ON THE PRECISION OF PROBABALISTIC PROCESS GENERATORS IN GPSS

Julian M. Scher, Ph.D.
Department of Computer Science
New Jersey Institute of Technology

ABSTRACT

The precision of several "standard" probabalistic process generators, as provided to the GPSS user in various textbooks and language manuals, are investigated in this paper. In particular, the unique requirements in GPSS for integral increments in its simulation clock, as well as its background integer and truncation features, are seen to have an inherent impact on methodologies for the generation of probabalistic processes. Thus, GPSS requires that we invoke a discrete approximation to any underlying continuous probabalistic process. The GPSS programmer should be keenly aware of these aspects; failure to do so could lead to a model lacking the desired verisimilitude to the object system under study.

A methodology is then developed in this paper which utilizes a least squares approach to yield process generators for the exponential, Gaussian and other continuous distributions. This approach is seen to overcome several of the difficulties associated with the conventional GPSS process generators, and will be useful in numerous instances.

INTRODUCTION

Simulation practitioners who utilize the GPSS language will, for any reasonably sophisticated application, be required to generate random variates from a probabalistic process. When this underlying probabalistic process happens to be continuous, the design philosophy inherent in the GPSS processor will significantly impact the selected generator, since the increments of the simulation clock in GPSS are restricted to integer values. Practitioners in GPSS must therefore be prepared to accept a compromise - a discrete approximation to the underlying continuous probability process. One approach that is often suggested (1) for the minimization of the error in the resultant discrete approximation is in the prudent selection of an appropriate implicit time unit for running the simulation. Adherence to this rule entails that the implicit time unit be chosen as a lowest common denominator of the times to be encountered during a simulation run.

ANALYSIS - THE STANDARD GPSS POISSON PROCESS GENERATOR

The generation of random variates from a Poisson process is equivalent to requiring that the interarrival times follow the exponential probability density function, given by

$$f(t) = ue^{-ut} \qquad t \geq 0$$

where u^{-1} will be the mean interarrival time. An application of the well-known inverse probability transform method (2) provides us with the following generator for simulating exponential interarrival times:

$$t_i = -u^{-1}\ln(1-r_i)$$

where r_i is distributed U(0,1). The "standard" GPSS implementation of this consists of an approximation to $-\ln(1-r)$ with a sequence of 23 straight-line segments, utilizing the GPSS "FUNCTION" and follower cards. The 24 points comprising this approximation may be found in (3) and elsewhere; we shall refer to this function as EXPON and present them below as vectors X and Y, where $X_1 = Y_1 = 0$, $X_2 = .1$, etc.

$X = r_i$	$Y = -\ln(1-r_i)$
0	0
.1	.104
.2	.222
.3	.355
.4	.509
.5	.690

.6	.915
.7	1.2
.75	1.38
.8	1.6
.84	1.83
.88	2.12
.9	2.30
[illegible]	[illegible]
.94	2.81
.95	2.99
.96	3.2
.97	3.5
.98	3.9
.99	4.6
.995	5.3
.998	6.2
.999	7.0
.9998	8.0

Following normal GPSS procedures, we assign the mean interarrival time to the "A" field of a "GENERATE" block, with "FN$EXPON" in the "B" field; the GPSS processor will then multiply the two resultant values, and the truncated result becomes the GPSS-approximated exponential variate.

The fundamental postulate (4) which leads to the system of differential equations by which we derive the Poisson process requires that there be a negligible probability of having two or more arrivals in a given instant of time. An equivalent interpretation of this is that the probability of observing an interarrival time of zero should be negligible. Schriber (5) cautions us that for mean interarrival times below 50, the suggested GPSS generator seriously violates this postulate. To reiterate his example, the block "GENERATE 5,FN$EXPON" will, via the inherent truncation mechanisms referred to earlier, yield an interarrival time of zero more than 18% of the time (whenever the value generated from EXPON is less than .2). By making the implicit time unit smaller by a factor of 10, obtaining an equivalent interarrival time of 50 (the minimum recommended by Schriber), the basic Poisson postulate will still be violated about 2% of the time.

