AN APPRECIATION OF COMPUTER SIMULATION

Ram Vaswani

IBM Corporation

ABSTRACT

Students learning computer simulation often are bewildered by the complexity of computer simulation models. To overcome this difficulty, we have used an exercise which requires modeling a simple system.

The student constructs a model and performs hand simulation. Next, the student uses a program deck to perform computer simulation for the same problem. Finally, the student constructs an analytic model and compares the results of simulation runs with calculated results.

Experience shows that this three-way exercise, involving hand simulation, computer simulation, and analytic model, gives the beginning student a feeling for and confidence in the validity of computer simulation.

INTRODUCTION

Simulation often is associated with the modeling and analysis of complex and large-scale systems. Examples of such large-scale system simulation models include:

1. The world system model which interrelates population, capital investment, geographic space, natural resources, pollution, and food production, and estimates the dynamics of change in the world. (2,5)

2. A growth company for which decisions must be made about prices, production levels, investments in plant and equipment, dividends, and the means to be used for financing growth. (1)

Generally, such simulation models require expertise in modeling and programming and the use of large-scale computers.

When a student begins to study the computer simulation technique, he feels the need to be convinced about the validity of this technique. For example, the student feels the need to see simple demonstrations that show that outputs from computer simulation runs agree with those calculated by the application of alternative analytic techniques.

In most cases for which computer simulation technique has been used to analyze a real life problem, the situation generally has been too complex to be analyzed by any alternative analytic method. For such cases, computer simulation emerges as the only method that can be used to aid policymakers in understanding and controlling these complex systems. Since such complex systems cannot be analyzed by alternative methods, there is no benchmark or standard of comparison that can be used to give a beginning student some confidence in the validity of computer simulation. For this reason, there is a need to design training exercises that an instructor can use in the classroom environment to demonstrate the similarity between computer simulation results and those calculated analytically. Such exercises should have the objective of building a student's confidence in the correctness of computer simulation outputs.

The purpose of this paper is to describe a training exercise that I have used to introduce students to the subject of computer simulation. This exercise involves the following five steps:

1. Model a simple and realistic situation.
2. Perform hand simulation.
3. Write a computer program of the model and make computer runs.
4. Derive an analytic solution.
5. Compare results derived in Steps 2, 3, and 4.

Each one of these steps will be described in sequence.

STEP 1. THE MODEL

The system that I model in the classroom is a subsystem of a computer system and has relevance to designers of computer systems. As will be discussed later on, the model used to represent this subsystem can also be used to represent systems in a few other fields, e.g. a weapon system or a salesman who calls on a prospect to secure an order. Thus, although, in this paper, I shall use the model to represent a subsystem of a computer system, the reader

should bear in mind that the model can also be adapted for training specialists in other fields.

The subsystem that I model in the class has features of an IBM 3330 direct access storage device (DASD). This DASD is attached to a control unit which is attached to a central processing unit (CPU) via a channel (Illustration 1). Since several DASD's share a common channel, the channel might not be free, when required, to access a record stored on DASD. When such a situation occurs, the record goes past the READ/WRITE head and a new attempt must be made to access the record during the next disk rotation. For this subsystem, consisting of a DASD, a shared channel, and a CPU, we want to estimate the effect of congestion in the channel on our ability to access a record on DASD.

A verbal description of the subsystem to be modeled is given in Illustration 2. This description is handed to the student. In addition to the verbal description, Illustration 2 contains a simplifying assumption and suggests a measurement for the degree of difficulty experienced in accessing a record, viz. the probability of gaining access to the record in up to N disk rotations.

Illustration 3 contains a logical flow chart for the subsystem described in Illustration 2. GPSS symbols and convention have been used in this flow chart. (3) A transaction (request for the service of the channel) is generated

and the value N (the maximum number of disk rotations permissible for reading the record) is stored in parameter No. 1 of this transaction. The TRANSFER block simulates the channel:

a) With probability P, the channel is free. When the channel is free, the record on DASD is accessed and the transaction is terminated. This corresponds to the starting of the SEARCH and READ commands; consequently, the task is classifed as a success when it enters the TERMINATE block.

b) With probability (1-P), the channel is not free. When this event occurs, the sector in which the record resides goes past the READ/WRITE head and a new attempt to read the record must be made during the next disk rotation. In GPSS flow chart convention, this operation is simulated by entering the transaction in a LOOP block. This block reduces by one the value stored in the first parameter of the transaction and returns the transaction to the TRANSFER block (see (a)).

