The Single Server Queue in Discrete Time-
Numerical Analysis I

by

Marcel F. Neuts (*)
Purdue University

Department of Statistics
Division of Mathematical Sciences
Mimeograph Series No. 270
November 1971

(*) This research was supported by the National Science Foundation - grant GP 28650.

Abstract

This is the first of a sequence of papers dealing with the computational aspects of the transient behavior of queues in discrete time.

It is shown that for a substantial class of queues of practical interest, a wealth of numerical information may be obtained by relatively unsophisticated methods.

This approach should prove useful in the analysis of unstable queues which operate over a limited time interval, but is by no means limited to such queues.

Mathematically the service unit is modeled in terms of a multivariate Markov chain, whose particular structure is used in iterative computation. Many important queue features may then be derived from the n-step transition probabilities of this chain.

1. Introduction

The theory of queues, and more generally that of stochastic models, suffer from the insufficient development of the interface between structural-analytic results on one hand and directly applicable numerical methods on the other hand.

The practical queue analyst tends to use of the extensive theoretical work on service systems only those rare steady-state results, which are analytically simple. This is often done with little regard for the mathematical assumptions underlying these results. Moreover such results usually do not answer the real questions one is facing in the design or the improvement of a service facility.

While simulation techniques are widely used, their implementation requires a thorough understanding of the probability structure of the queue as well as very substantial computing funds. In most instances of simulation studies familiar to this author, the structure of the queue was incorrectly or insufficiently used and exorbitant computing times were reported. One recognizes that simulation is often the only resort in studying a complex system; however, for those models whose structure is mathematically well understood, it is desirable that algorithms making use of the existing theory be developed.

Before we consider one such model in detail, let us discuss some of the difficulties and the desiderata related to such queue algorithms.

a. The finiteness of the waitingline

In reality, unbounded queues do not exist. The unbounded queue is strictly a mathematically convenient abstraction. By removing one "boundary" of the queuelength process one obtains simpler stochastic processes. It further becomes possible to give an elegant treatment of

the intuitive quality of stability of the queue.

In practical situations, there is either a finite waitingroom, usually with loss of those customers who find the waiting space full; or else the buildup of the queue beyond a certain limit creates utter chaos and the object of the study is precisely the design of a sufficiently fast service unit or a sufficiently large waitingroom to make this a very rare event.

Throughout this paper we shall limit our attention to bounded queues. In the numerical examples considered the largest value of the maximum queue length L_1 was at most one hundred.

b. Transient versus Steady-state behavior

There are very few results on the transient behavior of queues, which are analytically explicit. Even the latter are nearly all ill-suited for direct numerical analysis. An interesting discussion of a simple transient queue and the difficulties of its computational analysis may be found in E. L. Leese and D. W. Boyd [1].

The steady-state probabilities of some simple queues are the only ones available in books written for the applied worker. Their relevance to the concrete problems is often limited; they clearly have no bearing whatsoever on the solution of unstable queues. Moreover, since the limiting process by which they are obtained has an averaging (or mixing) property, such results convey no information on the fluctuations of the queuelength and the waiting times. Ignoring such fluctuations in a design may have catastrophic results.

In recent years the study of weak convergence properties of queueing processes and the resulting diffusion approximations have

shed a new light on service systems of which the *macroscopic time-behavior* rather than the short-range fluctuations is the most important feature. This promising approach is mathematically fairly sophisticated and has not yet received sufficient investigation from the viewpoint of computation.

Our concern in this paper is in a sense with the small-scale service system whose short range behavior is important. Therefore the models studied here are unlikely to be well approximated by a diffusion process. Nevertheless a comparison of both the direct solution and a diffusion approximation method with regards to accuracy and computing time is of interest, but will not be undertaken here.

c. Continuous versus discrete parameter models

It is known that finite $M|G|1$ and $GI|M|1$ queues may be conveniently studied in relation to an imbedded Markov renewal process with a finite number of states. The transient behavior of such queues, as well of many related ones, may be computed in principle in terms of the successive matrix-convolution products of the transition matrix of this imbedded process. If a queue with a waitingroom of size L_1 is investigated, each such a matrix-convolution product may require as many as $(L_1 + 1)^2$ evaluations of the convolution product of two functions. To perform this operation accurately is time-consuming, so that the computer implementation of this analysis is likely to result in considerable computing time.

In an earlier study of the single server queue *in discrete time*, S. Dafermos and the author [2] argued at length for the advantages of analyzing many queues in terms of a discrete time parameter. These

arguments will not be repeated here. From the viewpoint of numerical analysis the most obvious advantages of a discrete time model are:

a. The ease with which one or more supplementary variables may be introduced so as to imbed the queueing process in a multivariate Markov process.

b. The fact that convolution products of sequences of numbers may be computed with much greater ease than those of functions of a real variable may be evaluated accurately.

