

TRUNCATION POINT ESTIMATION USING MULTIPLE REPLICATIONS IN PARALLEL

Falko Bause
Mirko Eickhoff

LS Informatik IV
Universität Dortmund
D-44221 Dortmund, GERMANY

ABSTRACT

In steady-state simulation the output data of the transient phase often causes a bias in the estimation of the steady-state results. A common advice is to cut off this transient phase. Finding an appropriate truncation point is a well-known problem and is still not completely solved. In this paper we consider two algorithms for the determination of the truncation point. Both are based on a technique which takes the definition of the steady-state phase more closely into consideration. The capabilities of the algorithms are demonstrated by comparisons with two methods most often used in practice.

1 INTRODUCTION

The output process $\{X_j\}_{j=1}^n$ of a discrete-event simulator is influenced by the initial state $S(0)$ chosen by the analyst. In general this influence is significant in the beginning and decreases with increasing model time. If one is interested in the steady-state behavior of the model, this initialization bias is obviously undesirable. A common way to reduce the influence of $S(0)$ is to truncate the “most” influenced observations x_0, \dots, x_{l-1} .

Following this strategy the problem is to find an appropriate truncation point l . Many detection rules and methods can be found in the literature. Some are based on one simulation run (see (Pawlikowski 1990)) and some employ many independent simulation runs starting with the same initial state (cf. (Fishman 2001, Welch 1983)). The latter approach is called multiple (independent) replications in parallel (MRIP, see (Ewing, Pawlikowski, and McNickle 1999)) and will be considered in this paper.

Let $F_j(x|S(0)) := \Pr[X_j \leq x|S(0)]$ denote the cumulative distribution function (CDF) at model time index j . Assuming an ergodic system we have $F_j(x|S(0)) \rightarrow F(x)$ if $j \rightarrow \infty$. $F(x)$ is called the steady-state distribution and the primary concern of steady-state simulation is to

determine this distribution or derived measures. In practice, the calculation of $F(x)$ is only possible for special cases, because the output process might never reach steady-state in finite time. By eliminating the transient phase, the steady-state distribution or derived measures are estimated on the basis of the observations x_{l+1}, \dots, x_n , hopefully giving results close to the theoretical steady-state distribution. In the following we will call the observations x_{l+1}, \dots, x_n the steady-state phase. In the literature the steady-state phase is, e.g., defined by the remark “relatively free of the influence of initial conditions” (Fishman 2001) or by the remark $X_{l+1}, X_{l+2}, \dots, X_n$ “will have approximately the same distribution” (Law and Kelton 2000). These definitions are not precise enough for an algorithmic detection of the truncation point l and the practical knowledge of the analyst is still very important. It is therefore not much surprising, that the most common methods for detection of the truncation point are based on a visual inspection of the output data. Other, completely algorithmic methods give only proper results under special conditions (see (Gafarian, Ancker, and Morisaku 1978)) and are mostly based on one long simulation run. Nearly all methods use as a criterion convergence of the mean, although the steady-state phase is defined by the convergence of $F_j(x|S(0))$ towards the steady-state distribution $F(x)$.

In this paper we focus on the output analysis of MRIP. One main advantage of k independent replications is, that it is possible to get k random values of $F_j(x|S(0))$ and thus an estimation of the CDF of $F_j(x|S(0))$. The accuracy of this estimation increases with an increasing number of replications. A significant advantage of MRIP is that the execution of k replications in parallel needs about the same amount of time than one single simulation run (provided we use, e.g., k computers), but gives more data for a better estimation of $F_j(x|S(0))$. Since the prices of hardware are decreasing and performance is increasing, the execution of MRIP becomes a feasible method in practice.

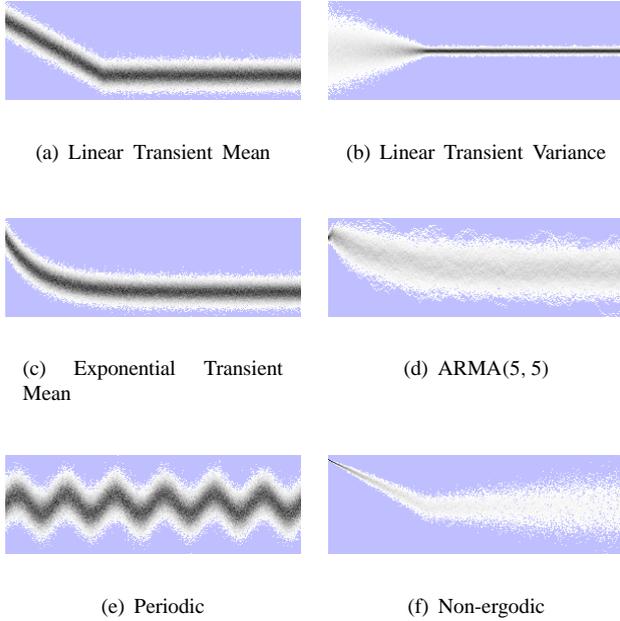


Figure 1: PDF Over Time: Abscissa Represents Index of Model Time; Ordinate Shows Range of Y Divided Into Equally Spaced Y -intervals (A Dark Gray Point at Interval v Represents a High Probability of $[Y_t \in v]$, while a Light Gray Represents a Low Probability)