In the exceptional case, when a transaction has passed through the TRANSFER block N times, the value in the transaction's first parameter will be reduced to zero. In this case, further attempts to find the service of the channel are not permitted and the transaction goes from LOOP to TERMINATE block. This corresponds to failure in starting the

ILLUSTRATION 1

Direct Access Storage Device (DASD) and Shared Channel
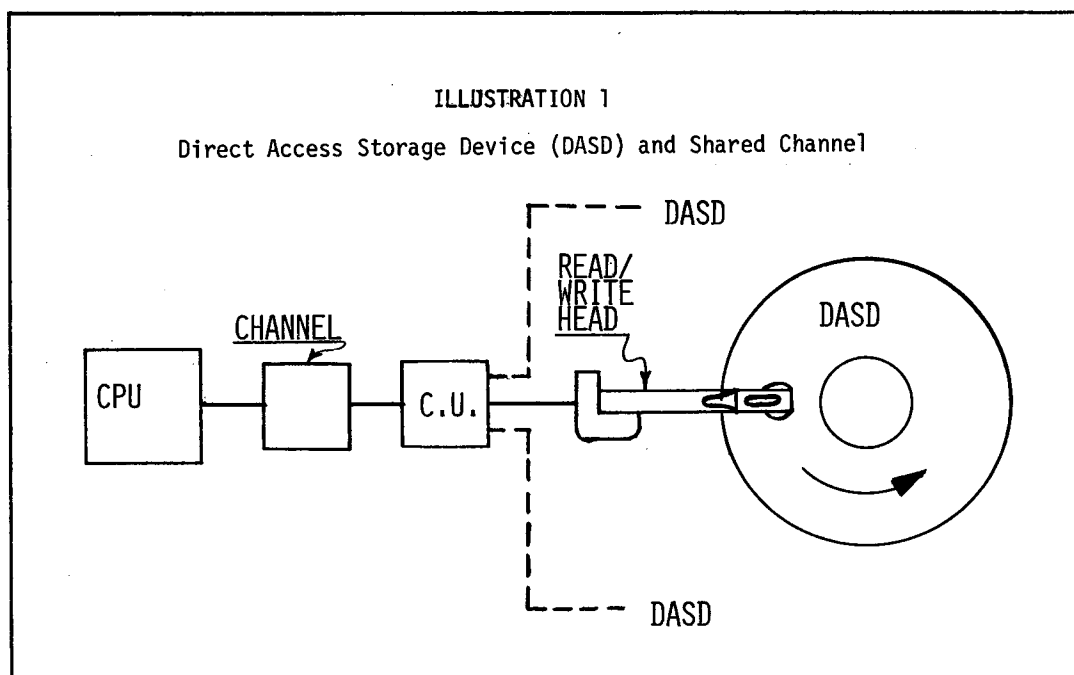
SEARCH and READ commands in N disk
rotations; consequently, the task is
classified as a failure when it enters
the TERMINATE block.

This flow chart constitutes a model of our sub-
system and provides a basis for carrying out
STEPS 2 and 3.

ILLUSTRATION 2

Verbal Description for Reading a
Record from DASD

The operation of reading a record from a DASD
requires the execution of the following four
commands:

. seek
. set sector
. search
. read

After the first two commands are completed,
the last two commands tie up the channel
throughout their duration and can be exe-
cuted only when the channel is free.

If the channel is free when required, the
search and read commands are executed im-
mediately. If the channel is not free
during the first disk rotation, attempts
are made, repetitively, after each disk ro-
tation to see if the channel is free, in
order that the search and read commands can
be started.

Make the simplifying assumption that the
probability for finding the channel free in
any disk rotation is P and that this pro-
bability is constant and independent of
the number of previous attempts to find
the service of the channel. Based on this
assumption, evaluate the probability for
starting the search and read commands in
N disk rotations.

