# Model Validation

29.03.2018

What do we mean by "Model Validation"?

There is no way in which we can PROVE that our simulator (model) matches reality - all we could prove is that it does NOT.

What are the stages that we go through in order to determine that the simulator is - as far as we can tell - acceptable?

- Design Phase.

This consists of using

A) our intuitive and mathematical understanding of the process we are trying to model;

B) our experience in designing computer models of reality;

C) techniques of software engineering developed to aid the transition from A) and B) to a codable specification. These might simply include modular decomposition and top-down design, or more sophisticated specification languages and environments.

- Validation of Design

A) Conceptual Phase: determine the logical flow of the system, and formulate the relationships between the various subsystems. Identify the factors likely to influence the performance of the model and decide how you will track them.

Validation by **review**: have a third party examine the design in detail.

Validation by **tracing from output to input**: this is just the opposite from the usual direction, and might indicate gaps and misunderstandings.

- Validation of Design

B) Implementation Phase: selecting procedures for the model; coding the model; using the model.

Validation by checkpoints and milestones: at the end of the model design phase; at the end of the implementation of each module; at the integration of two or more modules.

The latter two require that individual module tests have been identified and designed, and that the flow of information between modules can be examined in detail. This blends into the next phase...

- Verification Phase.

Does the system that has been coded meet the specifications?

At this point, the design might well be "wrong", and this phase is **not** meant to catch design mistakes. It should catch any deviations of the actual implementation from the design specifications.

How: code walk-throughs, careful tracing of module interfaces and module dependencies, tests that were designed at specification time for this purpose.

- Validation Phase

At this point we have a running program that we believe satisfies the design specifications and - because the design was achieved by people with experience in designing such simulation models using some design validation methods - is probably close enough to reality so that its inputs and outputs can be meaningfully compared with those of the "real thing".

We want to establish that the model is an "adequately accurate" representation of the reality we are interesting in modeling. If this cannot be established, our model will be useless for both explanation and prediction - both of which are goals of "good models".

- What does Validation Consist of?

A) Comparison of results of simulation model with historical data;

B) Use of the simulation model to predict behavior of the real system, and compare prediction with actual behavior.

Both of these methods are standard when it comes to validating a physical theory: the theory must not contradict any "well-known facts"; it must provide new predictions (not available through established theory) that are subject to falsification by experiment.

- How do you determine that the system has been validated?

Even in Physics, theoretical predictions usually do not match experimental results exactly.

Solution: run statistical tests, until you can conclude, from statistical considerations, that the probability of some competing explanation being true is smaller than a preassigned quantity.

This means that the system must be designed for multiple controlled observations and the collection of all appropriate data (= measures of performance).

## *Simulation Run*:

This is an uninterrupted recording of the simulation system's performance given a specified combination of controllable variables: range of values, values of some parameter, queue arrival distributions, etc.

## *Simulation Duplication*:

This is a recording of the simulation system's performance given the same or replicated conditions and/or combinations, but with different random varieties.

This is also related to "*regression testing*": a correct model (or piece of software) must satisfy certain input/output relations. Any time the software is modified it must pass all the old tests; if it is "improved", it must pass all the old tests plus appropriate new ones.

## *Simulation Observation*:

This is a simulation run or a segment of a simulation run that is sufficient for estimating the value of each of the performance measures.

# *Steady State or Stable State*:

This state of a simulation system is achieved when successive system performance measurements are statistically indistinguishable - the second one provides no new information about the future behavior of the system.

Steady State corresponds, for example, to the constant solution of a differential equation. A Stable Steady State corresponds to a stable constant solution or, more likely, to an asymptotically stable one.

(A condition of a system that does not change over time)

# How do we identify a *steady state* of some performance measure?

By deciding on a "small" positive value, say ε, and deciding that we have reached steady state when, over a "long" period of time, the performance measure (or some appropriate function of it) has remained within ±ε of some value.

“Small" and "long" are terms that are meaningful only in the context of the particular model being studied.

It may also be possible to actually predict, from the model, the constant value.

# *Transient* (temporary / transitory) *State*:

The time (and set of performance measurements) that correspond to the initial conditions becoming insignificant to the future behavior of the system.

Some systems may have no Steady State, so that no termination for a Transient State could be identified.

Most simulations appear to be interested in examining the steady state behavior. This is appropriate under some conditions - and in the case where modeling based on discrete queueing theory is being used: most such queueing theory results depend on our being able to obtain steady state predictions.

