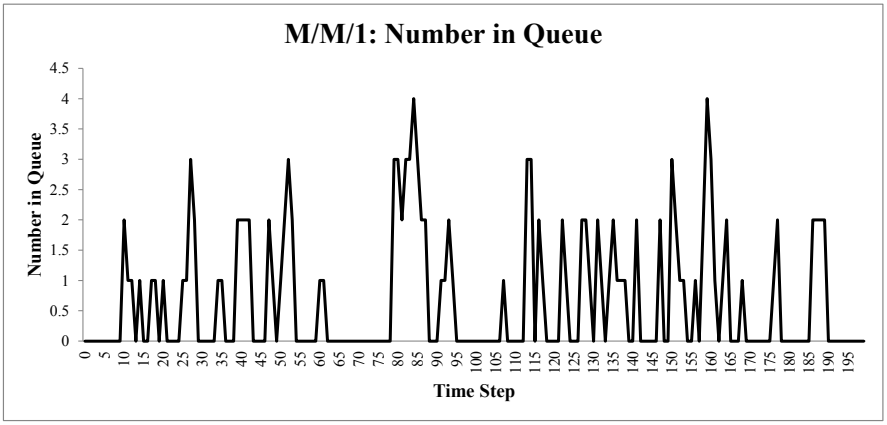

Figure 1: Time series showing the number of people in the queue in each time step for an M/M/1 queuing system.

Table 1: Examples of terminating and non-terminating simulation models.

Example	Classification
Simple queuing system (M/M/1 queue)	Non-terminating
Steady-state performance of server usage	Non-terminating
Battle simulation	Terminating
Transient behavior of servers	Terminating
Rise of an infectious disease epidemic	Terminating

is generally considered most important. Even in factories where production stops at the end of each day, if work in progress is kept on the factory floor, production can be modeled as being continuous. For such simulations, we want to remove any transient results at the start of the simulation model when calculating our statistics of interest to avoid them introducing a bias to the results, i.e. define a **warm up period**. We also wish to determine how long we need to run the simulations for to obtain accurate results. It is important to note that in the steady-state, output is not constant and does still vary, but it will vary according to a fixed or steady-state distribution.

Two approaches exist for dealing with non-terminating simulations. In the first approach, the user runs multiple replications and calculate results excluding observations collected during the warm up period. In the second approach, we can use the method of **b**atch means, described in Section 3.2 in which we observe one long simulation run and collect results in batches.

Example 2: Webserver Example The webserver example is adapted from earlier work (Currie and Lu 2011). The model describes the workings of a webserver in which there are a number of threads available to process requests and a number of users active in the system. Users submit requests to the server, each of which must be processed by a thread. We take our performance measure to be the number of requests waiting in the queue for a thread to become available.

3.1 Warm Up

Figure 2 shows the output from Example 2, which is a model of a web server. It is possible to observe an initial period where the output is increasing, before the data series settles down to what can be regarded as its steady-state behavior. The strange behavior at the start of the simulation run is often termed the **initial transient** and if output from this period is included in the calculation of statistics of interest (e.g. including the initial small values for the number in the queue for the web server example) they are likely to bias the results such that we do not get a good approximation to the steady-state mean. The most common way of dealing with the initial transient is to delete the output from this period, which we define to be the **warm up period**. For example, if we have a series of simulation output data $y_{i1}, y_{i2}, \ldots, y_{iT}$ coming from T observations of a particular model output y during the ith replication, we would calculate the mean for replication i, $\bar{y}_i$, as being equal to

$$\bar{y}_i = \frac{\sum_{t=t_0+1}^{T} y_{it}}{T - t_0},$$

where the warm up period is set to be equal to t_0. Other statistics, such as the variance or range of variables would be calculated in a similar way, i.e. by ignoring the first t_0 observations.

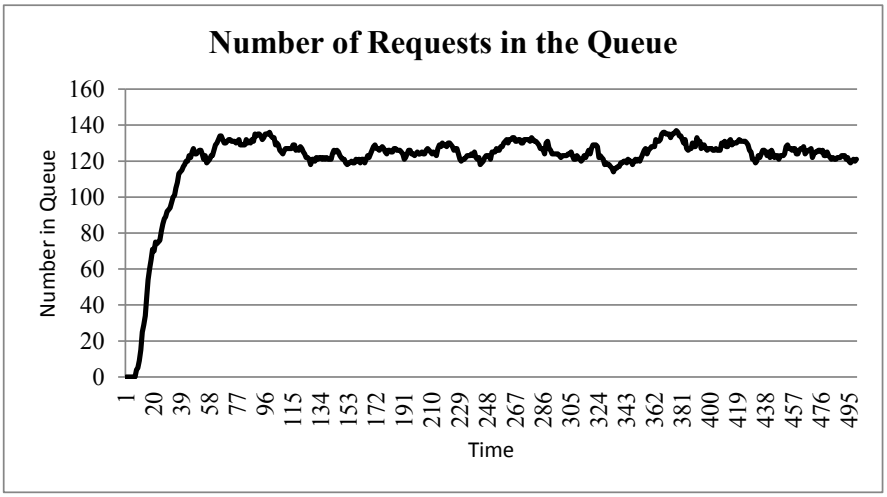

Figure 2: Time series showing the number of requests in the queue for the web server system demonstrating the warm up.