Compare this estimate with theoretical
calculations.

STEP 2. HAND SIMULATION

Naylor, et al. recommend that prior to incurring
the expense of writing a computer program, hand
calcualtions be performed so as to evaluate a
simulation model and its inputs. (6) The
reasoning behind this recommendation is that if
results of hand calculations compare favorably
with historical or actual results, then this
comparison will give us a confidence in the
model and the inputs. While such hand calcula-
tions are onerous and expensive for a large-
scale simulation, this training exercise was
designed with an objective of ease of hand
calculations.

In the classroom, I assign specific value to:

P = the probability of the channel being
free,

N = the maximum number of disk rotations
allowed to access a record.

Since, at this stage, the student has not been
exposed to the use of random numbers, I select
a convenient value for P(=0.5) and use a coin
to generate random outcomes. Also, to make
recordkeeping reasonable, I set N=2. Illustra-
tion 4 shows an appropriate worksheet for hand
simulation for P=0.5 and N=2 on which outcomes,
based on tosses of a coin, for 20 transactions
have been recorded. On the basis of this set
of experiments, the estimate for the probability
of success in accessing a record in up to two
disk rotations is 0.8.

This hand simulation leads students to try other
values for P and N, e.g. P=0.25 (which represents
a busier channel than when P=0.5) and

ILLUSTRATION 3

Flow Chart for Starting SEARCH and
READ Commands
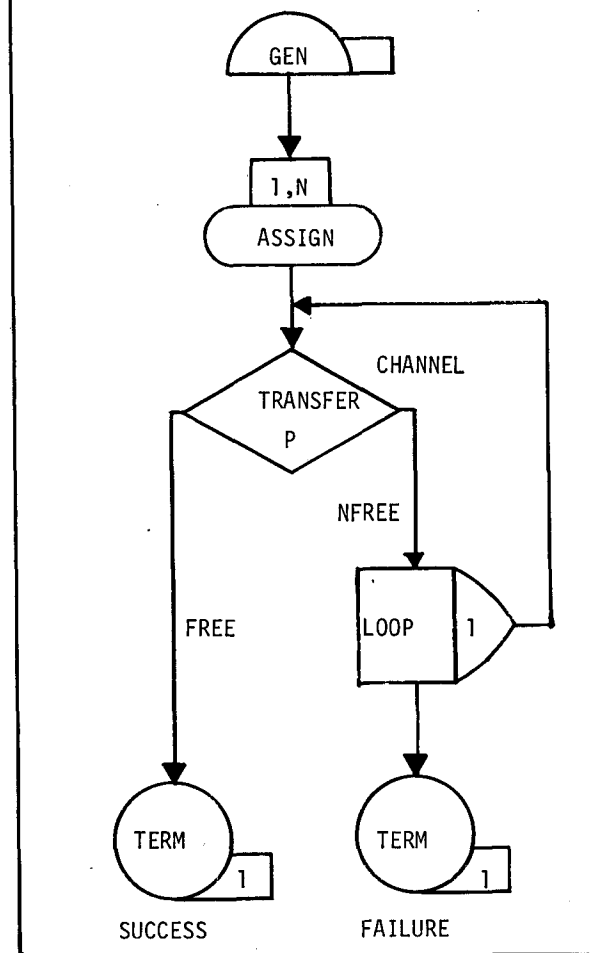
N=3, 4, ... At this point, the student is introduced to random number tables. (7)

At the end of Step 2, the student:

. begins to gain some familiarity with a simulation model.

. understands the use of random numbers.

. recongnizes the stochastic nature of results and realizes the need for increasing the precision of results.

. appreciates the problem in recordkeeping when large value are assigned to N and the number of transactions (n) put through the simulator is increased.

This appreciation naturally leads to Step 3.