There are other situations where the transient state may be the most important, since that is where system queues (or buffers) might be overloaded. The "usual solution" is to provide enough system capacity to handle "most" transients. This requires that we find good prediction for the size of all "transients" in the measures of performance.

A further problem with transients is that the actual behavior depends on the exact initial conditions of the system.

As it turns out, some systems may have very complex initial conditions - possibly extending over long periods of time.

Some systems may exhibit very complex behavior - so that some initial conditions will tend to one steady state (or, more generally, an "attractor" which may exhibit a very complex geometry) while others will lead to another all without varying the parameters of the system. These are so-called bistable (or multistable) systems.

Studying transient behavior will thus require a large number of runs; a possibly complex geometric analysis of the "phase-space" of the system; the ability to describe and set arbitrary initial conditions for the system; and sophisticated statistical techniques to determine means, variances and other statistics of the relevant performance measures.

If one wishes to avoid dealing with transient behavior, one must be able to specify initial conditions near steady state: this may require being able to "load the model" with a specific history.

In simple cases this might just mean deferring data collection for a period of time; in others it might mean that consistent performance measure values have to be synthesized over a long enough time period and that these measures have to be "inserted" into the behavior of the model.

## *Validation*:

How accurately is the simulation model representing the actual physical system being simulated?

Several terms have been introduced, corresponding to techniques that help in answering this question. One must always remember that there is no way to prove that the model is a faithful reproduction of reality.

All we can prove is that it is not. But we might be able to set up enough different experiments so that passing of all the experimental tests will allow us to conclude that the probability the model is inaccurate is very small.

# Internal Validity.

This is affected by variability due to internal "noise" effects: stochastic models with high variance due to internal processing will provide outputs whose analysis may not be very useful: are the changes in the outputs due to the model or to incidentals of the implementation? (e.g., numerical approximation errors due to the presence of singularities in some of the functions used).

## Face Validity:

Compare model output results with actual output results of the real system.

## Variable-Parameter Validity:

Compare sensitivity to small changes in internal parameters or initial values with historical data. Compare model dependencies with historical data, looking for the same dependencies.

## Event or Time-Series Validity:

Does the model predict observable events, event patterns and variations in output variables?

# Some Ideas about Data Collection (Sampling).

The text discusses ways to determine whether the data collected are correlated or not.

One would like to obtain stochastically independent data sets, or one would like, at least, to determine the level of correlation between data sets.

This is where the covariance - or the coefficient of correlation - comes in, since it allows us to determine something about dependence.

## Repetition:

How many runs and how long should they be?

## Blocking:

How do we avoid the contributions of transient periods (this assumes we are interested in steady-state behavior).

We can attempt to determine whether two runs are independent in the following way.

Let the $x_i$ denote individual observations, $n$ the number of observations. Let the *average* estimated performance measure be.

$$\hat{\mu} = \sum_{i=1}^{n} \frac{x_i}{n}$$

If each if the $x_i$ is independent, the *confidence* of this performance measure is just the estimate of the variance.

$$\hat{\sigma}_{\mu}^{2} = \frac{\sigma^2}{n}$$

If $\{x_i;\ i = 1,\ldots,n/2\}$, $\{y_i;\ i = 1,\ldots,n/2\}$ are two sets of observations, we can try to find out whether they are correlated. We observe that the mean of the union of the two sets can be written as

$$\mu = \frac{1}{n/2}\sum_{i=1}^{n/2}\frac{x_i + y_i}{2}$$

The *variance* can be written as $\hat{\sigma}_\mu^2 = \frac{\sigma^2}{n}(1+\alpha)$

Where $\alpha$ is the *replication correlation coefficient*. The formula can be derived through the following observations:

$$Var\left(\frac{X}{2}+\frac{Y}{2}\right) = Var\left(\frac{X}{2}\right) + Var\left(\frac{Y}{2}\right) + 2Cov\left(\frac{X}{2},\frac{Y}{2}\right)$$

Where $Cov(X,Y) = E(X,Y) - \mu_X \mu_Y = \alpha\,\sigma_X \sigma_Y$

and α is the coefficient of correlation. Using this in the original formula, and under the assumption that the two samples come from the same population (equal population variance):

$$Var\left(\frac{X}{2} + \frac{Y}{2}\right) = \frac{1}{4} Var(X) + \frac{1}{4} Var(Y) + 2\alpha\sigma_{\frac{X}{2}}\sigma_{\frac{Y}{2}}$$

$$= \frac{1}{4}\sigma^2 + \frac{1}{4}\sigma^2 + 2\alpha\left(\frac{1}{2}\sigma\right)\left(\frac{1}{2}\sigma\right) = 2\sigma^2(1+\alpha)$$

Since the variance of the means is given by the population variance divided by sample size, we have:

$$\hat{\sigma}_{\mu}^{2} = \frac{1}{n/2}\left(\sigma^{2}_{\frac{X}{2}+\frac{Y}{2}}\right) = \frac{1}{n/2}\,2\,\sigma^{2}(1+\alpha) = \frac{\sigma^{2}}{n}(1+\alpha)$$

If two runs (replications) are independent, then $\alpha=0$, since they must behave exactly as one run of twice the length (i.e. $n$). We now have a way to test whether two successive runs are correlated or not.