2 MULTIPLE REPLICATIONS IN PARALLEL

Let $\tilde{F}_j(x|S(0))$ denote the empirical CDF of the k values from the independent replications. $\tilde{F}_j(x|S(0))$ is an estimation of $F_j(x|S(0))$ and, of course, the accuracy of this estimation increases with increasing k . Let $x_{i,j}$ denote the j th observation of the i th replication with $1 \leq i \leq k$ and $1 \leq j \leq n$. If all $x_{i,j}$ are observations of the same measure and all $x_{1,j}, \dots, x_{k,j}$ are observed at the same model time or event, than the sequence $\{x_{i,j}\}_{i=1}^k$ can be considered as an independent random sample of X_i . Let $\bar{x}_j = \frac{1}{k} \sum_{i=1}^k x_{i,j}$ denote the column average.

2.1 Method of Fishman

It is the nature of simulation that all estimated measures have a random error. In addition, measures might have a systematic error caused by the transient phase. This systematic error is known as the initialization bias and Fishman's advice to detect it, is to examine the sequence of column averages $\{\bar{x}_j\}_{j=1}^n$ (cf. (Fishman 2001)). Increasing the number of replications k decreases the random error, but the systematic error remains unaffected.

1. Compute and plot $\{\bar{x}_j\}_{j=1}^n$.
2. If the graph does not show a suitable "warm-up interval", increase n and goto step 1.

2.2 Method of Welch

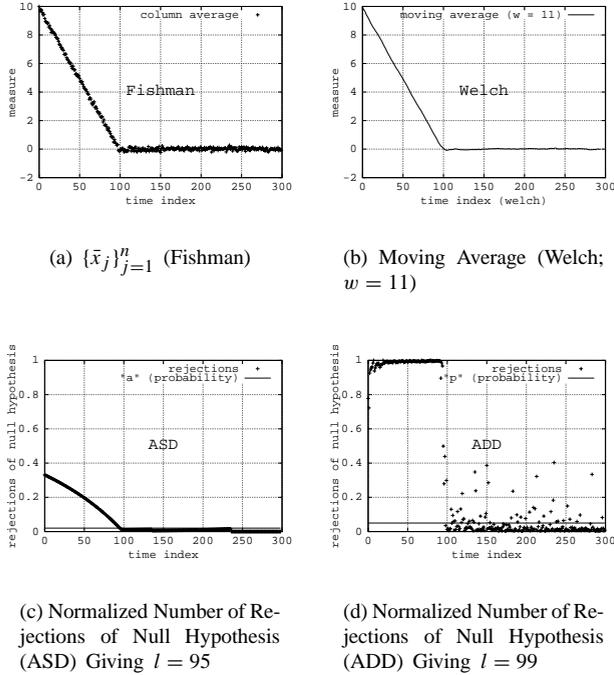
The method of Welch extends the method of Fishman. It is based on the following experiences. The random error appears as a high-frequency oscillation in the sequence of column averages whereas the systematic error is in most cases a low-frequency oscillation. A sliding window comprising a sequence of column averages might be able to reduce the effect of the random error. Given a particular window size we can define an overall average from these column averages. Since the window is moving this average is called the moving average and sliding of the window results in a sequence of such moving averages. In practice, this sequence often seems to converge after some point l giving the wanted truncation point. For details see (Law and Kelton 2000, Welch 1983).

1. Compute $\{\bar{x}_j\}_{j=1}^n$.
2. Move a window of size $w = 2K + 1$ across $\{\bar{x}_j\}_{j=1}^n$. Calculate and plot the average of the interval in each single step. In order to avoid conflicts at the beginning, one starts with a window of size 1 ($K = 0$), which is increased until the desired size w is reached. After that, the window size is kept constant.
3. As long as the plot is not smooth, increase the window size by some value $v > 0$ and repeat step 2 with $w := 2(K + v) + 1$.

As mentioned, the methods of Fishman and Welch require interaction of an experienced user. In the following we introduce two algorithms which need less user interaction. In practice, different situations might be encountered. On the one hand, one might have to deal with a static set of data, e.g., if the simulation has been terminated. On the other hand, one might be confronted with a dynamic set of data. E.g., if analysis of the output data is performed on-the-fly, additional data will become available during the simulation experiment. The following two algorithms take these two possible scenarios into account. One algorithm deals with a static set of data, the other deals with a dynamic set of data. Both aim at the detection of the truncation point l and are based on convergence characteristics of the empirical CDF of the random samples of X_i .

