

OUTPUT ANALYSIS

OOSimL Simulation Models

Technical Report

José M. Garrido
Department of Computer Science

February 2017

College of Computing and Software Engineering
Kennesaw State University

© 2016 J. M. Garrido

1 Introduction

The goal in analyzing output data from running a simulation model is to make a valid statistical inference about the initial and long-term average behavior of the system based on the sample average from N replicate simulation runs.

Output Analysis is the analysis of data generated by a simulation run to predict system performance or compare performance of two or more system designs. In stochastic simulations, multiple runs are always necessary. The output of a single run can be viewed as a sample of size 1.

Output analysis is needed because output data from a simulation exhibits random variability when random number generators are used. i.e., two different random number streams will produce two sets of output which (probably) will differ. The statistical tool mainly used is the *confidence interval* for the mean.

For most simulations, the output data are correlated and the processes are non-stationary. The statistical (output) analysis determines:

- The estimate of the mean and variance of random variables, or
- The number of observations required to achieve a desired precision of these estimates.

2 Performance Measures

The output data of simulation models are used to help evaluate the *performance* measures of a system.

Normally, the other basic data output from a simulation run is called the trace. This contains the values of the random variables, the times of occurrence of these values (events), and all other data about the state of the system being modeled.

The performance measures or summary statistics reflect the characteristics of the stochastic process being modeled.

The most common measures used to represent the results of simulation runs are:

- The mean (or average), which gives the fundamental characteristic of the data
- The standard deviation, the most common index of dispersion to summarize variability of the data
- A frequency plot, or histogram, the simplest way to represent distribution of the data.

The mean and the standard deviations can be computed by accumulating values of the random variable while the simulation run is carried out.

3 Comparing Sample Data

To obtain reliable *estimates* of the performance measures of interest, certain statistical analysis of the collected data has to be carried out. Two of these analysis methods are:

- Point estimation, which is single value of the sample data of interest;
- Interval estimation, which is a pair of values, $[l, u]$, of the sample data; also called the confidence interval.

With interval estimation, the desired parameter to be estimated lies between the two values, l and u , that define the interval with a given probability. Both point and interval estimations are needed for meaningful data interpretation. It is useful to have the confidence interval be short, and the desired quantity found in the interval with a high probability.

For example, the sample mean, \bar{x} , is an approximation (estimate) of the population mean, μ . If there is a finite number, n , of samples, then there are n estimates of the mean. Since it is not feasible to obtain the exact value of the population mean, one approach to follow is to find the probability that the population mean is within a certain interval $[l, u]$. This probability is expressed as:

$$P[l \leq \mu \leq u] = 1 - q.$$

The interval $[l, u]$ is called the *confidence interval*, and the value $1 - q$ is called the confidence coefficient. The method consists in finding the bounds for the interval, and finding the probability that the population mean (μ) is within that interval.

Simulations results depend on two types of input. First, the system parameters also called the configuration parameters; for example, the number of servers. Second, the workload parameters that are random samples from the various probability distributions; for example, inter-arrival and service periods.

It is important to differentiate between the effects of variations of the random samples and the effects of the configuration parameters. Otherwise, the interpretation of the results will not be correct.

To obtain improved accuracy of the estimates of performance measures, variance reduction methods are used.

4 Statistical Inference

Statistical Inference is a set of methods to assist a decision maker to draw conclusions about a population from a specific sample. There are two major groups of methods:

- Hypothesis testing, which is used in decision making
- Estimation, which involves establishing a degree of accuracy associated with a point estimate.

Point estimates are quantities that are characteristics of a population/distribution, such as μ and σ^2 , and are called parameters. Generally these population parameters are unknown, so these parameters can be estimated from a sample. Since this estimate a parameter by a single number, this process is referred to as *point estimation*. For example, \bar{X} is an estimate of μ and S^2 is an estimate of σ^2 .

When making statistical inferences about the population mean, the goal is to make a valid statistical inference about the value of the mean, μ , based on the value of the sample estimate, \bar{X} .

4.1 Confidence Intervals

A *Confidence Interval* is an interval estimate for a parameter that specifies a level of confidence that provides a way of quantifying imprecision. The goal of a confidence interval procedure is to form an interval, with endpoints determined by the sample, that will contain, or “cover” the target parameter with a pre-specified (high) probability called the *confidence level*.

The Central Limit Theorem (CLT) states that as we increase the number of samples ($N \geq 30$), the distribution of the means will be approximately normally distributed. To apply the CLT:

- Each single run of a stochastic simulation model can be considered a single sample.
- Each independent model replication, where replications are performed using different random number streams, produces another sample point.