To see what negatively correlated runs can do, assume we have obtained the following set of observations:

X = (0.3211106933, 0.3436330737, 0.4742561436, 0.5584587190, 0.7467538305, 0.3206222209e-1, 0.7229741218, 0.6043056139, 0.7455800374, 0.2598119527, 0.3100754872, 0.7971794905, 0.3916959416e-1, 0.8843057167e-1, 0.9604988341, 0.8129204579, 0.4537470195, 0.6440313953, 0.9206249473, 0.9510535301)

from a uniform random number generator. The "complementary" set (1 - r, for each r in the first set) is

Y = (0.6788893067, 0.6563669263, 0.5257438564, 0.4415412810, 0.2532461695, 0.9679377779, 0.2770258782, 0.3956943861, 0.2544199626, 0.7401880473, 0.6899245128, 0.2028205095, 0.9608304058, 0.9115694283, 0.395011659e-1, 0.1870795421, 0.5462529805, 0.3559686047, 0.793750527e-1, 0.489464699e-1)

The coefficient of correlation is given by Maple V as

$\alpha$ = describe[covariance](X,Y)/sqrt(describe[variance](X)*

describe[variance](Y)) = -.9999999996

Very close to -1. What is the actual variance for the joint population? The formula we use must lead us to a run of 20 items where each item is the mean of the corresponding two items in the separate populations.

But $(x_i + y_i)/2 = (r_i + (1 - r_i))/2 = 1/2$, $\mu$ = 1/2 and the sample variance is given by

$$\sum_{i=1}^{n/2} \frac{\left((x_i + y_i)/2 - \mu\right)^2}{n/2 - 1} \equiv 0$$

Negative correlation leads to a smaller variance than independence, while positive correlation leads to a larger variance.

If we want to estimate means and variances of different runs - i.e. runs with different conditions. We have estimated sample means and variances to be

$$\hat{\mu}_1, \hat{\mu}_2, \hat{\sigma}_1^2, \hat{\sigma}_2^2$$

The mean and variance of the difference are

$$\hat{\mu}_D = \hat{\mu}_1 - \hat{\mu}_1, \hat{\sigma}_D^2 = \hat{\sigma}_1^2 + \hat{\sigma}_2^2 - 2\alpha\hat{\sigma}_1\hat{\sigma}_2$$

Where $\alpha$ is the usual coefficient of correlation. The difference in the formula is due to the difference for the statistics - rather than the sum. It is immediate to see that positively correlated runs will diminish the variance of the difference. To check whether $\hat{\mu}_D = 0$ positively correlated runs will make the variance small.

One of the problems is the elimination of transient information. In this case it will be advisable to have long runs, in which the early part has been ignored.

One method that helps find out a reasonable length for the run involves computation of the *autocorrelation function*. This is defined as:

$$\alpha(\Delta) = \frac{E\left[\left[x_t - \mu\right]\left[x_{t+\Delta} - \mu\right]\right]}{\sigma^2}$$

Where $x_t$ is an observation at time $t$; $\mu$ is the mean of the observations and $\sigma^2$ is their variance. Note that $\Delta = 0$, gives that $\alpha(0) = 1$.

The sample mean is computed in the usual manner, while the variance of the sample means is given by the formula

$$\hat{\sigma}_{\mu}^{2} = \frac{\sigma^{2}}{n}\left[1 + 2\sum_{\Delta=1}^{n-1}\left(1 - \frac{\Delta}{n}\right)\alpha(\Delta)\right].$$

Notice that $\alpha(\Delta) = 0$ for all $\Delta > 0$ implies that the observations are independent - and this is reflected in the formula. Correlated observations will thus enlarge the variance of the sample means.

# The Blocking Method

This simply consists of

A) Wait until transients are over;

B) collect successive blocks of observations of length k is such a way that the "block means" satisfy independence conditions.

C) Use the block means as “observations” to compute the sample mean and the variance of the sample means.

The independence conditions can be checked via any of the methods already mentioned.