3 STATIC DATASET

Assume that $\tilde{F}_n(x|S(0))$ is the latest estimation of the steady-state distribution $F(x)$, because the simulation terminated at model time index n . In order to find a proper truncation point, the first model time index l must be detected such that $\tilde{F}_j(x|S(0)) \approx \tilde{F}_n(x|S(0))$ for all $j > l$. One measure for equality of two random variables is the maximum difference of their empirical CDFs. The maximum difference of two CDFs X_1 and X_2 is given by $\max_x |F_1(x) - F_2(x)|$ where $F_i(x)$ is the proportion of X_i values less than or equal to x .


 Figure 2: Plots of a Linear Transient Phase ($Y_t^{(1)}$)

Given two empirical CDFs by two samples, the maximum difference can be determined by sorting these samples and comparing indices of corresponding X-values. All this gives the following algorithm.

3.1 Algorithm for a Static Dataset (ASD)

1. Calculate $\tilde{F}_j(x|S(0))$ for $1 \leq j \leq n$ by sorting $\{x_{i,j}\}_{i=1}^k$.
2. Compute the maximum differences $\{d_j\}_{j=1}^{n-1}$ of $\tilde{F}_j(x|S(0))$ and $\tilde{F}_n(x|S(0))$.
3. Compute for all j with $1 \leq j \leq n-1$ the number of differences which miss the threshold in the interval $[j, n-1]$.
4. Choose l to be the minimum value of j after which only $(a * 100)\%$ of the $d_j, d_{j+1}, \dots, d_{n-1}$ exceed the threshold $z_{2,k;1-\alpha}$.

The threshold $z_{2,k;1-\alpha}$ is the same threshold used in the Kolmogoroff-Smirnov two-sample test. Values of the threshold are tabulated. For details see (Hartung, Elpelt, and Klösener 1985). In our experiments we used an α -level of 0.05 and set $a = 0.02$.

4 DYNAMIC DATASET

Analysis of the output data parallel to the execution of the numerous simulations in the MRIP approach has some advantages. Most important is that the analysis process has the opportunity to guide the execution of the replications.

The decision whether to terminate or to continue the simulations can be based on specific analysis results. Thus the set of data is now not limited a priori and its size varies dynamically with proceeding execution of the simulations.

4.1 Algorithm for a Dynamic Dataset (ADD)

1. Choose a ratio $1 : r$, and a level $p, 0 \leq p \leq 1$. Initialize $n := 0$
2. Observe $r + 1$ new X-intervals of all replications and compute the $r + 1$ new random samples: $\{x_{i,n+1}\}_{i=1}^k, \dots, \{x_{i,n+r+1}\}_{i=1}^k$
3. Set $n := n + (r + 1)$
4. Set $TS := \{x_{i,\frac{n}{r+1}}\}_{i=1}^k$
5. Compare TS with $\{x_{i,j}\}_{i=1}^k$ for $j = \frac{n}{r+1} + 1, \dots, n$ using the Kolmogoroff-Smirnov two-sample test. If more than $(p * 100)\%$ of the compared random samples $\{x_{i,j}\}_{i=1}^k$ have a different probability distribution than TS : goto 2. Otherwise terminate with $l := \frac{n}{r+1}$.

In (Bause and Eickhoff 2002) this algorithm for a dynamic dataset (ADD) was first introduced. Similar to ASD, this algorithm is based on the maximum difference of the empirical CDFs. ADD divides the observed random samples into three parts. The first part comprises the random samples, which have been assigned to the transient phase. The second part is the random sample, denoted as the test-sample TS , which is tested whether it is a proper estimation for l . Finally, the third part are those random samples, which are assumed to be in the steady-state phase. This last part might be used to estimate $F(x)$. Note that n is not bounded and therefore the algorithm need not terminate, which, e.g., might happen in case of a non-stationary model.

Since the analysis can be performed in parallel to the simulation, the algorithm might be employed to determine a suitable stopping criteria for the simulation (cf. (Bause and Eickhoff 2002)). One key property of ADD is, that it preserves a predefined ratio of $1 : r$ between the transient and the steady-state phase, following an advice of Law and Kelton (Law and Kelton 2000). Surely, a proper choice of r depends on the model. Our experiences show, that $r = 10$ is a proper choice for many datasets and we used this choice in the following experiments.

Testing whether two random samples originate from the same distribution is again performed using the Kolmogoroff-Smirnov two-sample test. Selecting this test seems reasonable, since it makes no assumptions on the distribution of the X_j . When using ADD we selected an α -level of 0.05 and also set p to 0.05.