4.2 Calculating A Confidence Interval

A $100(1 - \alpha)$ percent confidence interval, for $0 < \alpha < 1$, of the mean is a range of values running from a lower bound to an upper bound wherein we can be $100(1 - \alpha)$ percent confident that the true population mean falls. The quantity α is sometimes

called the *significance level*. For example, for a 95% confidence, $\alpha = 0.05$. The following formula is used to compute the confidence interval.

$$\bar{X} \pm t_{n-1, 1-\alpha/2} \sqrt{\frac{S^2}{n}}$$

In the formula:

\bar{X}	is the sample mean
n	is the number of replications
S_n	is the sample standard deviation
$t_{n-1, 1-\alpha/2}$	is the value from the t-distribution table

For a confidence interval, the level of precision can be calculated where the number of independent replications is known a priori.

To use confidence intervals with specified precision, the number of replications necessary can be computed to obtain a specified level of precision. This procedure uses a half-length of a $100(1 - \alpha)$ percent confidence interval.

The number of replicates to run is calculated based on how accurate you want the sample average from your replicates to be. The accuracy is a measure of how close the sample average is to the true long-term average performance of the system. The following formula is used for calculating the number of replicates to run.

$$N = \left(\frac{t_{n-1, 1-\alpha/2} S_n}{e} \right)^2$$

In the formula:

N	is the number of replications to run
n	is the number of replications already made in a pilot run
S_n	is the standard deviation from the n pilot replications
e	is the desired accuracy in the sample mean
$t_{n-1, 1-\alpha/2}$	is the value from the t-distribution table

5 Terminating and Non-terminating Runs

A terminating simulation run is carried out to study the behavior of a system over a particular time interval. For example, the behavior of a database server during the *peak period* of the day. A non-terminating simulation run is carried out to study the steady-state or long-term average behavior of a system.

5.1 Analyzing Terminating Simulations

In a terminating simulation run, the simulation starts at a defined state and ends when the simulation reaches some other defined state or time. The majority of service systems are modeled as terminating systems.

Analyzing terminating simulations involve carrying out several simulation runs using different seeds for the random number generations. Data is gathered for successive time intervals during the simulation period.

5.2 Analyzing Non-terminating Simulations

In a non-terminating simulation, the long-term average behavior of a system is analyzed. In this analysis, an adequate length of the simulation period needs to be calculated. The simulation runs then are performed to gather data for the statistical analysis of the steady-state behavior of the system.

Non-terminating simulation runs begin with a warm-up, or transient, state and gradually moves to a steady state.

6 Validation of Simulation Models

Validation of a simulation model provides assurance or confidence that the model is sufficiently adequate and appropriate for use according to the purpose of the model. Validation deals with ensuring that the model closely represents the real system. This should address the concerns of users that the model and results of its simulation runs are acceptable within certain range specified by the purpose of the model.

Verification of the simulation model deals with ensuring that the model is correctly implemented. If a model has not been validated, it is unlikely that it will be used in a real-world setting.

In order for a model to be acceptable, the model should be thoroughly verified and validated before use. This chapter presents an overview of several concepts and techniques for verification and validation of simulation models.

7 Verification Techniques for Simulation Models

During the formulation of the conceptual model in model development, a set of assumptions is taken (about the real system). Verification addresses the question of whether the model implements these assumptions correctly. Debugging and low-level testing are some of the techniques used for verification. Formal methods are a more detailed and mathematical set of methods.

Object-oriented modeling and programming provide enhancements to modularity and information hiding and improve developing and debugging large and complex programs and simulation models. Some of the advantages offered by object-

orientation were discussed in previous chapters. Object-orientation helps and enhances the debugging of software that implements simulation models.

7.1 Simulation Traces

The *trace* of a simulation run is a list of the sequence of events that occur when a simulation model is executed. The trace listings help verify the model. These allow the modeler to check the correctness of the model implementation. The trace facility of simulation models should provide selective tracing, the user would be able to select the level of detail in the trace.

Graphical outputs of simulation runs help in summarizing the traces and in displaying indication of the behavior of the model. For example, the graph that displays the length of the queue of customer arrivals. This type of information helps in debugging the model implementation.

7.2 Simulation Tests

Several tests are available for simulation runs. Results for all these tests should be carefully analyzed and documented. Some of these tests are:

- The *seed independence* test. In this test, the results of simulation runs are compared with different seed values. The results should be similar.
- The *parameter test* or *continuity test*. In this test, several simulation runs are carried out for each set of input parameter values. For small changes in the input, small changes in the output should be expected.
- *Degeneracy tests*. These tests check that the simulation runs of the model exhibit good behavior for extreme values of simulation parameters. These extreme cases are the allowed bounds (upper and lower) for the various parameters.
- *Consistency tests*. These tests check that the simulation runs generate similar results for input values of parameters that have similar effects. For example, various values of input parameters could produce the same or similar workload on the model.

