Experiential Learning Enters the Eighties, Volume 7, 1980

# COMPUTER SIMULATION: A TOOL TO TEACH QUEUING THEORY

John H. Reed, Clarion State College


## ABSTRACT

This paper presents computer simulation as a technique for teaching queuing theory. Discussion is limited to the single-server, first-in first-out, poisson arrivals, exponential service time queuing model. Four phases are used in the presentation, the definition phase, the flowcharting phase, the derivation and programming phase, and the reporting phase. The first three phases teach students the basic concepts of queuing theory and permit them to examine what is occurring inside the queuing model. The last phase, the reporting phase, is used to reinforce the concepts of queuing theory and to illustrate the differences and similarities between analytical solutions and computer simulation solutions to queuing models.


## BACKGROUND

The School of Business Administration at Clarion State College offers two courses in operations research. One of these courses, Operations Research II, covers the stochastic processes of queuing theory and inventory control A problem arises in that many of the students which are enrolled in the business program have a weak background in mathematics and probability theory. This deficiency makes it very difficult to present to the students even elementary stochastic aspects of inventory control and queuing theory As a result, an inordinate amount of time is spent on mathematics and probability theory. In fact, more time is spent teaching mathematics than teaching the central topics, queuing theory and inventory control.

## ROLE OF COMPUTER SIMULATION

The answer to this dilemma is “computer simulation.” The use of computer simulation has a number of valuable aspects:

I. Simulation can be used as a pedagogical device, in such disciplines as business administration and economics, for teaching students basic skills in theoretical analysis, and decision-making.

2. Simulation requires explicit definition and identification of variables which helps to provide the student with insight into the system.

3. Simulation permits one to study a system in a real or compressed time period. Thus one can study and experiment with the interactions of a system and thereby gain a better understanding of the system.

4. The experience one gains in designing a simulation may be more valuable than the simulation itself. The construction of a simulation requires that the designer have a thorough understanding of the *system.* This understanding can reveal subtle relationships which have important bearings on the interaction of variables. Without such revelations, the system may not be understood. [1]

## APPLICATION OF COMPUTER SIMULATION

It is the objective of this paper to illustrate how simulation can be used and integrated with the more traditional approaches to mathematics to teach students one such topic, queuing theory.

A characteristic of queuing theory is that its mathematical equations are frequently very lengthy and complex. By using simulation, one cannot necessarily make the derivations of queuing theory problems more understandable. However, simulation can be used to provide the student with a better and clearer understanding of the starting assumptions and the meaning of the end result equations. This paper illustrates the use of computer simulation in teaching the single- server, first-in first-out, poisson arrivals, exponential service time, queuing model, henceforth called the single-server queue.

While the derivations of the single-server queue are lengthy and complex, the final equations are very simple and straight forward. These equations are so simple, that it becomes difficult for the student to connect them with the original equations and assumptions. To make this connection, the students are introduced to a fixed time incremented simulation which describes this queuing problem. The topic, the single-server queue, is presented in four phases, the definition phase, the flowcharting phase, the derivation and programming phase, and the reporting phase. During the first phase, variables are identified, terms and variables defined, rules and assumptions under which the queuing system operates are presented, and a real world example of the queuing model is presented. At this time, discussion of mathematical expressions and equations are kept to a minimum.

During the second phase students are presented with a flow diagram for a fixed time incremented simulation which adheres to all of the assumptions and rules tinder which the queuing model operates. The flow diagram is short, concise and appears in Appendix I. Since it operates in fixed increments of time, it is not too difficult for the students to comprehend.

The third phase Involves derivations of the equations describing the single-server queue. These equations are derived using the birth-death process and can be found in many operations research texts such as Introduction to Operations Research by Frederick S. Hillier and Gerald J. Lieberman or Fundamentals of Operations Research for Management by Shiv K. Gupta and John M. Cozzolina. Since this process is lengthy and complex, not all of the students are able to follow the presentation. While the equations are being derived in class, the students are given a computer programming assignment. They are to program on a computer time-sharing system the problem presented in the flow diagram of phase two. At Clarion, BASIC is used as the programming language, however, FORTRAN or ALGOL would also be appropriate computer languages. A BASIC program for the fixed time Increment simulation is given in Appendix II.