5 EXPERIMENTAL COMPARISON

As mentioned, it is well-known that a proper estimation of the truncation point has a significant impact on the quality of

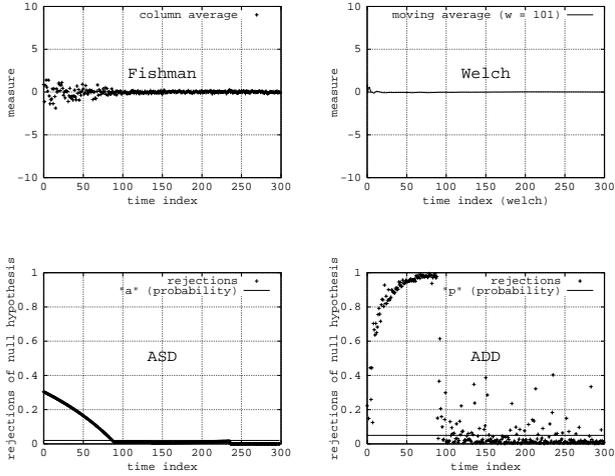


Figure 3: Plots for a Constant Mean with a Transient Variance ($Y_t^{(2)}$) – ASD Gives $l = 87$ and ADD $l = 91$

the simulation results (cf. (Fishman 2001, Law and Kelton 2000, Welch 1983)). Gafarian, Ancker and Morisaku define quality in more detail by attributes like accuracy, precision and generality (see (Gafarian, Ancker, and Morisaku 1978)). Here we want to deal with the quality in a more general view and test ASD, ADD and the methods of Fishman and Welch on some typical attributes of the output data of common models.

In the following examples we use output data, that is not taken from simulations of "real" models. Instead we employ some well-known artificial processes with known properties. Especially characteristics of the truncation point are known in advance giving a precise criterion for comparison. We used the random number generator described in (L'Ecuyer, Simard, Chen, and Kelton 2002) and some additional transformations, described below.

Let $\{\epsilon_t\}_{t=1}^{\infty}$ denote an independent Gaussian white noise process (Hamilton 1994). This process is transformed into six processes with different attributes. In the following we employ $k = 100$ independent replications for each process.

5.1 Linear Transient Mean

The first inspected dataset (see Fig. 1(a)) is a realization of the process

$$Y_t^{(1)} = f_t^{(1)} + \epsilon_t. \quad (1)$$

A transient phase originates from $f_t^{(1)}$ which is defined as

$$f_t^{(1)} = \begin{cases} x - t \frac{x}{T} & \text{if } t < l, \\ 0 & \text{else.} \end{cases} \quad (2)$$

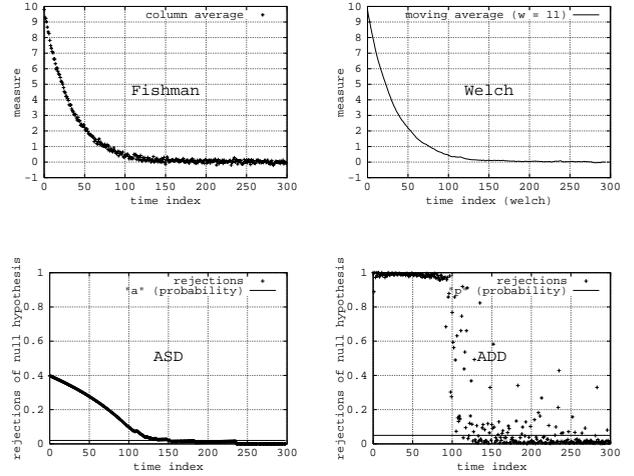


Figure 4: Plots of an Exponential Transient Phase ($Y_t^{(3)}$) – ASD Gives $l = 149$ and ADD $l = 115$

$f_t^{(1)}$ is defined such that the transient phase ends at index l and $Y_t^{(1)}$ is the white noise process beyond index l . The parameter x describes the offset.

Figures 2(a)- 2(d) show results obtained from applying the methods of Fishman and Welch, ASD and ADD for $x = 10$ and $l = 100$. As depicted, all methods give a good estimation of l .

In Fig. 2(c) a cross represents the percentage of all d_h with $j \leq h \leq n - 1$, which exceed the threshold $z_{2,k;1-\alpha}$. In Fig. 2(d) a cross represents the normalized result of all performed Kolmogoroff-Smirnov two-sample tests at one step of the ADD. Normalization is done with respect to all tests, i.e. the normalized number of rejections of the null hypothesis is plotted. At about $t = l$ we find the first random sample being (approximately) equal to all remaining random samples. Obviously, due to randomness some "outlier" test samples still show differences, but the algorithm determines the truncation point accurately near $t = 100$.

5.2 Linear Transient Variance

The second considered process (see Fig. 1(b)) has a constant mean, but there is a transient behavior of the variance.

$$Y_t^{(2)} = f_t^{(2)} \cdot \epsilon_t \quad (3)$$

$$f_t^{(2)} = \begin{cases} x - t \frac{x-1}{T} & \text{if } t < l, \\ 1 & \text{else.} \end{cases} \quad (4)$$

Multiplication in equation (3) stretches the y-range of the white noise process. Again, $f_t^{(2)}$ describes a linear transient behavior ($x = 10, l = 100$).