Determining the warm up period is a problem that has been tackled by many authors and in a recent review, Hoad et al. (2010) found 44 different methods proposed to solve this problem. These can be classified into one of five categories, suggested in Robinson (2014):

1. Graphical methods: warm up length is decided by examining the time-series output of statistics of interest
2. Heuristic approaches: use simple rules to determine warm up length
3. Statistical methods
4. Initialisation bias tests: check whether the early data in the time series is biasing the calculation of statistics of interest
5. Hybrid methods: combinations of an initialisation bias test and another truncation method to decide on warm up length

We examine two methods in this tutorial for finding the warm up. The first, Welch's method (Welch (1981) or see Law (2014) for a good description) is a graphical method that is very popular because of its simplicity. The second, MSER-5, put forward by White (1997), is an heuristic method that has received a lot of attention in recent years (see Hoad et al. (2010), Law (2014), Mokashi et al. (2010) for experimental results).

3.1.1 Welch's Method

Welch's method (introduced in Welch (1981) and described in detail in Law (2014)) makes use of the moving average of the output statistic of interest and in principle, it is looking for the point in the time series at which the moving average flattens out. The point at which the time series flattens out is denoted as t_0 and corresponds to the warm up period. Estimating t_0 from just one replication of the simulation model would be difficult due to the variability in the output, and standard practice is to carry out at least five replications to decide on the warm up length (e.g. see Chapter 9 of Robinson (2014)) and to make the run length of the replications, T, as large as possible and much larger than the anticipated length of the warm up t_0. The procedure is as follows:

1. Run the simulation model n times, where $n \geq 5$ to obtain time-series of the output of interest, y.
2. Calculate the mean of the output data for each observation $t = 1, 2, \ldots, T$, $\bar{y}_t = \sum_{i=1}^{n} y_{it}$.
3. Calculate a moving average for the $\bar{y}_t$ using a window size w, where $w \leq \lfloor T/4 \rfloor$, i.e. the window size w should be less than or equal to the integer part of $T/4$ (Law 2014).
4. Plot the moving average $\bar{y}_t(w)$ as a time series.
5. If the data do not look smooth, increase the window size w and repeat steps 3 and 4.
6. The warm up period t_0 is the point at which the time series of the moving average becomes flat.

The moving average of a time series is equal to

$$\bar{y}_t(w) = \begin{cases} \left(\sum_{\tau=1}^{2t-1} \bar{y}_\tau\right)/(2t-1) & t = 1, \ldots, w \\ \\ \left(\sum_{\tau=0}^{2w} \bar{y}_{t-w+\tau}\right)/(2w+1) & t = w+1, \ldots, T-w \end{cases}$$

We apply Welch's method to the output data from the web server example to determine the optimal length of the warm up. Plotting the moving average with a time window of 10, and the number of replications equal to 10 produces the plot in Figure 3. As we can see from the figure, setting the warm up length to around 200 time units should safely remove the initial transient. With more variable data, the decision over the duration of the warm up can be less clear cut.

3.1.2 MSER-5

MSER stands for Marginal Standard Error Rule and is an heuristic method for determining the warm up duration introduced in White (1997). As the duration of the warm up period increases, the accuracy of the calculated result increases due to reduced bias; however, for a fixed total run duration, the precision of the result will decrease because the number of samples available for calculating the average of the performance indicator of interest will decrease. MSER aims to find a warm up length that balances bias reduction and increased precision. Similar to above, let t_0^* be the optimal warm up duration for the output series $y_{i1}, y_{i2}, \ldots, y_{iT}$ for replication i. Run the simulation for n replications, where we suggest $n = 5$ as a good starting point, as with Welch's method above. For $k = 1, 2, \ldots, b$, where $b = \lfloor T/5 \rfloor$,

$$z_k = \frac{1}{5n} \sum_{i=1}^{n} \sum_{j=1}^{5} y_{i,5(k-1)+j}$$

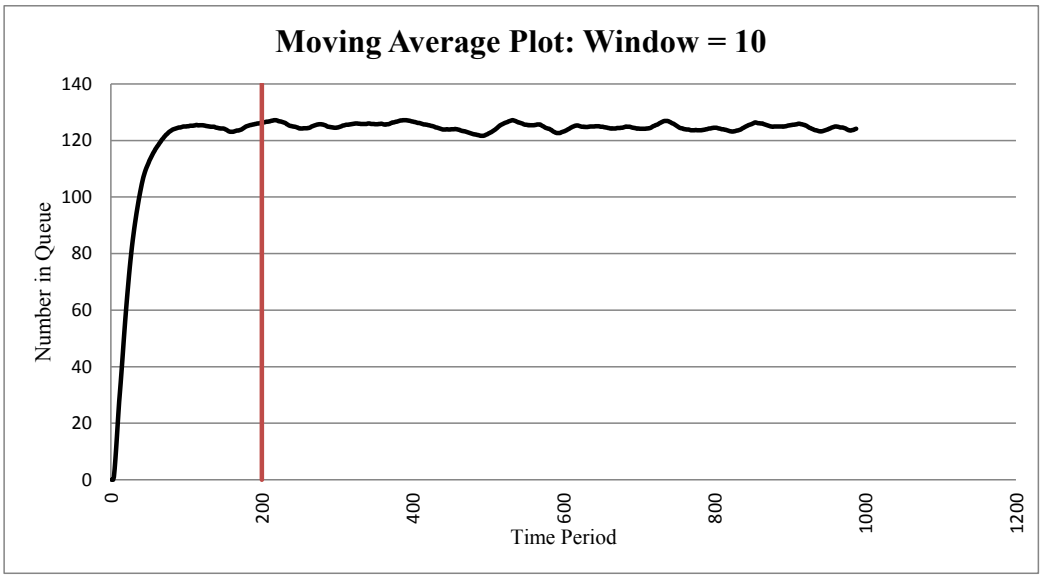

Figure 3: A plot of the moving average of the number of requests in the queue for the web server example, where the window length is set to be 10. The suggested warm up length is denoted by a vertical line.