In most cases of practical interest one may discern an elementary unit of time natural to the particular queue. Many queueing analysts nevertheless insists on thinking of discrete models as approximations to continuous ones. This insistence which may usually be traced to the prevailing attitude in applied mathematics before the advent of the computer, has some appeal for its mathematical elegance, particularly where methods of analysis may be used. From a computational viewpoint, continuous parameter models are often substantially more delicate to analyze and this without yielding additional insight into the real process which is being modeled.

d. Parametric versus general distributions

The insistence on specific parametric families of probability distributions (such as the gamma family) in stochastic models is also largely a holdover from the pre-computer era. Where a parametric assumption has an important structural consequence (e.g. the memory-less property of the negative exponential distribution) one should be very aware of this. However in cases where the parametric assumption yields only marginal simplifications, both the theoretical analysis and the computational methods should ignore it altogether. As a case in point,

the $M|E_k|1$ queue for $k > 1$ is only in some details easier to discuss than the $M|G|1$ queue. There is therefore little point in a special numerical method for the former which does not also include the latter.

In the sequel of this discussion, we shall therefore only stress *the structural assumptions* that are needed. Such items as service time distributions will be as general as possible.

In most specific models one may assume without loss of generality that service times and related random variables are bounded lattice random variables. This assumption, essential to our approach, is also the one which limits its range of applicability most. Our approach is not well suited to queues in which both very short *and* very long jobs may arrive. Different methods of numerical analysis are needed for these. It is commonly so that a given numerical method is well suited for a certain range of problems, but fails for similar problems outside this range. The assumptions which limit the applicability of the present model are discussed in the appropriate places in the sequel.

2. The Assumptions of the Model

a. The arrivals

We consider a single server queue in discrete time with a maximum queuelength L_1 (*). The elementary time interval is chosen as our time unit. We assume that the numbers of arrivals during the successive unit time intervals are independent, identically distributed random variables. Furthermore $p_\nu, \nu = 0, 1, \ldots, K$ is the probability that ν customers join the queue during a given unit time interval. $(p_0 + p_1 + \ldots + p_k = 1)$. In this discussion and in the FORTRAN program we shall insist that $1 \leq K < L_1$. The restriction $K < L_1$ is not essential and is usually satisfied in practice. Its removal requires minor modifications in the analysis and in the program.

In this discussion we assume that the p_ν are independent of time, but with minor obvious changes the recurrence relations are valid also for queues in which the arrival probabilities vary with time.

b. The service times

We assume that the service times of the successive customers are independent, identically distributed, (integer valued) random variables with values in the set $\{1, 2, \ldots, L_2\}$. We denote by r_ν, $\nu = 1, \ldots, L_2$, the probability that a customer requires ν units of service time. The values $L_1 = 100$ and $L_2 = 100$ appear to be practical upper limits to the computer implementation of the method suggested here.

Our assumption that every customer requires at least one unit of service time may easily be removed, but is satisfied in most all concrete

(*) For easy reference, we are using similar, though not identical, notation in the theoretical discussion and in our FORTRAN program.

applications. It suffices to introduce a quantity r_o that a service time is equal to zero and to modify the recurrence relations accordingly.

As pointed out below, it is also easy to modify our analysis to include the case where the service time density *depends on the time at which the service is initiated*.

c. The queue discipline

Except in discussing the waitingtimes, the order of service is immaterial. The waitingtimes will be discussed for the first-come, first-served discipline.

To settle the issue of simultaneous arrivals and beginning we assume that all arrivals in $[n-1,n)$ are added to the queue at time $n-0$. If a service starts at time n and requires ν units of time, we shall consider it to start at time n and to end at time $n+\nu-0$.

Only as many arrivals as to maintain the queuelength less than or equal to L_1 are accepted. Any excessive customers are assumed to be permanently lost.

d. The initial conditions

We assume that at time $n = 0$, there are i_o customers present. $0 \leq i_o \leq L_1$. If $i_o \geq 1$, the customer in service at time $n = 0$ requires j_o, $1 \leq j_o \leq L_2$ additional units of service time. We make the convention that if $i_o = 0$, then $j_o = 0$ and conversely.

3. The Markov Chain

We denote by X_n the queuelength at time n and by Y_n the number of additional units of service time required by the customer in service at time n. We make the convention that $X_n = 0$ if and only if $Y_n = 0$.

It follows readily from our assumptions that the bivariate sequence $\{(X_n, Y_n), n \geq 0\}$ is a Markov chain with state space consisting of the point (0,0) and all points (i,j), $i = 1, \ldots, L_1$; $j = 1, \ldots, L_2$, and with initial state (i_o, j_o)..

The transition probability matrix of the Markov chain is easily written down. We shall not do so since the recurrence relations make judicious use of its special structure.

For fixed i_o and j_o, we define the conditional probabilities:

$$(1) \qquad P_n(i,j) = P\{X_n = i,\ Y_n = j \mid X_o = i_o,\ Y_o = j_o\}$$

The probabilities $P_n(i,j)$ satisfy the following recurrence relations in n for all $n \geq 0$.

$$(2) \quad (a) \qquad P_{n+1}(o,o) = p_o [P_n(o,o) + P_n(1,1)]$$

$$(b) \qquad P_{n+1}(i,j) = p_o\, P_n(i,j+1) + \sum_{\nu=1}^{i-1} p_{i-\nu}\, P_n(\nu,j+1)$$

$$+ r_j\, \{p_i\, P_n(o,o) + p_o\, P_n(i+1,1) + \sum_{\nu=1}^{i} p_{i-\nu+1}\, P_n(\nu,1)\}$$

for $i = 1, \ldots, K$ and $j = 1, \ldots, L_2 - 1$.