For given u^{-1} (which will be assumed to be integer), we can easily derive an expression to evaluate the percentage of time our Poisson postulate is violated by the standard GPSS exponential generator.

Let $n = \min_i (Y_{i-1} < u \leq Y_i) \quad i=2,\ldots,8$

We then obtain

Prob(GPSS yields a zero interarrival tm)

$= \text{Prob}(u^{-1} * (\text{EXPON variate}) < 1)$

$= P(\text{EXPON variate} < u)$

Then, due to the linear interpolation which occurs in GPSS, the above probability becomes

$$X_{n-1} + \frac{(u - Y_{n-1})}{(Y_n - Y_{n-1})} * (X_n - X_{n-1}) .$$

In (6), we have evaluated this expression for values of u^{-1} ranging from 2 (which yields a probability of GPSS violating the Poisson assumption close to 40% of the time) to 400 (where the postulate is violated slightly more than 2 times out of a thousand).

A desirable characteristic for a GPSS process generator would be that it provides the same expected value as the probability distribution it seeks to mimic (we might refer to this as the unbiased property of the generator). It should be clear, however, that the expected value of the standard GPSS exponential generator should be less than 1/u, due to the inherent truncation mechanisms in GPSS. We may derive a precise expression for the expected value for the probability distribution inherent in the GPSS standard exponential generator as follows:

$$E(Y) = \sum_{k=1}^{8u^{-1}} k * P(\text{ GPSS interarrival time} = k)$$

$$= \sum_{1}^{8u^{-1}} k * P(\left[u^{-1} * \text{ EXPON variate}\right] = k)$$

Since values from EXPON are bounded above by 8, and since, if a pseudorandom number is generated $\geq .9998$, then the value assumed by GPSS for EXPON is 8, the above summation is equivalent to

$$\sum_{k=1}^{8/u\ -1} k * P(ku \leq \text{EXPON variate} < (k+1)u) + .0016/u$$

Invoking the imbedded linear interpolation which occurs, the above summation becomes

$$\sum_{k=1}^{8/u\ -1} k(X_{n(k+1)-1} - X_{n(k)-1} + \frac{((k+1)u - Y_{n(k+1)-1}) * (X_{n(k+1)} - X_{n(k+1)-1})}{(Y_{n(k+1)} - Y_{n(k+1)-1})}$$

$$-\frac{(ku - Y_{n(k)-1})*(X_{n(k)} - X_{n(k)-1})}{(Y_{n(k)} - Y_{n(k)-1})} ,$$

where $n(k)=\min_i(Y_{i-1} \leq ku \leq Y_i) \quad i=2,\ldots,24.$

The expected value of an exponential distribution with parameter u is 1/u; in (6) we have tabulated the expected value of the GPSS exponential generator for selected values of u, with results generally indicating that it will underestimate the true mean interarrival by about .4 to .5, depending on the parameter u. A comparable expression for computing the variance is also presented there.

In actuality, one might maintain that since the standard GPSS exponential process generator prohibits values above 8/u, what it seeks to discretely approximate is not the usual exponential distribution with which we are most familiar with, but rather an exponential distribution truncated from above, whose general form would be

$$f(t) = \frac{ue^{-ut}}{1-e^{-uT}} \qquad 0 \leq t \leq T .$$

One can readily show that the expected value of the exponential distribution truncated from above will be

$$u^{-1}(1 - \frac{Tu}{(e^{Tu}-1)}) .$$

For our application under study, $T=8u^{-1}$; substituting this into the above expression, one obtains

$$u^{-1}(1 - \frac{8}{(e^{8}-1)}) = .9973u^{-1}$$

or slighty less than u^{-1}. Clearly then, the standard GPSS exponential process generator also underestimates the mean of an exponential distribution truncated from above at T=8/u.

ANALYSIS - THE STANDARD GPSS NORMAL PROCESS GENERATOR

To generate a normal random variate R with mean M and standard deviation S, several references (see (3)) provide us with the standard GPSS process generator for a Normal variate with mean zero and standard deviation equal to one, to which we apply the following well-known transformation to obtain our N(M,S):

$$R_i = S*N(0,1)_i + M .$$

We next present the points comprising the standardized normal generator in GPSS as vectors X and Y.