are defined to be the b batch averages and

$$\bar{z}_{b,t_0} = \frac{1}{n(T-t_0)} \sum_{i=1}^{n} \sum_{t=t_0+1}^{T} y_{it}$$

is the estimated mean, ignoring the warm-up period.

We define b_0 to be the number of batches in the warm up period, such that $t_0 = 5b_0$. Then, the optimal value for b_0 is

$$b_0^* = \underset{b \gg b_0 \geq 0}{\arg\min} \left[\frac{1}{(b-b_0)^2} \sum_{k=b_0+1}^{b} (z_k - \bar{z}_{b,t_0})^2 \right]. \tag{1}$$

Here, we use $\gg$ to indicate that b must be much greater than b_0 and "arg min" to state that we are aiming to find the value of b_0 that minimizes the expression inside the square brackets.

The procedure for finding b_0^* involves evaluating Equation (1) for $b_0 = 1, 2, \ldots, b/2$. The optimal value for the warm up duration, $t_0^* = 5b_0^*$ is then chosen to correspond to the minimum from the $b/2$ results. If the heuristic suggests that the optimal warm up duration is $T/2$ then the sample size T is assumed to be insufficient and no valid result can be obtained.

We apply MSER-5 to the web server example. The plot of the MSER-5 test statistic given in Figure 4 shows an initial sharp decline, followed by an almost constant value. Looking at the numbers output by the method, MSER-5 suggests that the optimal warm up period is 80 time units. Considering the moving average plot in Figure 3, this corresponds to a point shortly before the moving average has flattened out and consequently seems like a reasonable estimate.

We include the MSER-5 rule here for two reasons. First, in experiments it has been found to be a robust method for evaluating the warm up duration (Hoad et al. 2010, White et al. 2000), the exception being output data that are cumulative (i.e. have slowly changing variance throughout), or where there are

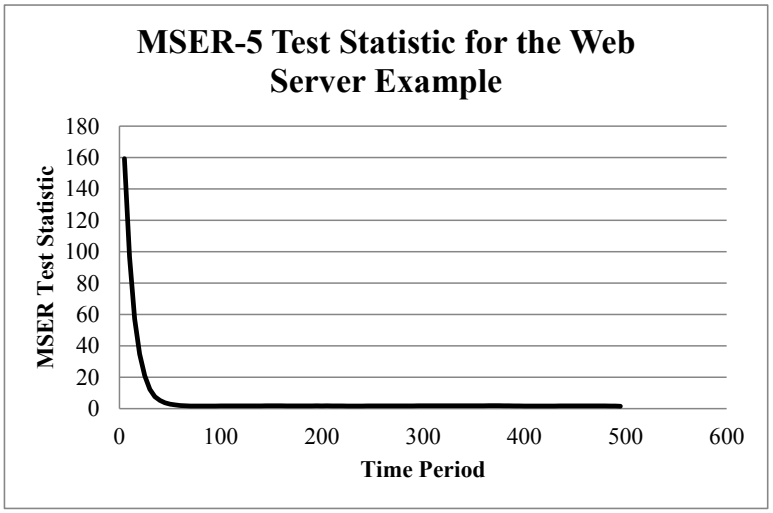

Figure 4: A plot of the MSER-5 test statistic for different possible warm up durations for the webserver example.

insufficient data for a robust analysis. Second, it is also a method that is relatively easy to automate with fairly short computation times. It therefore contrasts well with Welch's method, which we described in the previous section. As with all analyses, when using subjective or heuristic procedures, it is often useful to have a second opinion as a way of cross-checking their validity. In this example, the estimates of the optimal warm up period are very different (200 for Welch's method and 80 for MSER-5). We would be inclined to suggest using a warm up duration of 80 but when running a simulation model with very short computation times, using 200 would certainly reduce the possibility of a biased estimate.

3.2 Batch Means

In the batch means method, we split the series of output data from one replication of the simulation model into b batches of equal length, k such that $b = \lfloor T/k \rfloor$. As we now have just one replication we can drop the replication subscript and use y_j to describe our output. Given a series of outputs for the performance measure $y_1, y_2, \ldots, y_T$, the mean of the lth batch will be calculated as

$$\bar{y}_l(k) = \frac{1}{k} \sum_{j=1}^{k} y_{(l-1)k+j},$$

where $l = 1, \ldots b$. If b is sufficiently large, the set of batch means, $\bar{y}_l(k)$ can be regarded as being a set of independent observations. Determining k is the most difficult part of implementing batch means, and a number of different methods have been proposed. (See, e.g. Robinson (2014) for a good review of these methods). There is some trade off between setting k to be large and having the $\bar{y}_l(k)$'s approximately uncorrelated, but fewer observations, and setting k to be small, leaving more correlation between the batch means but a more accurate estimate of the confidence interval.