(c) $$P_{n+1}(i,j) = p_o\, P_n(i,j+1) + \sum_{\nu=i-K}^{i-1} p_{i-\nu}\, P_n(\nu,j+1)$$

$$+ r_j \{p_o\, P_n(i+1,1) + \sum_{\nu=i-K+1}^{i} p_{i-\nu+1}\, P_n(\nu,1)\}$$

for $i = K+1,\ldots, L_1 - 1$ and $j = 1,\ldots, L_2 - 1$.

(d) $$P_{n+1}(L_1,j) = P_n(L_1,j+1) + \sum_{\nu=1}^{K} (1 - \sum_{k=o}^{\nu-1} p_k)\, P_n(L_1 - \nu,j+1)$$

$$+ r_j \{\sum_{\nu=1}^{K} (1 - \sum_{k=o}^{\nu-1} p_k)\, P_n(L_1-\nu+1,1)\}$$

for $j = 1,\ldots, L_2 - 1$.

(e) $$P_{n+1}(i,L_2) = r_{L_2} \{p_i\, P_n(o,o) + p_o\, P_n(i+1,1)$$

$$+ \sum_{\nu=1}^{i} p_{i-\nu+1}\, P_n(\nu,1)\}$$

for $i = 1,\ldots, K$.

(f) $$P_{n+1}(i,L_2) = r_{L_2} \{p_o\, P_n(i+1,1) + \sum_{\nu=i-K+1}^{i} p_{i-\nu+1}\, P_n(\nu,1)\}$$

for $i = K+1,\ldots, L_1 - 1$.

(g) $$P_{n+1}(L_1,L_2) = r_{L_2} \{\sum_{\nu=1}^{K} (1 - \sum_{k=o}^{\nu-1} p_k)\, P_n(L_1 - \nu+1,1)\}$$

The recurrence is initialized by setting $P_o(i_o,j_o) = 1$ and $P_o(i,j) = 0$ for all other pairs (i,j). If the distribution of the random variables X_o, Y_o is given rather than exact values, this simply

amounts to a different definition of the initial array $P_o(i,j)$.

We note that the expressions in curly brackets in the formulas (2 b-g) depend on the first index only. This term corresponds to the case where the instant n+1 is the beginning of a new service.

This simplifying feature is used in the organization of the computer program. If the service time distribution depends on *the instant of initiation of a service*, only the factor r_j in the recurrence relations are affected.

The analogous, but simpler recurrence relations for the *unbounded* queue were examined analytically by Dafermos and Neuts [2].

The recurrence relations (2) are well suited for iterative computation. The author organized his computation so as to have at the n-th iteration only the quantities $P_n(o,o)$ and $P_n(i,j)$, $i = 1,\ldots, L_1$; $j = 1,\ldots, L_2$ in memory.

The quantities $P_n(o,o)$ and $P_n(i,j)$, $i = 1,\ldots, L_1$; $j = 1,\ldots, L_2$ make up the (conditional) *joint density of the queuelength X_n and the residual service time Y_n* at time n. This *joint* density is not in itself of great practical use; however from it the distribution (or the density) of the queuelength and the waitingtime at time n can be easily obtained. These "derived" queue features are considered next.

4. Derived Queue Features

a. The Queuelength Distribution

The probability that the queuelength at time n is zero, is given by $P_n(o,o)$. For $i = 1,\dots, L_1$, we have:

$$(3) \qquad P\{X_n = i \mid X_o = i_o,\ Y = j_o\} = \sum_{j=1}^{L_2} P_n(i,j).$$

Moments of the queuelength at time n may be obtained routinely.

b. The Waitingtime distribution

The waitingtime at time n is defined to be zero if and only if $X_n = Y_n = 0$. For $X_n = 1$, $Y_n = j$, the waitingtime equals j. For $X_n > 1$, $Y_n = j$ the waitingtime is the sum of j and $X_n - 1$ independent service times. The waitingtime is therefore an integer valued random variable with value between 0 and L_1L_2.

The density WT_ν, $\nu = o,\dots, L_1L_2$ is obtained as follows. Clearly $WT_o = P_n(o,o)$. The quantities WT_n for $1 \leq \nu \leq L_1L_2$ were obtained by evaluating the following convolution-polynomial.

Let $WT(\cdot)$ denote the density $\{WT_\nu,\ 1 \leq \nu \leq L_1L_2\}$ and let $R(\cdot)$ be the service time density $\{r_j,\ 1 \leq j \leq L_2\}$. Finally let $P_n(i,\cdot)$ be the density $\{P_n(i,j),\ 1 \leq j \leq L_2\}$ for $i = 1,\dots, L_1$, then $WT(\cdot)$ is given by:

$$(4) \qquad WT(\cdot) = P_n(1,\cdot) + P_n(2,\cdot) * R(\cdot) + \dots + P_n(L_1,\cdot) * R^{(L_1-1)}(\cdot)$$

where $R^{(k)}(\cdot)$ is the k-fold convolution of $R(\cdot)$ with itself.