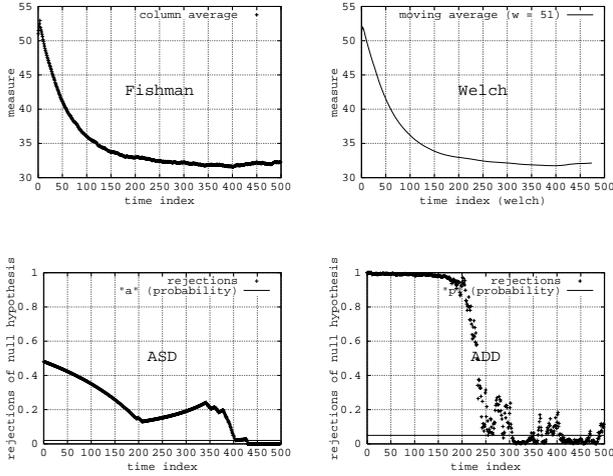


Figure 5: ARMA(5, 5) Process $Y_t^{(4)}$ – ASD Gives $l = 428$ and ADD Gives $l = 288$.

The large variance has an influence on the column averages. Since the random error is larger at the beginning, also the column average has a larger variance at the beginning (see Fig. 3(a)). But with an increasing number of replications, the random error decreases. The moving average smoothes the column averages. The random error is covered with increasing window size. In this example the method of Welch is not able to discover the end of the transient phase correctly (see Fig. 3(b)), because the standard hint to enlarge the window size leads to a smoothed curve without any slope.

Figs. 3(c) and 3(d) show, that ASD and the ADD are able to find an appropriate truncation point, because they are based on the empirical CDFs. The run of the curves in both plots show a sharp drop near the “theoretically best” truncation point.

5.3 Exponential Transient Mean

Our third example considers a dataset (cf. Fig. 1(c)) being realizations of the process

$$Y_t^{(3)} = f_t^{(3)} + \epsilon_t. \quad (5)$$

In contrast to the first dataset exhibiting a linear transient phase, the addend now has an impact which is exponentially disappearing

$$f_t^{(3)} = x \cdot e^{(t \frac{\ln(0.05)}{l})}. \quad (6)$$

The definition of $f_t^{(3)}$ results in permanent differences between two consecutive random samples, but beyond time index l (again $x = 10, l = 100$) a test sample will differ from the steady-state distribution at most by 5% of the dif-

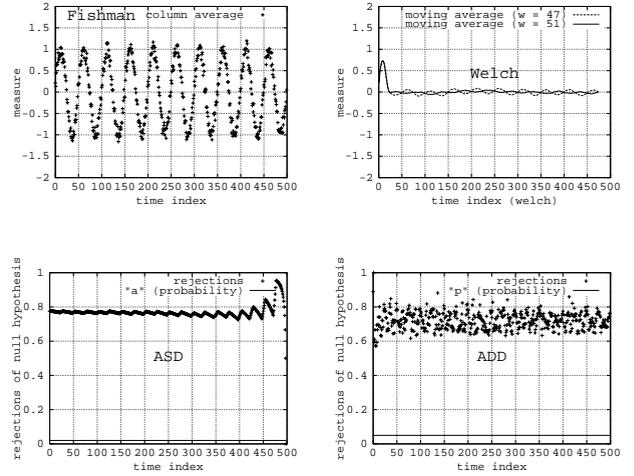


Figure 6: Plots of a Periodic Process ($Y_t^{(5)}$, No Steady-state) – ASD Gives $l = 498$ and ADD Gives No Truncation Point

ference between the first random sample and the steady-state distribution.

In this example there is obviously no clear defined truncation point, but this represents a typical situation encountered in practice. Since the influence of the initial state is not disappearing completely, it is difficult to estimate the truncation point visually (see Fig. 4(a) and 4(b)). A decision where to define the truncation point now depends very much on experience of the analyst.

Figures 4(c) and 4(d) show the results for ASD and ADD. Both algorithms give a truncation point in the interval [100, 150]. Their advantage is the automatic determination of the truncation point due to statistical assumptions.

5.4 ARMA(5, 5)

The fourth dataset (see Fig. 1(d)) is created by multiple realizations of the ARMA(5, 5) process which is defined by

$$Y_t^{(4)} = 1 + \epsilon_t + \sum_{i=1}^5 \frac{1}{2^i} (Y_{t-i} + \epsilon_{t-i}), \quad t \geq 0. \quad (7)$$

We selected $Y_{-5}^{(4)}, Y_{-4}^{(4)}, Y_{-3}^{(4)}, Y_{-2}^{(4)} = 0, Y_{-1}^{(4)} = 100$ as starting conditions giving a transient behavior, since $E(Y_t^{(4)}) = 32$ for $t \rightarrow \infty$.