$X = r_i$	Y = SNORM(N(0,1))
.0	-5
.00003	-4
.00135	-3
.00621	-2.5
.02275	-2
.06681	-1.5
.11507	-1.2
.15866	-1.0
.21186	-0.8
.27425	-0.6
.34458	-0.4
.42074	-0.2
.50000	0.0
.57926	0.2
.65542	0.4
.72575	0.6
.78814	0.8
.84134	1.0
.88493	1.2
.93319	1.5
.97725	2.0
.99379	2.5
.99865	3.0
.99997	4.0
1.0	5.0

The intent of the GPSS standardized normal processor is to exclude the tail area beyond 5 standard deviations from the mean; in actuality, then, what we are being provided with is a discrete approximation to what is referred to as a doubly truncated normal distribution, whose mean will equal zero and whose standard deviation, for our truncation points, will be .9999261 (see (7) for a discussion of some of the properties of doubly truncated normal distributions).

The aforementioned transformation from the standardized normal processor to the desired N(M,S) will take place in a GPSS FVARIABLE block, which has the effect of delaying the truncation until after all the arithmetic operations have been performed. As with the exponential, though, the truncation distorts the desired precision of the discrete approximation.

The most distinguishing property of the normal distribution is its symmetry; we shall demonstrate that the "recommended" GPSS normal process generator does NOT provide this symmetry, but is skewed to the left of the mean. To illustrate, let us suppose one wished to generate Normal random variates with mean equal to 6 and standard deviation equal to 1. To generate the value 5 from this distribution, one would require that the standardized generator yield

$$-1 \leq SNORM < 0.$$

Referring to the aforementioned table for SNORM, we find that a 5, which is one standard deviation below the mean, will be generated by our GPSS process generator over 34% of the time. On the other hand, in order for our GPSS process generator to yield a 7, which is one standard deviation above the mean, the requirement on the standardized generator becomes

$$1 \le SNORM < 2 .$$

Consulting our table, we find that a 7 would be generated about 13½% of the time. Thus, we are more than twice as likely to generate a value one standard deviation below the mean then we are to generate a value one standard deviation above the mean, for this distribution. Continuing in this manner, we find that we will generate a 4 (which is 2 standard deviations below the mean) over 13½% of the time, while only slightly more than 2% of the time will an 8 be generated, which is 2 deviations above the mean (i.e., a factor of 6 here). The table below summarizes the probability distribution for this GPSS discrete approximation to a N(6,1) :

Value	GPSS-generated Prob.
1	.00003
2	.00132
3	.0214
4	.13591
5	.34134
6	.34134
7	.13591
8	.0214
9	.00132
10	.00003
11	0.0

One readily determines that the mean of the GPSS discrete approximation to N(6,1) is 5.5, while the standard deviation is 1.0408265. Thus, the approximation to N(6,1) underestimates the mean by over 8%, overestimates the standard deviation by about 4%, and does not preserve the desired symmetry of the Normal distribution.

Let us derive a general relationship for determining the mean and standard deviation of the probability distribution inherent in the GPSS approximation to N(M,S). Let us define

$$n(k) = \min_i (Y_i \le (k-M)/S < Y_{i+1}) .$$

We shall restrict ourselves to positive values for the domain of k in the following formulas, which invoke the implicit linear interpolation which occurs in the GPSS processor. We shall define F(k) = k, and evaluate the expected value.

Let M be the mean, and S the standard deviation of GNORM.

$$E(Y) = \sum_{M-5S}^{M+5S} F(k)\,Prob(GNORM = k)$$

$$= \sum_{M-5S}^{M+5S} F(k)\,Prob(k \le GNORM < k+1)$$

$$= \sum_{M-5S}^{M+5S} F(k)\,P((k-M)/S \le SNORM < (k+1-M)/S)$$

$$= \sum_{M-5S}^{M+5S} F(k)\Big(X_{n(k+1)} - X_{n(k)} + \frac{\frac{k+1-M}{S} - Y_{n(k+1)}}{(Y_{n(k+1)+1} - Y_{n(k+1)})} * (X_{n(k+1)+1} - X_{n(k+1)}) - \frac{\frac{k-M}{S} - Y_{n(k)}}{Y_{n(k)+1} - Y_{n(k)}} * (X_{n(k)+1} - X_{n(k)})\Big).$$

It is clear that we can obtain the second moment about the origin by substituting $F(k) = k^2$ in the above expression, from which we can readily obtain the variance by subtracting from the second moment the square of the mean. A FORTRAN implementation of this revealed, for various N(M,S), that the discrete approximation consistently underestimated the true mean by .5, while always overestimating the standard deviation.