The density $WT(\cdot)$ may be computed for each n by one of two procedures. Either the convolutions $R^{(k)}(\cdot)$, $k = 1,\dots, L_1 - 1$ may be computed

once and for all and only the convolutions with the arrays $P_n(i,\cdot)$ need to be computed at those time points n at which the density of the waitingtime is desired. This procedure results in a substantial increase in the central memory storage required by the program.

Alternatively, the density WT(·) may be computed by a convolution analogue of Horner's algorithm for the numerical evaluation of ordinary polynomials. This procedure consists in the successive evaluation of the sequences

$$(5) \qquad WT^{(1)}(\cdot) = P_n(L_1,\cdot)$$

$$WT^{(k)}(\cdot) = WT^{(k-1)}(\cdot) * R(\cdot) + P_n(L_1-k+1,\cdot),$$

for $k = 2,\ldots, L_1$. The sequence $WT^{(L_1)}(\cdot)$ is the desired sequence WT(·).

The latter procedure does not result in the use of core storage for intermediate quantities. For purposes of comparison and as a guard against rounding errors, both procedures were programmed and tested on large scale examples. The Horner convolution algorithm performed slightly better in all examples and no evidence of rounding errors were found. Its much smaller core requirements make it the more efficient of the two procedures.

Moments of the waitingtime at time n may be routinely computed.

5. Report on Computational Trials

A FORTRAN IV program was written by the author and tested on a variety of cases on the CDC 6500 at Purdue University. Even in the largest examples no evidence of rounding errors was found, even though all probabilities were printed to five decimal places and all computations were performed in single precision.

The full output consisted, in addition to the summary of the input data, of the following.

(a) the mean of the queuelength

(b) the cumulative distribution of the queuelength

(c) the mean waitingtime

(d) the cumulative distribution of the waitingtime

(e) the joint density of the queuelength X_n and the residual service time Y_n

All these for all values of n up to some upper value to be specified.

Since the full output is very voluminous and contains much more information than may be needed in the analysis of a given queue, options were written into the program which permit the deletion of the items (c), (d) or (e) from the printed output. A further option was created which permits the computation of the waitingtime distribution to be performed at certain specified time points only.

No systematic study of the processing time was made. For smaller examples the computation times were generally below 10 seconds. The following computation times for some larger examples are given for purposes of illustration only.

$L_1 = 30$	(max. queue length)
$L_2 = 6$	(max. duration of one service)
$K = 2$	(max. number of arrivals)
$NNN = 250$	(number of time points computed)

In the following CP is the processing time in seconds and LL is the number of lines of output, including the program listing.

a. full printed output

CP = 215.159 LL = 19265

b. the queuelength and waitingtime distributions only.

CP = 157.860 LL = 10656

c. the queuelength distribution for all time points and the waitingtime distribution only for time points which are multiples of twenty-five.

CP = 23.763 LL = 4916

6. Conclusions

By the use of only the most elementary structural properties, we have shown that the transient behavior of a substantial class of single server queues may be analyzed numerically. The approach presented here should prove itself to be useful in studying the build-up of unstable queues and the fluctuations of queues at traffic lights, highway merging ramps, service counters in public offices and retail outlets and many others.

This approach is well suited for many queueing processes which do not lend themselves to diffusion approximation methods. The amount of computing time used in typical examples also indicates a very substantial saving over that needed to analyze similar models by simulation methods.

Further work is currently being done to extend the applicability of this approach to longer queues and to much longer time periods. This extension however requires the use of mathematically more sophisticated properties of the queueing process.

Bibliography

[1] Leese, E. L. and Boyd, D. W.

Numerical Methods of Determining the Transient Behavior of Queues with Variable Arrival Rates

Journal of the Canadian Operations Research Society, 4, 1-13, 1966

[2] Dafermos, S. and Neuts, M. F.

A Single Server in Discrete Time

Cahiers du Centre de Recherche Opérationnelle, 13, 23-40, 1971

Appendix

This appendix contains a listing of the FORTRAN IV program written by the author to compute the timedependent features of the discrete time queue.