The results are shown in Fig. 5. The methods of Fishman and Welch give no clear indication of a proper truncation point. Theoretically the sample at $l = 183$ differs at most by 5% from the steady-state distribution. Also ASD gives no proper truncation point. The reason is that the sample at $t = 500$ (i.e. $\tilde{F}_n(x|S(0))$) is an outlier and thus ASD gives a cautious estimation of the truncation point. The analyst would probably reject ASD’s choice and thus avoids running

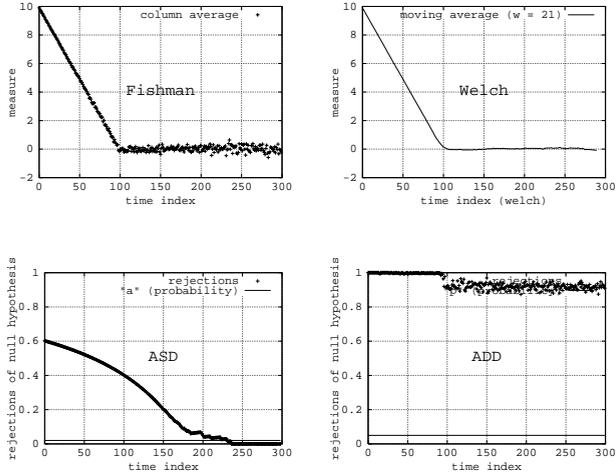


Figure 7: ($Y_t^{(6)}$, No Steady-state) – ASD Gives $l = 231$ and ADD Gives No Truncation Point

into further problems, since the resultant steady-state phase contains too few data for statistical analysis.

Not surprisingly, the results of ADD are the best. ADD determines a truncation point at $l = 288$. Beyond index l there are some peaks, e.g. at $t \approx 300$ and $t \approx 350 - 400$. These peaks result from the memory of the ARMA(5, 5) process. In comparison to the outliers seen before (e.g. in Fig. 2(d)), which are limited to a single point in time, an outlier of an ARMA(5, 5)-process is reproduced in the following 5 values. Thus, many outliers occurring close together will give a curve shown in Fig. 5(d).

5.5 Periodic

A periodic process (see Fig. 1(e)) is non-stationary and has no steady-state distribution. This implies that there is no truncation point.

$$Y_t^{(5)} = f_t^{(5)} + \epsilon_t \quad (8)$$

$$f_t^{(5)} = b \cdot \sin(\omega t) \quad (9)$$

Equation (9) defines a periodic process using a sine oscillation with amplitude b and cycle length $T = \frac{2\pi}{\omega}$.

Of course, the column averages give also a sine oscillation (cf. Fig. 6(a), here $b = 1$, $T = 50$). The method of Welch aims at smoothing the column averages with the advice to increase the window size whenever the resultant curve is not smooth. So there is some chance that the analyst will select (after some experiments on initially, maybe, smaller data sets) a window size close to the cycle length. Selecting a window size which conforms to the cycle length, results in a smooth curve of the moving average (cf. Fig. 6(b)). Based on such a curve, the analyst would be inclined to select an incorrect truncation point.

In contrast to the method of Welch, ASD shows a periodic behavior (cf. Fig. 6(c)), since the differences of the empirical CDFs are oscillating, too. ASD outputs $l = 498$ as truncation point, indicating that the inspected data set shows no steady-state behavior.

Figure 6(d) displays the results for ADD, which again gives the best result. Here the curve remains on a high level and does not reach the threshold, simply because there is no steady-state phase.

5.6 Non-Ergodic

A principal problem when applying simulation is its limited model time horizon. The observed data might satisfy some criteria at the beginning, but after the horizon something “unexpected” might happen. E.g., the system might show non-ergodic behavior (cf. (Bause and Beilner 1999)). Such an “unexpected” behavior is shown in Fig. 1(f) and the corresponding process is defined by

$$Y_t^{(6)} = f_t^{(6)} \epsilon_t + f_t^{(1)} \quad (10)$$

with

$$f_t^{(6)} = ct. \quad (11)$$

$f_t^{(1)}$ again results in a process with a typical initial transient behavior, but due to $f_t^{(6)}$ the process will not become ergodic afterwards. We used the following parameters for experiments: $c = 0.01$, $x = 10$, $l = 100$.

Not surprisingly, the methods of Fishman and Welch fail detecting non-ergodicity, see Fig. 7. ASD outputs a truncation point at $t = 231$, but again this is a cautious choice and the analyst will probably refuse to analyze the resultant short “steady-state phase”. In comparison to the other methods ADD gives a definite result and is able to show that there is no truncation point.

6 COMPARISON OF RUNNING TIMES

It is well-known that the amount of data collected during a simulation might become very large. In the previous section we pointed out the potential benefits of the proposed algorithms, ASD and ADD. In this section we consider the price one has to pay for better results and determine the worst-case time complexity of all the methods considered in this paper.

As before, let k denote the number of replications and n the amount of observations of each single replication. So the total number of observations is kn . Assume, that all basic arithmetic operations are in $O(1)$ (cf. (Cormen, Leiserson, and Rivest 1994)). In the following the mentioned running times consider worst-case time complexity.