4 TERMINATING SIMULATIONS

Terminating simulations are generally easier to deal with than non-terminating simulations because there is no need to determine the warm up duration. This also means that output statistics of interest, such as

means can be calculated using all of the data. Nonetheless, choosing the starting point for the simulation model is important, as discussed in Section 4.1. Output from a terminating simulation is usually **transient** rather than reaching some steady-state, which means that the output data are drawn from distributions that vary with simulation time. It can also change the way in which results should be presented. For example, for a terminating simulation, displaying time series of output data might be more useful than just providing point estimates.

4.1 Choosing Initial Conditions

In general, non-terminating simulations will have a warm up period and terminating simulations will need some care in setting up appropriate initial conditions. For some systems, e.g. a simulation of a shop from when it opens until closing time, the initial conditions might be obvious as the system will start empty. In other situations, e.g. when we are just interested in how the shop functions during a set period, such as the lunch hour, it would be wrong to assume that the system is empty at the start of the simulation period. There are a couple of ways of choosing representative initial conditions:

1. Observe the real system and gather data on the numbers of entities in each activity and queue at the start point
2. Run the simulation model from some earlier time (e.g. when the shop opens) to obtain the starting conditions for the period of interest

The first can be time consuming or even impossible if the simulation model describes a system that has not yet been built. The second might increase the computation time of any experiments that are carried out.

Example 3: Modeling Tuberculosis and HIV This example describes an infectious disease model of tuberculosis (TB) and HIV. A stochastic simulation model was built to describe the impact of the HIV epidemic on the incidence rate of new cases of TB disease in Zimbabwe. Part of the aim of the research was to investigate the impact of different interventions against TB and HIV. Details of the model can be found in Mellor et al. (2011).

Data were available on the incidence of TB disease (i.e. the number of reported cases of TB per year per 100,000 population) from 1980, shortly before the start of the HIV epidemic, until 2002. The epidemic is not expected to reach a steady-state and so a terminating simulation model was built to describe the transmission of disease in the population and the progression of the disease within individuals. The problem is ensuring that the simulation model is in a representative state in 1980 to enable us to reproduce the impact of HIV on the population of Zimbabwe. In this case, a long run length was needed due to the initial oscillation in the output, as shown in Figure 5. To save computational time in running the scenario analysis later in the project, the state of the simulation model at the end of the warm up was saved for a large number of warm up runs, and the first step of each replication of the simulation model was to sample from this set of initial states, to choose the initial conditions for that particular replication.

5 PERFORMANCE MEASURES

A good simulation project will have a clearly-defined purpose and set of questions. These will have helped to dictate the structure of the simulation model and the level of complexity that has been included in it, and can now be used to determine the performance measures of interest. As the name suggests, the performance measures that are calculated will provide some measure of how good the system you are modeling is.

When obtaining results from a simulation model it is very important that more than one run of the simulation model is used to generate them. Consider a non-simulation example of weather prediction as a guide. If you were asked to predict the average rainfall during December in Washington D.C., you would not (or at least should not) be content with looking at data from just one year, but instead, would want to average over as many years' worth of data as are available. The same is true of simulation models. No matter how long your simulation run is, you should carry out several runs or **replications** of your

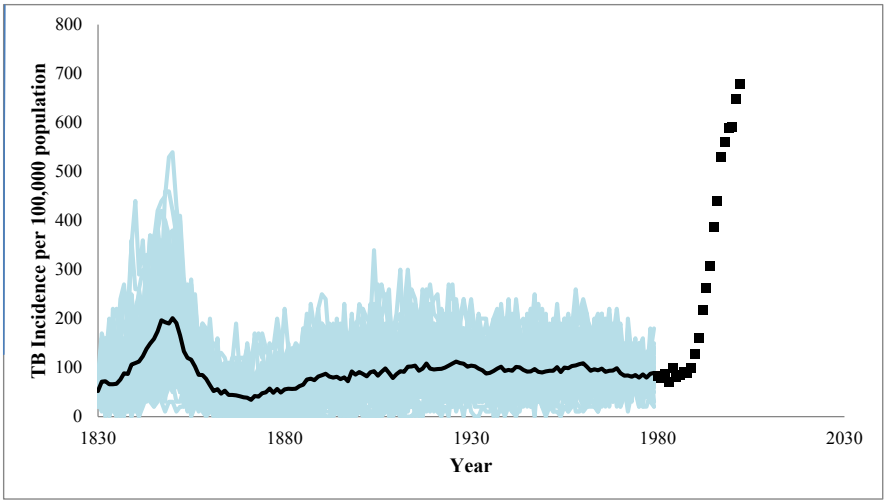

Figure 5: Time series showing the simulation output of TB incidence and setting up the initial conditions. The TB data for Zimbabwe are shown as black squares; results for 100 runs are shown in pale blue; the average model output is shown as a black solid line.

simulation model and use the means (across the replications) of the output performance indicators as your prediction for their values.

It is always good practice to report more than just a number for a performance indicator to give a measure of the robustness of the solution or degree of variability in the output. In this section we describe the calculation of the mean, variance and confidence intervals for simulation output. We also consider the comparison of different effects/treatments on simulation output (e.g. the comparison of several different set ups for a factory production line).