A sample output for a small scale problem is also included. This example analyzes a queue with 20 spaces for 20 time points. The computing time for this example was approximately 6 seconds of which 2.2 seconds were used for compilation. The distribution of the queuelength is printed for all time points and the distribution of the waitingtime is computed and printed for every fourth time point.

```
      PROGRAM QUEUE(INPUT,OUTPUT,TAPE5=INPUT)
C
C THIS PROGRAM WAS DEVELOPED BY MARCEL F. NEUTS - DEPART-
C MENT OF STATISTICS - PURDUE UNIVERSITY - WEST LAFAYETTE -
C INDIANA. OCTOBER 1971.
C
C THIS PROGRAM COMPUTES THE TIME DEPENDENT FEATURES OF A
C SINGLE SERVER DISCRETE TIME QUEUE WITH A FINITE WAITING-
C ROOM.
C WITH THE PRESENT DIMENSION STATEMENTS A WATTINGROOM OF
C SIZE UP TO ONE HUNDRED MAY BE STUDIED. THE DENSITY OF THE
C SERVICE TIME CAN BE CONCENTRATED ON UP TO THIRTY POINTS.
C
C THE FULL OUTPUT OF THIS PROGRAM INCLUDES THE FOLLOWING:
C
C 1. THE MEAN QUEUE LENGTH AT TIME N.
C 2. THE DISTRIBUTION OF THE QUEUE LENGTH AT TIME N.
C 3. THE MEAN WAITINGTIME AT TIME N.
C 4. THE DISTRIBUTION OF THE WAITINGTIME AT TIME N.
C 5. THE JOINT DENSITY OF THE QUEUE LENGTH AND THE RESIDUAL
C    SERVICE TIME AT TIME N.
C
C ALL THESE ARE COMPUTED FOR N UP TO A SPECIFIED VALUE NNN.
C
C BY USE OF THE VARIOUS OPTIONS, LISTED BELOW, SOME OF THESE
C FEATURES MAY BE DELETED FROM THE PRINTED OUTPUT.
C
C THE THEORETICAL DEVELOPMENT OF THE DISCRETE TIME QUEUE
C WITH AN UNBOUNDED QUEUE LENGTH MAY BE FOUND IN:
C
C * STELLA C. DAFERMOS AND MARCEL F. NEUTS
C * A SINGLE SERVER QUEUE IN DISCRETE TIME *
C * CAHIERS DU CENTRE DE RECHERCHE OPERATIONNELLE - 1971.
C
C THE FOLLOWING IS A COMPANION PAPER TO THE PRESENT PROGRAM:
C
C * MARCEL F. NEUTS
C * THE SINGLE SERVER QUEUE IN DISCRETE TIME - NUMERICAL
C * ANALYSIS *
C * PURDUE MIMEOGRAPH SERIES - DEPT. OF STATISTICS.
C * PURDUE UNIVERSITY - WEST LAFAYETTE - IN - 47907
C
      DIMENSION P(100),R(30),PP(100,30),X(100),Y(100,30)
      DIMENSION Z(100),W(100),WT(3000)
      INTEGER OPT(10)
C
C     ***** T H E   O P T I O N S *****
C
C IN ORDER TO PRINT OUT THE JOINT DENSITY OF THE QUEUE LENGTH
C AND THE RESIDUAL SERVICE TIME, SET OPT(1)=1 - OTHERWISE
C SET OPT(1)=0
C
      READ(5,999) OPT(1)
C
C IN ORDER TO PRINT OUT THE DISTRIBUTION OF THE WAITING-
C TIME, SET OPT(2)=1 - OTHERWISE SET OPT(2)=0
C
      READ(5,999) OPT(2)
```

```
C
C THE USER MAY WISH TO COMPUTE AND PRINT THE DISTRIBUTION OF
C THE WAITINGTIME ONLY AT TIME POINTS WHICH ARE A MULTIPLE OF
C A CONSTANT NR. THIS TO SAVE ON PROCESSING TIME AND ON THE
C NUMBER OF LINES OF OUTPUT. IN THIS CASE THE IDENTIFIER
C OPT(3) SHOULD BE SET EQUAL TO ONE AND THE NUMBER NR SHOULD BE
C GIVEN. OPT(3) AND NR ARE TO BE GIVEN IN AN I1,I2 FORMAT.
C
      READ(5,997) OPT(3),NR
      XNR=NR
      KTEST1=OPT(1)
      KTEST2=OPT(2)
      KTEST3=OPT(3)
      IF(KTEST3.EQ.1) KTEST2=0
      IF(KTEST3.EQ.0) KTEST4=0
C
C     ***** T H E   D A T A *****
C
C L1 IS THE SIZE OF THE WAITINGROOM. L1 IS LT. 101
C
      READ(5,1001) L1
C
C L2 IS THE NUMBER OF POINTS ON WHICH THE DENSITY OF THE
C SERVICE TIME IS CONCENTRATED. L2 IS LT. 31