To recapitulate our analysis, we have found that the discrete Normal approximation provided to us in GPSS texts and manuals

i) provides a skewed distribution;
ii) consistently underestimates the desired mean of the Normal by .5;
iii) overestimates the standard deviation.

THE LEAST SQUARES PROCESS GENERATION THEOREM FOR DISCRETE APPROXIMATIONS

We shall develop a methodology for probabalistic process generators in GPSS which will overcome several of the difficulties with current probabalistic process generation techniques caused by the imbedded truncation features in the language. Our philosophy is to accept as inevitable the truncation mechanisms inherent in GPSS, and to develop an approach which incorporates this truncation into a well-defined procedure. The methodology to be developed is seen to be a special case of a more general line-segmenting approximation technique presented by Stone (8).

Theorem 1. Let the cumulative distribution function of our desired continuous probability process be denoted by F(x), and let us assume that by invoking the Inverse Probability Transform, we obtain $F^{-1}(r)$, where r is U(0,1). Let the discrete approximation to $F^{-1}(r)$ be given by

$$y = \begin{cases} y_1 & u_0=0 \le r < u_1 \\ y_2 & u_1 \le r < u_2 \\ y_3 & u_2 \le r < u_3 \\ \vdots & \vdots \\ y_N & u_{N-1} \le r < u_N \\ y_{N+1} & u_N \le r \le u_{N+1}=1 \end{cases}$$

where $y=(y_1,y_2,y_3,\ldots,y_N)$ is given, and the breakpoints $u=(u_1,u_2,u_3,\ldots,u_N)$ are to be determined. The following equation yields the values for u_i which provides, in the least squares sense, the best discrete fit to the continuous probability process:

$$u_p = F(\tfrac{1}{2}(y_p+y_{p+1})) \qquad p=1,\ldots,N .$$

Proof: By the least squares criteria, we wish to determine $u=(u_1,u_2,\ldots u_N)$ which will minimize

$$Q(u) = \sum_{j=1}^{N+1} \int_{u_{j-1}}^{u_j} (F^{-1}(r) - y_j)^2 dr$$

To determine the normal equations, an application of Leibnitz's rule yields (9)

$$\frac{F}{u_p} = (F^{-1}(u_p) - y_p)^2 - (F^{-1}(u_p)-y_{p+1})^2$$

$$= 0 \qquad \text{for } p = 1,\ldots,N$$

Solving for $F^{-1}(u_p)$, we obtain

$$F^{-1}(u_p) = \tfrac{1}{2}(y_p+y_{p+1})$$

from which we obtain

$$u_p = F(\tfrac{1}{2}(y_p + y_{p+1}) .$$

Corollary When $y_p = p$, we obtain the breakpoints by

$$u_p = F(p+\tfrac{1}{2}) .$$

EXAMPLES AND APPLICATIONS OF THE LEAST SQUARES PROCESS GENERATION THEOREM

Let us consider an exponential probabalistic process generator, with $y_p = p$. Invoking the corollary to Theorem 1, and substituting into the cumulative distribution function for the exponential, we have

$$u_p = 1 - e^{-u(p+\frac{1}{2})}$$
$$\text{for } p=1,2,\ldots,N.$$

Since $y_1=1$, our new exponential process generator guarantees that we will never generate an interarrival time of zero, and thus will never violate the fundamental Poisson postulate. Let us next derive an expression for the expected value of an exponential variate generated by our procedure. We have