5.1 Estimating Variability in the Performance Measures

Suppose that there is a single performance measure of interest y, and assume that we have measured its mean $\bar{y}$ in n different simulation replications, with the value calculated in the ith replication equal to $\bar{y}_i$. We assume that we have calculated the $\bar{y}_i$ by excluding data from the warm up period, where relevant, as discussed in Section 3.1. Our estimate of the performance measure $\bar{y}$ is then equal to

$$\bar{y} = \frac{\sum_{i=1}^{n} \bar{y}_i}{n}. \tag{2}$$

An initial estimate of the variability of $\bar{y}$ would be to calculate the variance, or alternatively, its square root, the standard deviation,

$$S = \sqrt{\frac{\sum_{i=1}^{n} (\bar{y}_i - \bar{y})^2}{n-1}}.$$

The standard methods for estimating the confidence interval around the estimate of $\bar{y}$ in Equation (2) assume normality. Basically, this requires us to either have a large value of n, or alternatively for the $\bar{y}_i$ to have been averaged from a large sample of data, such that the central limit theorem applies. If the data are not normal, the confidence interval ceases to be valid and so it is important to test for normality.

A simple visual check of normality is a probability plot, where the vertical axis describes the actual quantiles for the data and the horizontal axis describes the expected quantiles, calculated assuming a normal distribution with given mean and variance. The closer the data points are to a straight line, the closer the

data are to being normal. There are also a number of formal normality tests available, and the so-called EDF tests (Anderson-Darling, Kolmogorov-Smirnoff, Cramer-von-Mises) are probably considered to be the most powerful (see e.g. (D'Agostino and Stephens 1986) for how these can be implemented, although most statistics packages will have them built in). The above tests work best with a moderate to high number of data points. Shapiro-Wilk is often useful, and tends to be more appropriate for small numbers of data points.

Assuming the output is normal, the $100(1-\alpha)\%$ confidence interval around the estimate of the performance measure, $\bar{y}$, is given by

$$\left[\bar{y} - t_{1-\alpha/2;n-1}\frac{S}{\sqrt{n}}, \bar{y} + t_{1-\alpha/2;n-1}\frac{S}{\sqrt{n}}\right]. \tag{3}$$

The quantity $t_{1-\alpha/2;n-1}$ describes the point on the t-distribution with $n-1$ degrees of freedom, where the cumulative distribution function is equal to $1-\alpha/2$.

5.2 Choosing the Number of Replications

In an ideal world, you would carry out a large number of very long simulations. If a simulation is run for a long time, the values of the $\bar{y}_i$ obtained will be very similar and consequently, S will be small. This will help to decrease the width of the confidence interval. If a large number of simulation runs are made, the confidence interval will narrow at a rate of $1/\sqrt{n}$ in addition to the t-statistic that is calculated becoming smaller because the tails of the t-distribution get less fat as the degrees of freedom increase. For a fixed computation budget a balance needs to be struck between running a few long simulation replications and a large number of short replications.

In some situations, we might have a pre-specified precision that we would like to achieve at the end of our simulation experiments. If we assume that the run length has previously been specified, we can now estimate the number of replications we need to perform to achieve a desired precision. The precision can be measured in two ways: absolute error or relative error.

We consider first absolute error, where we wish the final estimate of the performance measure, y, averaged over all of the replications to obey $|\bar{y}-\mu| < \varepsilon$, with probability $1-\alpha$, the significance level (e.g. 0.9 or 0.95), where μ is the true mean of the performance measure y. Using Equation (3), we can say that the number of replications, n_ε^a required to obtain a precision of ε is the minimum value of p such that $p \geq n$ and

$$t_{1-\alpha/2;p-1}\frac{S(n)}{\sqrt{p}} \leq \varepsilon,$$

where $S(n)$ is the standard deviation of the $\bar{y}_i$, $i = 1,\ldots,n$.

When using the relative error, we wish $|(\bar{y}-\mu)/\mu| < \varepsilon$, i.e. the the estimator should be within $100\varepsilon\%$ of the true value with probability $1-\alpha$. A similar principle is used to that for the absolute error, but now we need to take into account the mean of the relevant performance measure, $\bar{y}(n)$ calculated after the first n replications. In this case, the number of replications, n_ε^r that are needed for the estimator to be within $100\varepsilon\%$ of the true value with probability $1-\alpha$ is the minimum value of p such that $p \geq n$ and

$$t_{1-\alpha/2;p-1}\frac{S(n)}{|\bar{y}(n)|\sqrt{p}} \leq \frac{\varepsilon}{1+\varepsilon}.$$

For more details of the derivation of these formulae, see Law (2014). There is the potential for inaccuracy in the suggested values of n, the number of replications, especially for small values of n, for which the estimates of $S(n)$ and $\bar{y}(n)$ may not be particularly precise. Nonetheless, they can provide a good ballpark figure for the number of replications that should be made.

5.3 Variance Reduction Techniques

As the name suggests, **Variance Reduction Techniques** are designed to reduce the variance of the simulation output of interest. This allows for increased precision, i.e. narrower confidence intervals, for the performance measures at no additional computational cost. A complete description of variance reduction techniques lies outside the remit for this tutorial and we refer the reader to Kleijnen et al. (2013) for a thorough treatment of the subject. The most useful and arguably the most widely used of the variance reduction techniques is that of **Common Random Numbers** (CRN), which we describe briefly below.