Theorem 1 *The running time of the method of Fishman is $O(nk)$.*

Proof: One column average can be calculated in $O(k)$ steps. Altogether n column averages have to be calculated giving a running time of $O(nk)$. ■

Theorem 2 *The running time of Welch's method is $O(nk)$.*

Proof: Since the method of Welch is based on the results of the method of Fishman, it needs a running time of $O(nk)$ and some additional time in order to calculate the moving average. The window slides through the observations in at most n steps. Since each value of the moving average can be calculated by adding one new observation, subtracting the oldest observation and finally dividing by the window size, we have an additional overhead of $O(1)$. Thus the moving averages can be calculated in $O(n)$ steps and the resultant running time of Welch's method is therefore $O(nk) + O(n) = O(nk)$. ■

Theorem 3 *The running time of ASD is $O(nk \log(k))$.*

Proof: Determination of the maximum difference of two CDFs from two random samples of size k has a running time of $O(k \log(k))$, because the random samples have to be sorted. Thus sorting all n random samples needs $O(nk \log(k))$ steps.

The maximum difference of two CDFs (both given by sorted random samples) takes $2k$ steps. Since we do $n - 1$ random sample comparisons, additional $O(nk)$ steps are necessary.

The threshold is tabulated and a lookup is done in $O(1)$ steps. In order to determine the number of differences for all intervals $[j, n - 1]$ ($1 \leq j \leq n - 1$) having missed the threshold, we have to inspect each d_j once. Traversing this range in reverse order, starting with $j = n - 1$, gives the opportunity to rely on previous results. So the amount of differences, which missed the threshold, can be calculated in $O(1)$ per interval. Because there are $n - 1$ intervals this takes additional $O(n)$ steps.

Altogether this gives a running time of $O(nk \log(k)) + O(nk) + O(n) + O(1) = O(nk \log(k))$. ■

Theorem 4 *The running time of ADD is $O(n^2 k \log(k))$.*

Proof: As mentioned, the difference of the CDFs of two random samples can be calculated in $O(k \log(k))$. The lookup of the threshold takes $O(1)$, so one Kolmogoroff-Smirnov two-sample test has a running time of $O(k \log(k))$.

In the p th step of the algorithm pr comparisons have to be done in $\frac{n}{r+1}$ steps. Therefore the amount of comparisons is given by

$$r \sum_{p=1}^{\frac{n}{r+1}} p = r \frac{\frac{n}{r+1} (\frac{n}{r+1} + 1)}{2} = \frac{rn^2 + r^2n + rn}{2r^2 + 4r + 2}. \quad (12)$$

The only maximum of (12) subject to $1 \leq r \leq n - 1$ is at $r = \frac{n+1}{n-1}$ giving at most $\frac{1}{8}(1+n)^2$ comparisons.

Altogether the running time of ADD is given by the number of comparisons multiplied by the running time of one comparison, which results in a total running time of $O(n^2 k \log(k))$. ■

All worst-case running times can be limited by a polynomial and are thus "theoretically efficient". Not surprisingly (having in mind the results from the previous section), the methods of Fishman and Welch are the fastest. But ASD is not significantly slower, since in practice $n \gg k$ holds for most cases, so that the factor $\log(k)$ does not increase the running time significantly. ADD needs the most steps (additional factor $n \log(k)$), but the algorithm can be executed parallel to the execution of all replications (cf. (Bause and Eickhoff 2002)) giving a moderate overhead in practice.

7 SUMMARY AND CONCLUSIONS

The method of Fishman (Fishman 2001) smoothes the simulation data by calculating the mean at each time index. This method is easy to implement and therefore very popular. It creates expressive plots for simple transient behavior (cf. Fig. 2(a), 4(a) and 6(a)). But analysis of the steady-state phase just on the basis of the mean might lead to problems. As Welch remarked (cf. (Welch 1983)), convergence of the mean is a necessary, but not a sufficient condition for stationarity. Therefore, this method is not suitable for the analysis of complex transient behavior. In such cases, it is advisable to compare the results with a plot of the original data (cf. Figs. 1(b) and 3(a)).

The method of Welch (cf. (Welch 1983)) is based on the column average, too. It has the same advantages as the method of Fishman, but suffers also from the same disadvantages. The extension is, that the column averages are smoothed, again by calculating the means of the sliding window. This gives a better distinction between the random and the systematic error. But smoothing might lead to inaccurate results: The moving average is calculated from the means at different points in time and, in general, the process changes over time. E.g., smoothness of the kink in Fig. 2(b) depends on the window size. A more serious problem occurs, e.g., when analyzing periodic processes, especially if the window size conforms to the cycle length (cf. Fig. 6(b)).

ASD and ADD solve some problems of the methods of Fishman and Welch, because they are based on the CDFs and not only on the mean. Thus they take the definition of steady-state better into account.

Even though the implementations of ASD and ADD are not difficult, they are more complicated than the methods of Fishman and Welch. Their execution is more costly, too. But this pays off, when analyzing real-world models with a complex transient behavior.