ILLUSTRATION 4

Worksheet for Hand Simulation

P = 0.5, N = 2

Expt. No.	First Toss	Second Toss	Result 1 = Success 0 = Failure
1	H		1
2	H		1
3	T	H	1
4	H		1
5	T	T	0
6	H		1
7	T	H	1
8	T	H	1
9	T	T	0
10	H		1
11	H		1
12	T	H	1
13	H		1
14	H		1
15	H		1
16	H		1
17	H		1
18	T	T	0
19	T	T	0
20	H		1

Total   16

Estimate for Probability of Success in Two Rotations = 16/20 = 0.8

STEP 3. COMPUTER SIMULATION

For small number of transactions (n), this problem can easily be simulated by hand. However, estimates for probability of starting SEARCH and READ commands will vary considerably from run to run when n is small.

A computer program can be written for the DASD probem using the flow chart given Illustration 3. In some courses, it might be advisable to ask the student to write such a computer program, either in the classroom or as a part of homework. This programming exercise will give the student a flavor for using a simulation language, such as GPSS V, to write and debug a program. On the other hand, when the teaching objective does not include exposing the student to the use of a simulation language for writing programs, a program deck can be kept ready for class use. A listing of such a program, using GPSS V, is given in the Appendix.

The student can use the program deck to simulate, rapidly, this computer subsystem with many different values for P and N. In addition, the student can repeat the flow of transactions (n) through the subsystem many times and thus improve the precision of his estimates. For example, one student, using the program deck listed in the Appendix, made five computer runs, each run consisting of n=1000 transactions. Inputs and outputs for these runs are summarized in columns 1, 2, and 3 of Table 1.

STEP 4. ANALYTIC SOLUTION

So far, the student has been shown (i) a model of a subsystem consisting of a DASD and a shared channel and (ii) a hand simulation for this subsystem. Further, he has experimented with the use of a computer program to simulate the subsystem. However, the student still does not have a basis for judging if results of hand or computer simulation are valid. The purpose of Step 4 is to provide a benchmark against which the student can compare the results of hand or computer simulation.

In Step 4, the student is introduced to the concept of the sample space and the probability associated with each sample outcome. Using this concept, we derive an analytic expression for:

f = the probability for failure in accessing a record on DASD in N consecutive disk rotations,

$$f = (1-P)^N.$$

Numerical values for f can be obtained from binomial probability tables. Define:

p = the probability of success in accessing a record in up to N disk rotations.

Then, p is given by:

$$P = 1-f = 1-(1-P)^N.$$

STEP 5. COMPARISON

In this step, the student compares estimates from computer simulation (Step 3) with analytic results (Step 4). Since estimates from computer simulation are subject to statistical variation, these estimates will differ from theoretical calcualtions. However, the difference between analytic and simulation results should be within reasonable limits.

At this stage, the student is introduced to the concept of confidence limits. If the probability for success estimated from computer runs falls within the confidence limits, then we have some assurance that the computer simulation model does seem to simulate the physical process of reading a record from DASD in up to N disk rotations.

What are reasonable confidence limits and how are these to be calculated? Since the process of reading a record in up to N disk rotations itself is a binomial process with probability of success:

$$p = 1-(1-P)^N,$$

this process has a standard deviation:

$$\sigma_p = \sqrt{p(1-p)/n},$$

where:

  n = the number of terminations or transactions.

Normally, three-sigma confidence limits are considered as reasonable and if this convention is accepted, confidence limits within which estimates from computer simulation should lie are given by $p \pm 3\sigma_p$.

These confidence limits are shown in columns 6 and 7 of Table 1. It is seen that for each of five runs given in Table 1, the estimate from computer simulation (p_n) falls within the calculated confidence limits.

CONCLUDING REMARKS

Although the main objective of this exercise is to help the student gain confidence in the validity of the computer simulation technique, the exercise provides an opportunity to discuss many other important concepts about modeling and simulation. Five such concepts are mentioned below:

1. The topic of generation of input data often is raised in the class. One of the parameters used in this exercise is P, the probability of channel availability and students often inquire the source from which the value for P can be obtained. Actually, the value for P cannot be obtained from any source, since this value depends upon the system and traffic parameters. In any specific case, it might be necessary to construct another simulation model designed to estimate the value for P. For example, Martin has specified a simulation model for a computing system consisting of three DASD's one channel, and one CPU. (4) Using this simulation model, Martin has estimated

TABLE 1

Comparison between Simulation Estimates and Analytic Results (n=1000 terminations)