The idea behind the technique of CRN is that when we are comparing different system configurations, we will reduce the variability if we run each of the system configurations under similar conditions. Consider the TB/HIV model of Example 3. We are interested in comparing the effects of different TB and HIV treatment and prevention interventions on the incidence rate of TB disease. If we compare the impact of two different interventions on epidemics with different characteristics (e.g. intervention 1 on an epidemic with a high peak and intervention 2 on a less severe epidemic), we will obtain much more variable results than if each comparison is made on the same epidemic, but the results are run for a range of different epidemics.

A first attempt at applying CRN might be to use the same random number stream for each system configuration; however, a little more care needs to be taken. Many simulation packages will automatically use CRN when running trials and will ensure that they are being applied correctly. When running a simulation of your own you should try to ensure that the same random numbers in the stream are being used to generate the same variables, e.g., for the TB/HIV example, the same random number should be used to generate the initial conditions used and the time to the first transmission of disease for all interventions being tested. Section 11.2 of Law (2014) has more details and further references for how this should be done.

As an example of the use of CRNs, we consider the comparison of two M/M/1 queues both with arrival rate of 1 but with different mean service times of 0.9 and 0.93. Runs were made in pairs, one for each queue. Each run simulated the processing of 5000 customers and recorded the average delay experienced by the customers in the run. Ten pairs of runs were made where all interarrival and service times were independent between pairs, and ten pairs where the same sequence of random numbers were used to generate corresponding interarrival and service times in each run of the pair. The differences in the recorded average delays between the two runs is displayed graphically for the ten independent pairs case and for the CRN correlated pairs case in Figure 6. It will be seen that the measured differences in CRN pairs has a much reduced variance, enabling the difference between the performance of the two queues to be much more clearly identified and measured.

6 MAKING COMPARISONS

The purpose of many simulation studies is to compare a number of different configurations for a system and, usually, to work out which is the optimal based on one or more performance indicators. This problem is one that is well-known in classical statistics and there is a large volume of literature available on experimental design (e.g. Box et al. (2005), Montgomery (2012), Wu and Hamada (2000)). We describe how confidence intervals can be calculated for a comparison between two systems and between many systems.

6.1 Comparing Two Systems

Assume that we wish to consider two different systems. For example, this could be two possible configurations for our queueing system or perhaps two different TB or HIV treatments in the case of our TB/HIV model. We run n replications for each of the two treatments, generating two data series of equal length for our output of interest. Note that in this case, we assume that we generate one output of interest for each replication of the simulation model and suppress the second subscript, $j = 1, \ldots, m$. Often this will be a mean, as we assume here, but it could also be a variance or some other measure of interest. The configuration is

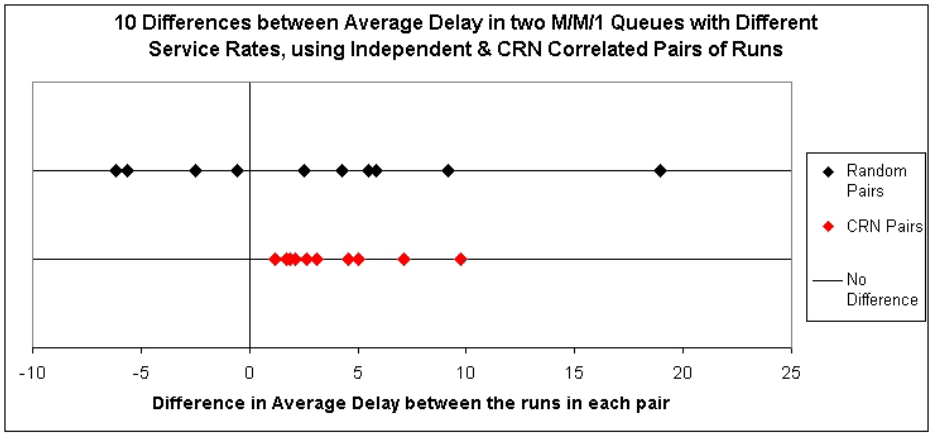

Figure 6: Comparison of two M/M/1 queues, both with arrival rate of unity, but with different mean service times of 0.9 and 0.93 respectively. Each run simulated 5000 customers, and were made in pairs, one for each queue. The plotted points are of the difference in the average delay experienced by customers in the two runs.

indicated inside the brackets, i.e. $\bar{y}_1(1), \bar{y}_2(2), \ldots, \bar{y}_n(1)$ and $\bar{y}_1(2), \bar{y}_2(2), \ldots, \bar{y}_n(2)$. We define Δ_i to be the difference between $\bar{y}_i(1)$ and $\bar{y}_i(2)$, such that

$$\Delta_i = y_i(1) - y_i(2), i = 1, \ldots, n.$$

We are interested in whether the distribution of the Δ_i simply describes random noise around zero (i.e. there is no difference in output between the two configurations) or describes a signal with non-zero mean. The sample mean of the Δ_i is equal to

$$\bar{\Delta}(n) = \sum_{i=1}^{n} \Delta_i / n$$

and its variance is given by

$$\mathrm{V}ar[\bar{\Delta}(n)] = \frac{\sum_{i=1}^{n} (\Delta_i - \bar{\Delta}(n))^2}{n(n-1)}.$$

Assuming that the data follow a normal distribution, the $100(1-\alpha)$ percent confidence interval around the sample mean $\bar{\Delta}(n)$ is equal to

$$\bar{\Delta}(n) \pm t_{n-1,1-\alpha/2} \sqrt{\mathrm{V}ar[\bar{\Delta}(n)]},$$

where $t_{n-1,1-\alpha/2}$ is the $1-\alpha/2$ point of the t-distribution with $n-1$ degrees of freedom. This can be looked up from tables, Excel or a statistical package.