The flowchart gives the student a picture of how the queuing system operates. However, the flowchart does not present the student with a one-to-one relationship with the programming language. The student must review and study the logic of the chart several times while translating it into a computer program. This forces the students to review and redefine the variables of the system and to explicitly describe to the computer how these variables interact. After completing this process, most of the students will have an understanding of the system.

In programming the flowchart, with two exceptions, the students are responsible for creating their own programs. The two exceptions occur in the generation of arrivals and the generation of service times. The students are given the BASIC routines for generation of poisson arrivals and exponential service times. They only have to insert these routines into the proper location of the program. When it comes time for the students to test and debug their programs, they are encouraged to insert, at the end of each iteration, instructions which have the computer print out values for the system variables. Thus during program execution, the output of these variables will not only help students in debugging the program but will also assist them is getting a visual picture of just what is happening as the queuing system operates. The students can observe for themselves arrivals, buildup of queues, variation of waiting times and variation of idle times. Furthermore, they can observe what happens to these same variables as changes are made to arrival rates or service times.

Upon completion of the derivation and programming phase, the reporting phase begins. Each student is assigned a unique set of arrival rates and service times. The students are then instructed to use these two parameters to solve the problem analytically and through computer simulation. The results of this effort are then written and reported in the following format:

1. Description of the problem.
2. Copy of computer program.
3. Simulation solution.
4. Analytical solution.
5. Differences and similarities between the two solutions. Why do these differences and similarities exist and what is the effect on them as the number of iterations is changed?

The analytical solution of the single-server queue Is very easy to calculate so the papers are quite easy to grade, even though each student has been assigned a unique set of parameters with which to work. The assignment of different parameters to each student helps to insure that each student does his own work. A sample problem is given with its analytical and simulation solutions in Appendix III.

When the students have completed their assignments they are informed that their computer simulation model not only works for poisson arrivals and exponential service times but also for any other probability distributions for which they may care to substitute in the program. Consequently, students, upon completion of the assignment, not only have studied poisson arrivals and exponential service times but with minimal effort can learn about queuing systems which involve other distributions.

## CONCLUSIONS

With the submission of the final paper a number of objectives have been accomplished:

1. Students have a better comprehension of a queuing system.
2. Students are forced to define terms very explicitly.
3. The use of the computer forces the students to describe their problem very clearly and to use precise logic in attaining its solution.
4. The written paper provides them with practice in analyzing and comparing different solutions to the same problem. In the real world of business or government, the ability to submit such reports can mean the difference between success and failure.

The first endeavors with this approach have been so successful that the same approach has been done with inventory control.

## REFERENCES

[1] Naylor, Thomas H., Joseph L. Balintfy, Donald S. Burdick and Kong Chu, Computer Simulation Techniques (New York: John Wiley & Sons, 1966) pp. 8-9.

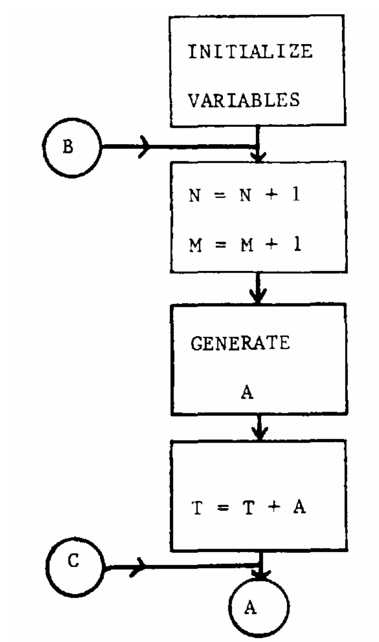
## APPENDIX I

### FLOWCHART FOR FIXED TIME INCREMENT QUEUING SIMULATION MODEL

Symbols

A = Time between i and i+1 arrivals.
C = Clock time.
I1 = Total idle time.
L = Time duration of simulation.
N = Total number of arrivals.
N = Number of units waiting to be served.
S = Service time for i th arrival.
T Arrival time of the i th unit.
W1 = Total waiting time for all units.