			Probability for Success in up to N Rotations		$\sigma_p = \sqrt{p(1-p)/n}$	Confidence Limits for p_n	
			Simulation (p_n)	Analytic (p)		Upper = $p + 3\sigma_p$	Lower = $p - 3\sigma_p$
	N	P					
Column No.	1	2	3	4	5	6	7
	3	0.5	0.890	0.8750	0.0105	0.9065	0.8435
	2	0.6	0.849	0.8400	0.0116	0.8748	0.8052
	2	0.4	0.624	0.6400	0.0152	0.6856	0.5944
	4	0.6	0.977	0.9744	0.0050	0.9894	0.9594
	4	0.4	0.852	0.8704	0.0106	0.9022	0.8386

the channel utilization rate corresponding to many traffic densities between 10 and 17.5 transactions per second.

2. In the worksheet in Illustration 4, a head is classified as a success and a tail as a failure. However, one could reverse this classification and classify a head as a failure and a tail as a success. This leads to a discussion of antithetic simulation and of the additional insight to be gained by running both a straight and its antithetic simulation. (8)

3. For computer system designers, the output from this exercise suggests additional investigations, e.g. exploring the tradeoff between P and N, i.e. the cost for increasing channel availability (e.g. by increasing the number of channels, or using a high speed channel, or reducing the traffic density, etc.) versus the penalty for increasing the delay in starting the SEARCH and READ commands.

4. Models have some degree of generality, i.e. a model might have applications in many diverse fields. For example, the application of the model used in this exercise is for training computer system designers. With some changes of wording, the same model can be used to represent:

    . A weapon system that can fire up to N rounds, with the kill probability for each round being a constant P. In this case, the model evaluates the total kill probability (p) for the weapon system. (9)

    . A salesman who calls upon prospects to secure orders for a vendor, with probability for securing an order during each sales call being a constant P. In this case, the model estimates the total probability (p) for securing an order in up to N sales calls. Since the selling cost incurred by the vendor includes in it an item which is a function of the number of sales calls that his salesman makes on the prospect, the vendor might want to set an upper limit on the number of calls (N) that his salesman is allowed to make.

    Factors that influence the value of P could be discussed, e.g.:

    (i) Sales training, experience, and rapport with customer.

    (ii) Sales support, including product leadership, image building, and customer education and service.

In each of these two cases, there might be a tradeoff between P and N that could be investigated.

5. The vendor's example given in the preceding paragraph provides a basis for discussion of the appropriateness of assumptions used in the model. For example:

    . What characteristics should the product possess to make this an appropriate model for the salesman?

    . What kind of prospect is assumed in the model so that the condition of constant P will hold true?

    If assumptions that underlie this model do not fit the case of the vendor, then conclusions derived from the model might be misleading. This property emphasizes the point that the model user must play an essential role during the model specification phase. The model user should satisfy himself that the analyst has incorporated the most appropriate assumptions in the model.

This training exercise has deliberately been simplified so that the student can follow the various steps involved in computer simulation and can understand the derivation of an analytic solution. At the same time, the exercise provides an opportunity to raise, discuss, and illustrate many concepts about modeling and computer simulation.

ACKNOWLEDGEMENT

BIBLIOGRAPHY

1. Burrill, C. W. and Quinto, L. Computer Model of a Growth Company. Gordon and Breach Science Publishers, New York, 1972.
2. Forrester, J. W. World Dynamics. Wright-Allen Press, Inc., Cambridge, Mass., 1971.
3. IBM General Purpose Simulation System V, User's Manual, Form SH20-0851, IBM Corporation, Data Processing Division, White Plains, N.Y.
4. Martin, J. Design of Real-Time Computer Systems. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1967, pp. 353 - 368.
5. Meadows, D. H., Meadows, D. L., Randers, J., and Behrens, W. W. III. The Limits of Growth. Potomac Associates, Washington, D.C., 1972.
6. Naylor, T.H., Balintfy, J. L., Burdick, D. S., and Chu, K. Computer

Simulation Techniques. John Wiley and Sons, Inc., New York, 1966, p. 37.