C
      READ(5,1001) L2
C
C K IS THE MAXIMUM NUMBER OF ARRIVALS PER UNIT OF TIME.
C K SHOULD BE AT LEAST ONE AND STRICTLY LESS THAN L1.
C
      READ(5,1001) K
C
C R(J) IS THE PROBABILITY THAT A SERVICE TIME LASTS FOR J
C UNITS OF TIME.
C ONE SHOULD VERIFY BEFOREHAND THAT THE SUM R(1)+...R(L2)
C IS EQUAL TO ONE.
C
      READ(5,1002) (R(J),J=1,L2)
C
C P(J) IS THE PROBABILITY THAT J CUSTOMERS JOIN THE QUEUE
C DURING A UNIT OF TIME.
C THE INDEX J RUNS FROM ONE TO K.
C
      READ(5,1002) (P(J),J=1,K)
C
C P0 IS THE PROBABILITY THAT NO CUSTOMERS ARRIVE DURING A
C UNIT OF TIME.
C ONE SHOULD VERIFY BEFOREHAND THAT P0 + P(1) +...+P(K)
C IS EQUAL TO ONE.
C
      READ(5,1003) P0
C
C I0 IS THE INITIAL QUEUE LENGTH.
C J0 IS THE INITIAL RESIDUAL SERVICE TIME.
C IF I0=0, THEN J0=0 AND CONVERSELY.
C I0 SHOULD NOT EXCEED L1.
C J0 SHOULD NOT EXCEED L2.
C
```

```
      READ(5,1001) I0
      READ(5,1001) J0
C
C NNN IS THE MAXIMUM TIME POINT FOR WHICH THE QUEUE
C FEATURES ARE COMPUTED. NNN SHOULD BE AT LEAST ONE AND
C AT MOST 9999. NOTE HOWEVER THAT THE PROCESSING TIME
C AND THE NUMBER OF LINES OF OUTPUT GROW PROPORTIONATELY TO
C THE VALUE OF NNN.
C
      READ(5,1004) NNN
      IF(I0.EQ.0.AND.J0.EQ.0)GOTO 2001
      PP(I0,J0)=1.
      GOTO 2002
 2001 P00=1.
 2002 N=0
      X11=P0
      DO 21 I=1,K
      XI=I
      X11=X11+P(I)
      X1=X1+XI*P(I)
   21 CONTINUE
      DO 22 J=1,L2
      XJ=J
      X11=X11+R(J)
      X2=X2+XJ*R(J)
   22 CONTINUE
      X11=X11-2.
      X11=ABS(X11)
      PRINT 1014
      PRINT 1022,K,L2,L1
      N1=0
      PRINT 1007,N1,P0,(J,P(J),J=1,K)
      PRINT 1012
      PRINT 1010,(J,R(J),J=1,L2)
      PRINT 1012
      PRINT 1008,I0,J0
      PRINT 1012
      PRINT 1009,NNN
      PRINT 1012
      PRINT 1013,X1,X2
      PRINT 1006
C
C     ***** D I A G N O S T I C S *****
C
      IF(L1.LE.K) GOTO 2005
      IF(I0.EQ.0.AND.J0.NE.0.OR.I0.NE.0.AND.J0.EQ.0)GOTO 2005
      IF(X11.GT..00000001) GOTO 2005
      IF(KTEST3.EQ.1.AND.NR.LT.2) GOTO 2005
C
C
C
      L11=L1-1
      L21=L2-1
      L22=L2+1
      M1=L1*L2
      K1=K+1
      DO 13 I=1,L1
      Y(I,L2)=0.0
```

```
   13 W(1)=1.-P0
      IF(K.EQ.1)GOTO 2003
      DO 20 I=2,K
      W(I)=W(I-1)-P(I-1)
   20 CONTINUE
C
C AT THIS STAGE THE INPUT DATA HAVE BEEN READ IN, THE
C PP-ARRAY HAS BEEN INITIALIZED AND THE INPUT DATA HAVE
C BEEN PRINTED OUT AND SUBJECTED TO SOME RUDIMENTARY
C DIAGNOSTIC TESTS. THE NEXT LINE STARTS THE MAIN LOOP
C WHICH IS REPEATED NNN TIMES.
C
C     ***** T H E   M A I N   L O O P *****
C
 2003 N=N+1
      IF(N.GT.NNN)STOP
C
C THIS PORTION OF THE PROGRAM COMPUTES THE NEW PP-ARRAY.
C PP(I,J) IS THE PROBABILITY THAT AT THE TIME CONSIDERED
C THERE ARE I CUSTOMERS IN THE SYSTEM AND THE RESIDUAL
C SERVICE TIME OF THE CUSTOMER BEING SERVED IS J.
C THIS IS FOR I BETWEEN ONE AND L1, FOR J BETWEEN ONE AND
C L2. THE IDENTIFIER P00 CONTAINS THE PROBABILITY THAT THE
C QUEUE IS EMPTY.
C
      PRINT 1000
      PRINT 1017,N
      PRINT 1012
      X00=P00+PP(1,1)
      DO 1 I=1,K
      X(I)=P(I)*P00+P0*PP(I+1,1)
      II=I-1
      DO 3 NU=1,I
      X(I)=X(I)+P(I+1-NU)*PP(NU,1)
    3 CONTINUE
    4 DO 2 J=1,L21