Flowchart



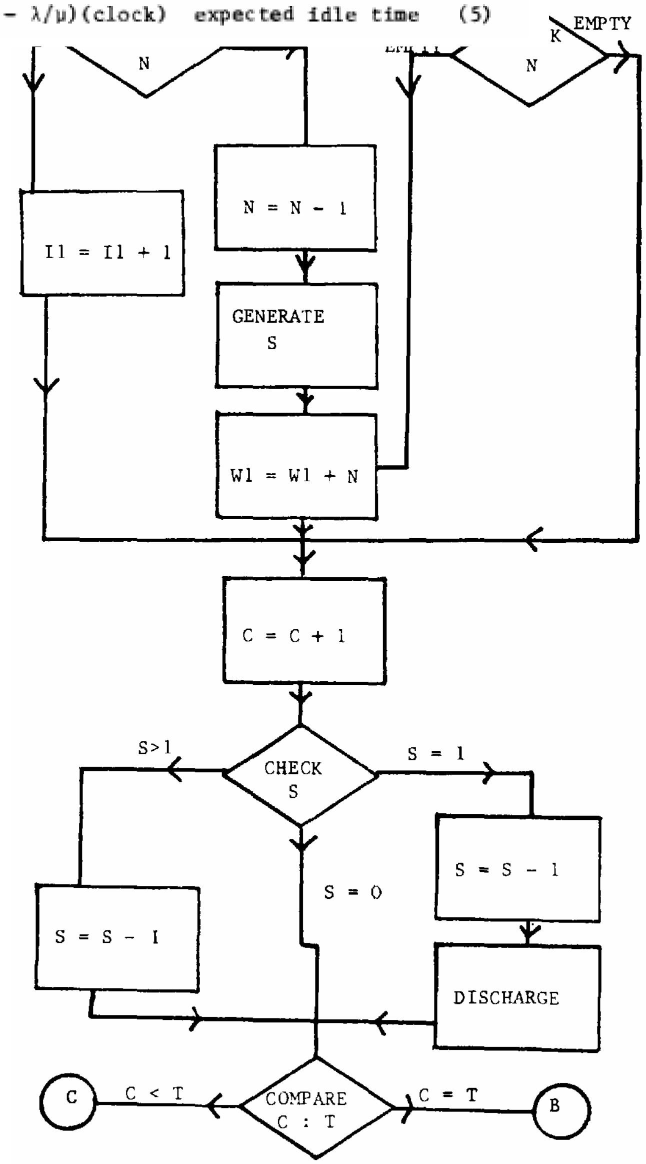
$L = \lambda/(\mu - \lambda)$ expected number of units in the system (queue and service station). (1)

$W = 1/(\mu - \lambda)$ expected waiting time in queue and service station. (2)

$L_q = \lambda^2/(\mu - \lambda)$ expected number of units in queue only. (3)

$W_q = \lambda/\mu(\mu - \lambda)$ expected waiting time of a unit in the queue. (4)

$I = (1 - \lambda/\mu)(\text{clock})$ expected idle time (5)




APPENDIX II

BASIC COMPUTER PROGRAM FOR FIXED TIME INCREMENT QUEUING SIMULATION MODEL

```
1 REM THIS PROBLEM IS A FIXED TIME
2 REM SIMULATION INCREMENTED IN TENTHS
3 REM OF A MINUTE
4 REM A = TIME BETWEEN I AND I+1 ARRIVALS
5 REM C = CLOCK TIME
6 REM I1 = TOTAL IDLE TIME
8 REM L = TIME DURATION OF SIMULATION
9 REM M = TOTAL NUMBER OF ARRIVALS
10 REM N = # OF UNITS WAITING TO BE SERVED
11 REM S = SERVICE TIME FOR I TH ARRIVAL
12 REM T = ARRIVAL TIME OF THE I TH UNIT
14 REM W1 = TOTAL WAITING TIME (ALL UNITS)
15 REM X = AVERAGE TIME BETWEEN ARRIVALS
16 REM Y = AVERAGE TIME BETWEEN DEPARTURES
20 PRINT 'TIME YOU DESIRE PROBLEM TO RUN'
22 INPUT L
24 PRINT 'TIME BETWEEN ARRIVALS'
26 INPUT X
28 PRINT 'TIME BETWEEN DEPARTURES'
30 INPUT Y
32 M=0
36 L=L*10
40 S=0
50 T=0
55 H=0
60 C=0
70 W1=0
80 I1=0
90 N=0
200 N=N+1
210 M=M+1
300 R=RND(-2)
310 A=-X*LOG(R)
320 A=A*10
330 A=INT(A+.5)
335 IF A=0 THEN 300
400 T=T+A
500 IF S>0 THEN 1000
600 IF N=0 THEN 800
700 N=N-1
710 R=RND(-3)
720 S=-Y*LOG(R)
730 S=S*10
740 S=INT(S+.5)
745 IF S=0 THEN 710
750 W1=W1+N
760 GOTO 2000
800 I1=I1+1
810 GOTO 2000
1000 IF N=0 THEN 2000
1100 W1=W1+N
2000 IF C=L THEN 4000
2100 C=C+1
2200 H=H+N
2300 IF S=0 THEN 2500
2400 S=S-1
2500 IF C=T THEN 200
2600 IF C<T THEN 500
4000 PRINT 'AVERAGE # IN QUEUE',H/L
4100 PRINT 'AVERAGE WAITING TIME',.1*(W1/M)
4200 PRINT 'TOTAL IDLE TIME',I1
4300 PRINT 'TOTAL ARRIVALS',M
5000 END
```