Table 1: Comparison of Example Results (No Problems: +; Minor Problems: o; Method Fails: -)

Process	Fishman	Welch	ASD	ADD
$Y_t^{(1)}$ ($d = 100$)	+ ($d \approx 100$)	+ ($d \approx 100$)	+ ($d = 95$)	+ ($d = 99$)
$Y_t^{(2)}$ ($d = 100$)	o [50, 100]	- ($d \approx 10$)	+ ($d = 87$)	+ ($d = 91$)
$Y_t^{(3)}$ ($d \approx 100$)	o [100, 200]	o [100, 200]	+ ($d = 149$)	+ ($d = 115$)
$Y_t^{(4)}$ ($d \approx 183$)	o [150, 400]	o [150, 400]	- ($d = 428$)	o ($d = 288$)
$Y_t^{(5)}$ (no l)	+	- ($d \approx 25$)	o ($d = 498$)	+
$Y_t^{(6)}$ (no l)	- ($d \approx 100$)	- ($d \approx 100$)	o ($d = 231$)	+
runtime	nk	nk	$nk \log(k)$	$n^2 k \log(k)$

Table 1 summarizes the results from our experiments. The methods of Fishman and Welch are very useful when estimating a steady mean. If the transient behavior is roughly known, these two methods are adequate. Whenever steady-state can not be deduced from a steady mean or the transient behavior is not roughly known, ASD and ADD are a more adequate choice. Their additional running time is acceptable, since they are able to find a proper truncation point for a larger amount of models.

ACKNOWLEDGMENTS

This research was supported by the Deutsche Forschungsgemeinschaft as part of the Collaborative Research Center "Modelling of large logistics networks" (559).

REFERENCES

- Bause, F., and H. Beilner. 1999. Intrinsic problems in simulation of logistic networks. *Proc. of the 11th European Simulation Symposium and Exhibition (ESS99)*:193–198.
- Bause, F., and M. Eickhoff. 2002. Initial transient period detection using parallel replications. *Proc. of the 14th European Simulation Symposium*:85–92.
- Cormen, T. H., C. E. Leiserson, and R. L. Rivest. 1994. *Introduction to algorithms*. MIT Press.
- Ewing, G., K. Pawlikowski, and D. McNickle. 1999. Akaroa-2: Exploiting network computing by distributing stochastic simulation. *Proc. of the 1999 European Simulation Multiconf*:175–181.
- Fishman, G. S. 2001. *Discrete-event simulation*. Springer.
- Gafarian, A. V., C. J. Ancker, and T. Morisaku. 1978. Evaluation of commonly used rules for detecting steady state in computer simulation. *Naval Research Logistics Quarterly* 78:511–529.
- Hamilton, J. D. 1994. *Time series analysis*. Princeton University Press.
- Hartung, J., B. Elpelt, and K.-H. Klösener. 1985. *Statistik*. R. Oldenbourg Verlag.
- Law, A. M., and W. D. Kelton. 2000. *Simulation modeling and analysis*. McGraw-Hill Higher Education.
- L'Ecuyer, P., R. Simard, E. J. Chen, and W. D. Kelton. 2002. An object-oriented random-number package with many long streams and substreams. *Operations Research* 50 (6): 1073–1075.
- Pawlikowski, K. 1990. Steady-state simulation of queueing processes: a survey of problems and solutions. *ACM Computing Surveys* 22:123–170.
- Welch, P. D. 1983. The statistical analysis of simulation results. In *The Computer Performance Modeling Handbook*, ed. S. Lavenberg, Academic Press:268–328.

AUTHOR BIOGRAPHIES

FALKO BAUSE holds a Ph.D. in computer science from the University of Dortmund. Since 1987 he has taught and done research work in the area of system engineering with emphasis on Stochastic Petri Nets. He initially defined the Queueing Petri Net formalism, which combines Queueing Networks with Stochastic Petri Nets. He also coauthored a book with the title "Stochastic Petri Nets – An Introduction to the Theory". He currently leads a project concerning the modeling and analysis of logistics networks supported by the Deutsche Forschungsgemeinschaft as part of the Collaborative Research Center "Modelling of large logistics networks" (559). His e-mail address is <bause@ls4.cs.uni-dortmund.de>.

MIRKO EICKHOFF holds a Diploma degree (Dipl.-Inform.) in Computer Science from the University of Dortmund. Since the beginning of 2002 he has taught in operating and distributed systems. He has done research work in the area of system modeling and output analysis with emphasis on simulation techniques using multiple replications. He has been involved in a research project concerning the modeling and analysis of logistics networks supported by the Deutsche Forschungsgemeinschaft as part of the Collaborative Research Center "Modelling of large logistics networks" (559). Currently, he works for Delmia (Germany) in the area of workload balancing in manufacturing industry. His e-mail address is <eickhoff@ls4.cs.uni-dortmund.de>.