      Y(I,J)=P0*PP(I,J+1)
      IF(I.EQ.1)GOTO 51
      DO 5 NV=1,II
      Y(I,J)=Y(I,J)+P(I-NV)*PP(NV,J+1)
    5 CONTINUE
   51 CONTINUE
    2 CONTINUE
    1 CONTINUE
      DO 6 I=K1,L11
      NU1=I-K
      NU2=NU1+1
      II=I-1
      X(I)=P0*PP(I+1,1)
      DO 7 NU=NU2,I
      X(I)=X(I)+P(I+1-NU)*PP(NU,1)
    7 CONTINUE
      DO 9 J=1,L21
      Y(I,J)=P0*PP(I,J+1)
      DO 8 NV=NU1,II
      Y(I,J)=Y(I,J)+P(I-NV)*PP(NV,J+1)
    8 CONTINUE
    9 CONTINUE
```

```
    6 CONTINUE
      X(L1)=0.0
      DO 10 NU=1,K
      X(L1)=X(L1)+W(NU)*PP(L1+1-NU,1)
   10 CONTINUE
      DO 11 J=1,L21
      Y(L1,J)=PP(L1,J+1)
      DO 12 NU=1,K
      Y(L1,J)=Y(L1,J)+W(NU)*PP(L1-NU,J+1)
   12 CONTINUE
   11 CONTINUE
      P00=P0*X00
C
C Z(I) CONTAINS FIRST THE DENSITY AND NEXT THE DISTRIBUTION
C OF THE QUEUE LENGTH AT THE TIME POINT CONSIDERED.
C XMN CONTAINS THE MEAN QUEUE LENGTH AT THE TIME POINT
C CONSIDERED.
C
      XMN=1.-P00
      DO 14 I=1,L1
      Z(I)=0.0
      DO 15 J=1,L2
      PP(I,J)=Y(I,J)+R(J)*X(I)
      Z(I)=Z(I)+PP(I,J)
   15 CONTINUE
   14 CONTINUE
      Z(1)=Z(1)+P00
      XMN=XMN+1.-Z(1)
      DO 18 I=2,L1
      Z(I)=Z(I)+Z(I-1)
      XMN=XMN+1.-Z(I)
   18 CONTINUE
      IF(KTEST3.EQ.0) GOTO 2008
      XN=N
      U1=AMOD(XN,XNR)
      IF(U1.LT.0.9) KTEST4=1
      IF(U1.GT.0.9) KTEST4=0
      IF(KTEST4.EQ.1) GOTO 2010
      IF(KTEST4.EQ.0) GOTO 2009
 2008 CONTINUE
C
C THIS PORTION OF THE PROGRAM COMPUTES THE DISTRIBUTION OF
C THE VIRTUAL WAITINGTIME AT TIME N. THE ALGORITHM IS AN
C ANALOGUE OF HORNER≠S METHOD FOR THE EVALUATION OF ORDI-
C NARY POLYNOMIALS, BUT ADAPTED HERE TO CONVOLUTION PRODUCTS.
C
      IF(KTEST2.EQ.0) GOTO 2006
 2010 CONTINUE
      DO 32 J=1,L2
      WT(J)=PP(L1,J)
   32 CONTINUE
      DO 33 J=L22,M1
      WT(J)=0.0
   33 CONTINUE
      MN1=1
      MN3=2
      MN4=L2
      DO 34 JX=1,L11
```

```
      JX1=L1-JX
      MN2=MN4
      MN4=MN2+L2
      MN5=MN4+2
      DO 35 J=2,MN4
      JR=MN5-J
      WT(JR)=0.0
      MN6=MAX0(1,JR-MN2)
      MN7=MIN0(L2,JR-1)
      DO 35 NU=MN6,MN7
      WT(JR)=WT(JR)+R(NU)*WT(JR-NU)
   35 CONTINUE
      WT(1)=0.0
      DO 37 JJ=1,L2
      WT(JJ)=WT(JJ)+PP(JX1,JJ)
   37 CONTINUE
   34 CONTINUE
      WT(1)=P00+WT(1)
      ZMN=2.-P00-WT(1)
      DO 36 J=2,M1
      WT(J)=WT(J)+WT(J-1)
      ZMN=ZMN+1.-WT(J)
   36 CONTINUE
 2006 CONTINUE
C
C THE PRINT STATEMENTS FOR THE REQUIRED OUTPUT.
C
 2009 PRINT 1016,N,XMN
      PRINT 1012
      PRINT 1018,N
      PRINT 1012
      PRINT 1011,N1,P00,(I,Z(I),I=1,L1)
      PRINT 1012
      IF(KTEST2.EQ.0.AND.KTEST4.EQ.0) GOTO 2007
      PRINT 1021,N,ZMN
      PRINT 1012
      PRINT 1020,N,N1,P00,(J,WT(J),J=1,M1)
 2007 CONTINUE
      IF(KTEST1.EQ.0) GOTO 2004
      PRINT 1012
      PRINT 1019,N
      PRINT 1015,P00
      PRINT 1012
      DO 17 I=1,L1
      PRINT 1005,I,(PP(I,J),J=1,L2)
   17 CONTINUE
 2004 GO TO 2003
 2005 PRINT 998
C
C THE FORMAT STATEMENTS.
C
  997 FORMAT(I1,I2)
  998 FORMAT(≠  ATTENTION: THERE ARE ERRORS IN THE INPUT≠,
     *≠ DATA. PLEASE CHECK. ≠)
  999 FORMAT(I1)
 1000 FORMAT(≠1≠)
 1001 FORMAT(I3)
 1002 FORMAT(3F7.5)
```

```
1003 FORMAT(F7.5)
1004 FORMAT(I4)