APPENDIX III

This appendix presents sample calculations using the analytical equations for the single-server queue and the results of two computer runs using the simulation program.

The analytical equations can be found in any operations research text. These equations are functions of the average arrival rate (A) and the average departure rate (p) . Most of the information required from the system can be calculated from the following five equations:

$$L = (1/2)/(1 - 1/2) = 1 \tag{6}$$

$$W = 1/(1 - 1/2) = 2 \tag{7}$$

$$L_q = (1/2)^2/\ (1 - 1/2) = 1/2 \tag{8}$$

$$W_q = (1/2)/(\ 1\ )(1 - 1/2) = 1 \tag{9}$$

$$I = (1 - (1/2)/1)(10000) = 5000 \tag{10}$$

For a queuing system with an average time between arrivals of two minutes and an average time between departures of one minute then A = i/2 and u = 1. By substituting these figures into equations one through five the results are as follows:

$$L = (1/2)/(1 - 1/2) = 1 \tag{6}$$

$$W = 1/(1 - 1/2) = 2 \tag{7}$$

$$L_q = (1/2)^2/\ (1 - 1/2) = 1/2 \tag{8}$$

$$W_q = (1/2)/(\ 1\ )(1 - 1/2) = 1 \tag{9}$$

$$I = (1 - (1/2)/1)(10000) = 5000 \tag{10}$$

Using the same arrival and departure rates for the computer program over a simulated 10,000 minute period gives the following results:

```
 RUN
TIME YOU DESIRE PROBLEM TO RUN
?10000
TIME BETWEEN ARRIVALS
?2
TIME BETWEEN DEPARTURES
?1
AVERAGE # IN [illegible]                .48386
AVERAGE W[illegible]                    .958708
TOTAL IDL[illegible]                    49344
TOTAL A[illegible]
*
```

Note that since the results of the computer program are dependent upon two random number generators the answer will vary each time the program is run. Furthermore if the time frame over which the program is run is shortened, then variation between run results will increase.

If a student desires to take a closer look at what is occurring in the queuing system then by changing a couple of statements in the program the printout will yield the system status from minute to minute. The following printout gives the status of the system during the first ten minutes of operation. For this computer run, the time between arrivals and departures was changed to one and two minutes respectively.

```
 RUN
TIME YOU DESIRE PROBLEM TO RUN
?10
TIME BETWEEN ARRIVALS
?1
TIME BETWEEN DEPARTURES
?2
                                                TIME
TIME   #IN QUE   ARR TIME   DEP TIME   WAITING
  0       0          2          4          0
  1       0          2          3          0
  2       1          3          2          1
  3       2          4          1          3
  4       2          5          1          5
  5       2          7          2          7
  6       2          7          1          9
  7       2          8          1         11
  8       2          9          8         13
  9       3         10          7         16

AVERAGE # IN QUEUE                1.6
AVERAGE WAITING TIME              2.22222
TOTAL IDLE TIME                   0
TOTAL ARRIVALS  9
*
```