Returning to Example 3, modeling of TB and HIV in Zimbabwe, we might be interested in comparing two methods for detecting new cases of TB disease: intervention 1, in which we visit households in which someone has previously been diagnosed with TB disease; and intervention 2, in which we visit households where someone is in the later stages of HIV. We make 50 replications of the simulation model for each of these interventions and record the number of cases of TB disease that are found in each replication. Results are displayed in a box plot in Figure 7 and suggest that intervention 2 is superior to intervention 1.

In order to evaluate the difference between the two interventions, we follow the procedure given above. First, we run the Anderson-Darling test to determine the normality of the differences between the two data series. This is a hypothesis test with null hypothesis, H0: the data are a random sample from a normal distribution. The p-value we obtain for this test is 0.090. Therefore, using a significance level of 0.05, we

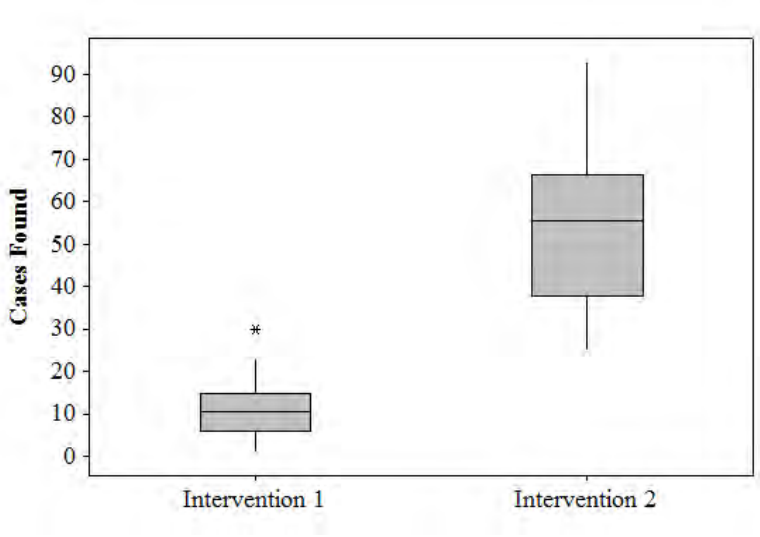

Figure 7: Box plot showing the number of cases of TB disease found using intervention 1 and intervention 2.

cannot reject the null hypothesis that the data are normal, i.e. it is valid to continue with our calculations of confidence intervals assuming normality.

The mean difference between the data series is 43 cases and the variance is equal to 170. With 50 data points, the t-distribution has 49 degrees of freedom and consequently, an estimated 90% confidence interval for the difference in means is given by $[-21,-65]$, where the negative sign is because we use the definition $\Delta_i = y_i(1) - y_i(2)$ given above. Since this confidence interval does not contain 0, we can say that we are 90 percent certain that the mean output from method 1 differs from method 2, and furthermore it appears that method 2 is superior.

The method of Common Random Numbers, described above, can be very useful for reducing the width of the confidence interval for a fixed value of n. If the data are not normal (and Section 5.1 discusses how we can test this), the confidence interval calculated here will not be exact, and in this case other methods are needed for determining whether or not there is a significant difference between the $\bar{y}_i(1)$ and the $\bar{y}_i(2)$. Plotting the Δ_i will help, but techniques such as bootstrapping (e.g. see (Efron and Tibshirani 1998)) may be helpful in determining whether the difference is significant at the α confidence level.

6.2 Comparing Many Systems

When we compare k alternative systems for $k > 2$, we are simultaneously calculating several confidence intervals. As the number of confidence intervals calculated increases, the confidence we have that **all** of the statements that we make are true decreases. To account for this, we can make use of the *Bonferroni Inequality*.

Assume that we wish to calculate C confidence intervals, i.e. we are making C comparisons and each comparison has a corresponding significance level $\alpha_l, l = 1,\ldots,C$. Then, the Bonferroni inequality states that

$$Pr[\text{all statements } S_l, l = 1,\ldots,C, \text{ are true}] \geq 1 - \sum_{l=1}^{C} \alpha_l.$$

For example, returning again to Example 3, modeling of TB and HIV in Zimbabwe, we now wish to compare the impact on TB deaths of five interventions. For each intervention, we calculate a 90% confidence interval for the number of TB cases averted compared with the baseline, as shown in Table 2.

Table 2: Comparison of the impact of five interventions on the number of TB cases averted per 100,000 population in Zimbabwe.

Intervention	Mean TB Cases Averted	90% Confidence Interval	98% Confidence Interval
1	1.31	[-0.85, 3.48]	[-1.75,4.37]
2	1.01	[-0.81, 2.51]	[-1.50,3.20]
3	2.46	[-1.41, 6.33]	[-3.01,7.93]
4	55.8	[18.89,92.70]	[3.61, 107.99]
5	5.22	[0.01, 10.43]	[-2.15,12.59]

In this case, we are subtracting the number of TB cases generated by the model when running with each of the five interventions, from the number generated when using the baseline treatment. Consequently, we are making five comparisons. As each individual comparison has a significance level $\alpha_l = 0.10$, the overall significance of the result is $5 * 0.10 = 0.5$, i.e. there is only a 50% chance that all of the confidence intervals contain their respective means. In order to obtain a 90% confidence interval for the set of comparisons, we would need to compute $(100 - 10/5)\% = 98\%$ confidence intervals for each individual result. Table 2 shows the effect that this has on the confidence intervals, making them much wider than before.