$$E(Y) = \sum_{j=1}^{N+1} jp_j$$

$$= 1-e^{-1.5u} + \sum_{j=2}^{N} j(u_j-u_{j-1})$$

$$+ (N+1)e^{-u(N+.5)}$$

$$= 1 - e^{-1.5u} +2(e^{-1.5u} - e^{-2.5u})$$
$$+3(e^{-2.5u}-e^{-3.5u})+\ldots+(N+1)e^{-u(N+\frac{1}{2})}$$

$$= 1+e^{-1.5u} +e^{-2.5u}+\ldots+e^{-(N+\frac{1}{2})u}$$

$$= 1+e^{-1.5u}(1+e^{-u}+\ldots+e^{-(N-1)u})$$

$$E(Y) = 1 +e^{-1.5u} \cdot \frac{(1-e^{-Nu})}{(1-e^{-u})}$$

In (6) we have evaluated an equivalent expression for the expected value, with results indicating an error in the generated mean of less than 10% of that of the "standard" exponential generator. Similar expressions and results are readily obtained for the standard deviation of our proposed exponential generator.

If we let q denote the percentage of time that we wish to generate the largest possible interarrival time, N+1, then we may analytically determine the number of breakpoints N by noting that the probability of generating an interarrival time of N+1 is $e^{-u(N+\frac{1}{2})}$. Setting this equal to q and solving for N, we have

$$N = \lceil u^{-1}\ln(q^{-1}) \rceil - \tfrac{1}{2} \rceil .$$

Some broad suggestions for the implementation of this generator in GPSS are presented in (6).

Our attention next turns to the development of a process generator for the Normal distribution. The following theorem will simplify this task considerably.

Theorem 2. In GPSS, to generate discrete Normal variates which best fit, in the least squares sense, the continuous Normal distribution, utilize the following

FVARIABLE V$STDEV*FN$SNORM+V$MEAN+1/2 .

Proof: Without loss of generality, let us evaluate the probability of generating an integer W, where

$$M-5S \le W < M+5S,$$

invoking the Least Squares Process Generation Theorem for a N(M,S) with CDF given by F(x). We readily obtain from our Theorem 1 that a W should be generated whenever, for a random number r,

$$F(W-\tfrac{1}{2}) \le r < F(W+\tfrac{1}{2}) .$$

Denoting the standardized CDF by Z(x), we thereby obtain

$$Z\left(\frac{W - \frac{1}{2} - M}{S}\right) \le r < Z\left(\frac{W + \frac{1}{2} - M}{S}\right)$$

This is equivalent (but see note below) to requiring that

$$\frac{W - \frac{1}{2} - M}{S} \le \text{SNORM} < \frac{W + \frac{1}{2} - M}{S} \qquad (1)$$

Equation (1) them becomes

$$W \le S*\text{SNORM} + M + \tfrac{1}{2} < W+1 .$$

This last result, however, is what we obtain from our FVARIABLE statement in GPSS. Q.E.D.

Note: The proof and implementation of Theorem 2 are dependent upon having a precise evaluation of the probability statement of equation 1, rather than an approximate linear interpolation. Depending upon the individual GPSS programmer's concern for precision, as well as the amount of additional effort one wishes to expend for his simulation program, four options are available:

Option 1: Continue utilizing the 25 point SNORM in the above FVARIABLE.

Option 2 : Expand SNORM to more points, to increase precision by minimizing the slight error due to linear interpolation.

Option 3 : Build a FORTRAN quadrature routine for Normal evaluation, and access it via the GPSS HELP.

Option 4 : Use Theorem 1 directly to determine the breakpoints.

Let us now return to our earlier problem of constructing a GPSS process generator for N(6,1). We first will apply Option 4 and use Theorem 1 directly to obtain the breakpoints. With the aid of 6-place Normal tables in (10), we would build our GPSS FUNCTION to provide the following:

Range	Value	Prob.
0.0 ≤ r < .000003	1	.000003
.000003 ≤ r < .000233	2	.000920
.000233 ≤ r < .006210	3	.005977
.006210 ≤ r < .066807	4	.060597
.066807 ≤ r < .308538	5	.241731
.308538 ≤ r < .691462	6	.382924
.691462 ≤ r < .933193	7	.241731
.933193 ≤ r < .993790	8	.060597
.993790 ≤ r < .999767	9	.005977
.999767 ≤ r < .999997	10	.000230
.999997 ≤ r ≤ .999999	11	.000003

We note the desired symmetry present in this GPSS discrete generator, which has its mean equal to 6. Another calculation yields the standard deviation of this distribution as 1.0408333.