7. RAND Corporation. A Million Random Digits with 100,000 Normal Deviates. The Free Press, Glencoe, Illinois, 1955.

Many textbooks include small tables of uni-formly distributed or standardized normally distributed random numbers. For example, see:
Shook, R. C. and Highland, H. J. Probability Models with Business Applications. Richard D. Irwin, Inc., Homewood, Illinois, 1969, pp. 481 - 485.

8. Rechtschaffen, R. N. "Queueing Simulation Using a Random Number Generator," IBM Systems Journal, Vol. 11, No. 3, 1972, pp. 255 - 271.

9. Zimmerman, R. E. "A Monte Carlo Model for Military Analysis," in Operations

Research for Management, Vol. II, J. F. McCloskey and J. M. Coppinger (eds.), The Johns Hopkins Press, Baltimore, Md., 1956, pp. 376 - 400, esp. p. 380.

APPENDIX

GPSS Program Listing

```
BLOCK                                                                    STATEMENT
NUMBER  *LOC    OPERATION  A,B,C,D,E,F,G,H,I        COMMENTS             NUMBER
        *       *****************************************************      1
        *       *                                                 *        2
        *       *              EXERCISE NO. 1                      *        3
        *       *                                                 *        4
        *       * READING A RECORD FROM DASD IN UPTO N DISK ROTATIONS *    5
        *       *                                                 *        6
        *       *****************************************************      7
        *                                                                  8
                SIMULATE   5                    MAX COMPUTER RUN TIME       9
        *                                                                 10
        *       SAVEVALUE INITIALIZATION                                  11
        *                                                                 12
                INITIAL    X$NMAX,3             LIMIT ON NUMBER OF ATTEMPTS 13
                INITIAL    X$PCHAN,500          PROB. OF FINDING FREE CHANNEL 14
        *                                                                 15
1               GENERATE                        CREATE REQUEST FOR CHANEL USE 16
2               ASSIGN     1,X$NMAX             WRITE NMAX IN PARAMETER NO. 1  17
3       CHANL   TRANSFER   .X$PCHAN,NFREE,FREE  IS CHANNEL FREE?            18
4       FREE    TERMINATE  1                    CHANNEL IS FREE.  TERMINATE.  19
5       NFREE   LOOP       1,CHANL              CHANNEL NOT FREE.  TRY AGAIN. 20
6               TERMINATE  1                    FAILED TO GET CHANNEL IN NMAX  21
                START      1000                 REPEAT EXPERIMENT 1000 TIMES   22
        *                                                                 23
                CLEAR                           CLEAR STATISTICS AND XACTION   24
        *                                                                 25
                INITIAL    X$NMAX,2                                        26
                INITIAL    X$PCHAN,600                                     27
        *                                                                 28
                START      1000                 REPEAT EXPERIMENT 1000 TIMES   29
        *                                                                 30
                CLEAR                                                     31
        *                                                                 32
                INITAL     X$NMAX,2             LIMIT ON NUMBER OF ATTEMPTS 33
                INITIAL    X$PCHAN,400                                     34
        *                                                                 35
                START      1000                 REPEAT EXPERIMENT 1000 TIMES   36
        *                                                                 37
                CLEAR                                                     38
        *                                                                 39
                INITIAL    X$NMAX,4                                        40
                INITIAL    X$PCHAN,600                                     41
        *                                                                 42
                START      1000                 REPEAT EXPERIMENT 1000 TIMES   43
        *                                                                 44
                CLEAR                                                     45
        *                                                                 46
                INITIAL    X$NMAX,4                                        47
                INITIAL    X$PCHAN,400                                     48
        *                                                                 49
                START      1000                 REPEAT EXPERIMENT 1000 TIMES   50
        *                                                                 51
                END                                                       52
```

RAM VASWANI, Senior Instructor, IBM Systems
Science Institute, has taught computer simulation
to students attending the Data Base Architecture
course at the Institute. His other interests in-
clude modeling and simulation of business systems,
financial management, and planning. He has taught
at the American University, the City University
of New York, and the North Carolina State
University. He received a Doctor of Engineering
Science degree in Industrial and Management Engi-
neering from Columbia University in 1958.