8 Validation of Simulation Models

Validation attempts to answer the question of whether the *assumptions* about the real system are correct. However, a simulation model is an approximation of a real

system. Users of the simulation model expect these assumptions to be as close to reality as possible. The model could be invalid in that the assumptions taken are not realistic.

A model is valid for some particular purpose, i.e., under certain specific assumptions. Validation is model dependent; the techniques used in one model may not be applicable to another model.

A great part of the validation effort is concerned with comparing observations about the real system with observations from the simulation model.

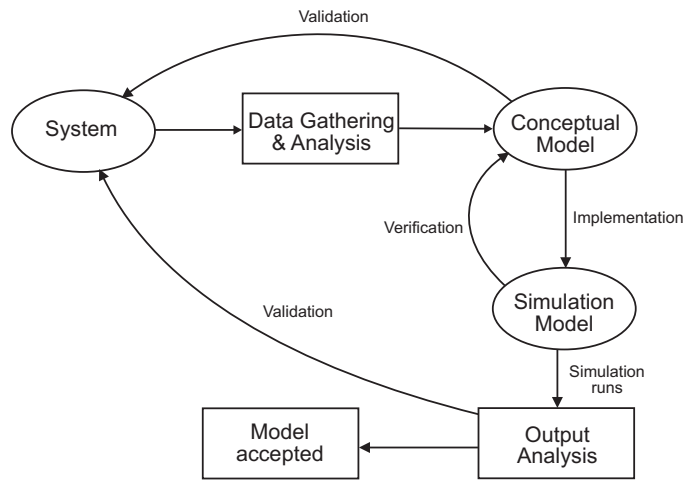


Figure 1: Validation and verification in model development

Figure 1 illustrates verification and validation as part of the model development process. After the model has been accepted as valid, it can be used for conducting experiments, for making inferences about the system represented, and for decision-making.

Validation can be considered at various levels of the model development process.

- At the level of the conceptual model, *conceptual model validity* provides assurance that the assumptions made in specifying the conceptual model are correct.
- At the level of the simulation model, *operational validity* provides assurance that the outputs of simulation runs of the model are sufficiently accurate according to the purpose of the model.
- *Data validity* provides assurance that the data used in model development and conducting experiments is adequate and correct.

8.1 General Validation Approach

Validation of a simulation model involves analyzing:

- Assumptions about the real system
- Values of the input parameters and their distributions
- Simulation results.

Using expert knowledge is the most common way to validate the simulation model. Experts compare results of simulation runs with measurements of the real system. This test is sometimes called *results validation*.

The comparison analysis may use classical statistical tests, such as: t-test, Mann-Whitney, two-sample chi-square, and two-sample Kolmogorov-Smirnov. However, these tests may not be completely applicable because most output processes of simulations and real systems are non-stationary and auto-correlated.

8.2 Black Box Validation

This validation approach considers the real system and the simulation model as black boxes. The only aspect of interest is the results of both, the real system and the model. The external behavior of both are analyzed and compared whether they are acceptably similar.

Two sets of measurements are compared. One set are the measurements of the output from the real system subject to some operational set of conditions. The other set measures the results of the simulation model running under the same (or almost the same) set of conditions. The model is considered *black box* valid if the results of both observations are very close to each other.

An important technique compares the simulation model and the real system by inputting the model with historical system input data, which was used on inspections of the real system. This technique is called *historical data validation*.

Methods of statistical inference are used carefully to analyze the comparison and draw conclusions about the validity of the model. This is relevant when the observations are used to test some hypothesis.

There are two general types of validation errors. Errors of the first type occur when a valid model is wrongly rejected. This can result because there is a percentage of error in the statistical inference as used to test a hypothesis.

The errors of the second type occur when a hypothesis that is false is taken as true, or an invalid model is taken as valid. Another type of validation error occurs when the people carrying out the tests ask the wrong questions.

8.3 White Box Validation

White box validation considers the structure and internal behavior of the real system and the simulation model as known. With black box validation, the main concern is on external behavior and the predictive nature of the simulation model.

The behavior of the simulation model depends mainly on the probability distribution chosen for the random variables. These represent uncertain behavior of the system and implemented in the simulation model. The selection of these distributions can be a very difficult task.

The internal behavior of the model depends on logical rules. These rules should follow the rules that influence the internal behavior of the real system. It is convenient that this type of rule checking is carried out before the model is fully implemented. It is also important to validate the dynamic behavior of the system, not only of its individual components. The trace is a good source for this part of the validation process.