We will next apply Option 1 to the same problem, i.e., we use the modified FVARIABLE (with a 1/2 correction factor we discussed) and the 25 point SNORM. We can analyze the probability distribution obtained from the following:

Range	Value	Prob.
0.0 ≤ r < .000015	1	.000015
.000015 ≤ r < .000690	2	.000675
.000690 ≤ r < .006210	3	.005520
.006210 ≤ r < .066810	4	.060600
.066810 ≤ r < .309415	5	.242605
.309415 ≤ r < .690585	6	.381170
.690585 ≤ r < .933190	7	.242605
.933190 ≤ r < .993790	8	.060600
.993790 ≤ r < .999310	9	.005520
.999310 ≤ r < .999985	10	.000675
.999985 ≤ r ≤ .999999	11	.000015

For this distribution (which is symmetric with the desired mean), the standard deviation is 1.044854, which is slightly higher than the previous one.

As our last example, let us derive a discrete process generator in GPSS to approximate the continuous probability process given by the density function

$$f(x) = \frac{x^2}{243} \qquad 0 \leq x \leq 9$$

Invoking Theorem 1, we see that we will need to evaluate F(.5), F(1.5),..., F(8.5), where the CDF F(x) is easily found to be

$$F(x) = \frac{x^3}{729} \qquad 0 \leq x \leq 9$$

The GPSS processor for this distribution is characterized in the following table:

Range	Value	Prob.
$0.0 \leq r < .0001714$	0	.0001714
$.0001714 \leq r < .0046296$	1	.0044582
$.0046296 \leq r < .0214334$	2	.0168038
$.0214334 \leq r < .0588134$	3	.0373800
$.0588134 \leq r < .1250000$	4	.0661866
$.1250000 \leq r < .2282235$	5	.1032235
$.2282235 \leq r < .3767146$	6	.1484911
$.3767146 \leq r < .5787037$	7	.2019891
$.5787037 \leq r < .8424211$	8	.2637174
$.8424211 \leq r \leq .9999999$	9	.1575789

The expected value of this discrete approximation is calculated to be 6.7638893, which compares quite well with the true expected value of the continuous distribution of 6.75 . Similarly, the standard deviation of this discrete approximation turns out to be 1.783933, which is very close to the true standard deviation of 1.7428425.

CONCLUSION

The "time-honored" probabalistic process generation methodologies in GPSS have been analyzed, with their precision in certain regards found to be lacking. The process generation methodologies suggested in this paper should prove useful to the GPSS programmer who has a concern for precision and the attainment of verisimilitude of his models to the object systems under study.

BIBLIOGRAPHY

1. Maisel,H. and Gnugnoli,G.,Simulation of Discrete Stochastic Systems, Science Research Associates, 1972, pp.218-219.
2. Fishman,G.,Concepts and Methods in Discrete Event Digital Simulation, Wiley, 1973, pp. 167-168.
3. Gordon, G., The Application of GPSS V to Discrete System Simulation, Prentice-Hall,1975.
4. Feller,W.,An Introduction to Probability Theory and Its Applications - Volume 1, Wiley, 1961, pp. 400-402.
5. Schriber,T.,Simulation Using GPSS, Wiley, 1974, pp. 165-166.
6. Scher,J., "A Least Squares Approach for The Simulation of the Poisson Process in GPSS," Modeling and Simulation, Volume 6, 1975, pp. 541-546.
7. Johnson,N. and Kotz,S., Continuous Univariate Distributions - 1, Houghton-Mifflin,1970,pp. 81-87.
8. Stone,H., "Approximation of Curves by Line Segments," Mathematics of Computation, Volume 15, 1961, pp. 40-47.
9. Hildebrand,F., Advanced Calculus for Applications, Prentice-Hall, 1964, p. 360.
10. Owen,D., Handbook of Statistical Tables, 1962, Addison-Wesley, pp. 3-13.