As the number of confidence intervals being computed simultaneously increases, for a fixed number of simulation runs, the width of each individual confidence interval increases to achieve the same level of significance. If there is a need to achieve an overall precision, then the effect will be to increase the number of simulation replications. Therefore, it is worth spending some time before running the simulations to ensure that only necessary comparisons are being made. For a more complete description of multiple comparisons, we would recommend Section 12.2 of Banks et al. (2009).

When comparisons are being made between many different system configurations, ranking and selection methods can be used and Chapter 8 of Banks et al. (2009) provides a useful description of how such methods can be applied. Nonetheless, a simple plot showing the mean and confidence intervals for each of the configurations can be valuable in providing an indication of which might be optimal.

7 CONCLUSION

We have presented a small selection of methods here that can be used to analyze simulation output data. Hopefully, these will be the methods that are needed first in a simulation project to obtain simple, but valid results. For projects, where the output analysis required is much more involved, further reading will be necessary. The two main points we would like a reader to remember from this tutorial are: assume nothing about the output data without testing it first (e.g. independence, normality, lack of time-dependence); and plot data before running statistical tests to ensure that the tests are working as you would expect them to and to explain their results.

REFERENCES

Banks, J., J. Carson, B. Nelson, and D. Nichol. 2009. *Discrete-Event System Simulation (5th Edition)*. Prentice-Hall.

Box, G., J. Hunter, and W. Hunter. 2005. *Statistics for Experiments*. John Wiley and Sons Ltd.

Currie, C., and L. Lu. 2011. "Modeling Server Usage for Online Ticket Sales". In *Proceedings of the 2011 Winter Simulation Conference*, edited by S. Jain, R. R. Creasey, J. Himmelspach, K. P. White, and M. Fu, 752–760. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

D'Agostino, R., and M. Stephens. 1986. *Goodness-of-Fit Techniques. (Statistics: a Series of Textbooks and Monographs Vol. 68)*. Marcel Dekker.

Efron, B., and R. Tibshirani. 1998. *An Introduction to the Bootstrap*. Chapman and Hall.

Hoad, K., S. Robinson, and R. Davies. 2010. "Automating Warm-Up Length Estimation". *Journal of the Operational Research Society* 61:1389–1403.

Kleijnen, J., A. Ridder, and R. Rubinstein. 2013. "Variance Reduction Techniques in Monte Carlo Methods". In *Encyclopedia of Operations Research and Management Science*, edited by S. I. Gass and M. C. Fu, 1598–1610: Springer.

Law, A. M. 2014. *Simulation Modeling and Analysis (Fifth Edition)*. McGraw-Hill.

Mellor, G., C. Currie, and E. Corbett. 2011. "Incorporating Household Structure into a Discrete-Event Simulation Model of Tuberculosis and HIV". *ACM Transactions on Modeling and Computer Simulation* 21:26.

Mokashi, A., J. Tejada, S. Yousefi, A. Tafazzoli, T. Xu, J. Wilson, and N. Steiger. 2010. "Performance Comparison of MSER-5 and N-Skart on the Simulation Start Up Problem". In *Proceedings of the 2010 Winter Simulation Conference*, edited by B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, and E. Yücesan, 971–982. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Montgomery, D. 2012. *Design and Analysis of Experiments (8th edition)*. John Wiley and Sons.

Robinson, S. 2014. *Simulation: The Practice of Model Development and Use (2nd Edition)*. Palgrave.

Welch, P. D. 1981. *On the Problem of the Initial Transient in Steady-State Simulation*. IBM Watson Research Center.

White, J. K., M. Cobb, and S. Spratt. 2000. "A Comparison of Five Steady-State Truncation Heuristics for Simulation". In *Proceedings of the 2000 Winter Simulation Conference*, edited by J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, 755–760. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

White, J. K. P. 1997. "An Effective Truncation Heuristic for Bias Reduction in Simulation Output". *Simulation* 69:323–334.

Wu, C., and M. Hamada. 2000. *Experiments: Planning, Analysis and Parameter Design Optimization*. John Wiley and Sons Ltd.

AUTHOR BIOGRAPHIES

CHRISTINE S.M. CURRIE is Associate Professor of Operational Research in Mathematical Sciences at the University of Southampton, UK, where she also obtained her Ph.D. She is Editor-in-Chief for the Journal of Simulation. Christine was a Track Coordinator at WSC 2013 and has 9 published WSC papers. She chaired the UK Simulation Workshop, SW16. Her research interests include mathematical modeling of epidemics, Bayesian statistics, revenue management and optimization of simulation models. Email: christine.currie@soton.ac.uk.

RUSSELL C. H. CHENG is Emeritus Professor of Operational Research at the University of Southampton. He has an M.A. and the Diploma in Mathematical Statistics from Cambridge University, England. He obtained his Ph.D. from Bath University. He is a former Chairman of the U.K. Simulation Society, a Fellow of the Royal Statistical Society and Fellow of the Institute of Mathematics and Its Applications. His research interests include: design and analysis of simulation experiments and parametric estimation methods. He was a Joint Editor of the IMA Journal of Management Mathematics. His email and web addresses are R.C.H.Cheng@soton.ac.uk and http://www.personal.soton.ac.uk/rchc.