1005 FORMAT(3X,I3,10F7.4,(6X,10F7.4))
1006 FORMAT(//)
1007 FORMAT(≠  THE DENSITY OF THE NUMBER OF ARRIVALS PER≠,
    *≠ UNIT OF TIME≠,//(2X,10(I4,F8.5)))
1008 FORMAT(≠  THE INITIAL QUEUE LENGTH IS ≠,I3,/≠  THE≠,
    *≠ INITIAL RESIDUAL SERVICE TIME IS ≠,I3)
1009 FORMAT(≠  THE NUMBER OF TIME POINTS COMPUTED IS≠,
    *I4)
1010 FORMAT(≠  THE DENSITY OF THE SERVICE TIMES≠,//2X,
    *(10(I4,F8.5)))
1011 FORMAT(3X,10(I4,F8.5))
1012 FORMAT(/)
1013 FORMAT(2X,≠THE MEAN NR. OF ARRIVALS PER UNIT-TIME:≠,
    *F10.4,/≠  THE MEAN SERVICE TIME:≠,F10.4)
1014 FORMAT(≠1≠,////≠  THE TRANSIENT BEHAVIOR OF A ≠,
    *≠DISCRETE TIME QUEUE WITH A FINITE WAITINGROOM≠,
    *//)
1015 FORMAT(≠  THE QUEUE IS EMPTY WITH PROBABILITY≠,
    *F9.5)
1016 FORMAT(≠  AT TIME N =≠,I4,≠ THE MEAN QUEUE LENGTH≠,
    *≠ EQUALS≠,F10.4)
1017 FORMAT(≠  THE QUEUE CHARACTERISTICS AT TIME N = ≠,
    *I4)
1018 FORMAT(≠  THE DISTRIBUTION OF THE QUEUE LENGTH ≠,
    *≠AT TIME N = ≠,I4/)
1019 FORMAT(≠  THE JOINT DENSITY OF THE QUEUE LENGTH ≠,
    *≠AND THE RESIDUAL SERVICE TIME AT TIME N =≠,I4/)
1020 FORMAT(≠  THE DISTRIBUTION OF THE WAITINGTIME AT ≠,
    *≠TIME N = ≠,I4,//(3X,10(I4,F8.5)))
1021 FORMAT(≠  THE MEAN WAITINGTIME AT TIME N =≠,I4,
    *≠ IS≠,F12.4)
1022 FORMAT(≠  THE UPPER LIMIT OF THE NUMBER OF ARRIVALS≠,
    *≠ PER UNIT OF TIME IS≠,I3,/≠  THE UPPER LIMIT OF THE≠,
    *≠ NUMBER OF UNITS OF SERVICE-TIME PER CUSTOMER IS≠,
    *I3,/≠  THE UPPER LIMIT TO THE NUMBER OF CUSTOMERS≠,
    *≠ IN THE SYSTEM IS≠,I4/)
     END
```

```
THE TRANSIENT BEHAVIOR OF A DISCRETE TIME QUEUE WITH A FINITE WAITINGROOM

THE UPPER LIMIT OF THE NUMBER OF ARRIVALS PER UNIT OF TIME IS  5
THE UPPER LIMIT OF THE NUMBER OF UNITS OF SERVICE-TIME PER CUSTOMER IS  9
THE UPPER LIMIT TO THE NUMBER OF CUSTOMERS IN THE SYSTEM IS  20

THE DENSITY OF THE NUMBER OF ARRIVALS PER UNIT OF TIME

   0  .95000   1  .01500   2 0.00000   3  .01500   4  .01500   5  .00500

THE DENSITY OF THE SERVICE TIMES

   1  .50000   2  .10000   3  .20000   4  .05000   5  .05000   6  .02500   7  .02500   8  .02500   9  .02500

THE INITIAL QUEUE LENGTH IS    0
THE INITIAL RESIDUAL SERVICE TIME IS    0

THE NUMBER OF TIME POINTS COMPUTED IS  20

THE MEAN NR. OF ARRIVALS PER UNIT-TIME:      .1450
THE MEAN SERVICE TIME:    2.5000
```

```
THE QUEUE CHARACTERISTICS AT TIME N =     1

AT TIME N =    1 THE MEAN QUEUE LENGTH EQUALS      .1450

THE DISTRIBUTION OF THE QUEUE LENGTH AT TIME N =     1

   0  .95000   1  .96500   2  .96500   3  .98000   4  .99500   5 1.00000   6 1.00000   7 1.00000   8 1.00000   9 1.00000
  10 1.00000  11 1.00000  12 1.00000  13 1.00000  14 1.00000  15 1.00000  16 1.00000  17 1.00000  18 1.00000  19 1.00000
  20 1.00000

THE QUEUE CHARACTERISTICS AT TIME N =     2

AT TIME N =    2 THE MEAN QUEUE LENGTH EQUALS      .2650

THE DISTRIBUTION OF THE QUEUE LENGTH AT TIME N =     2

   0  .90962   1  .93111   2  .93835   3  .96767   4  .99127   5  .99881   6  .99922   7  .99964   8  .99990   9  .99999
  10 1.00000  11 1.00000  12 1.00000  13 1.00000  14 1.00000  15 1.00000  16 1.00000  17 1.00000  18 1.00000  19 1.00000
  20 1.00000

THE QUEUE CHARACTERISTICS AT TIME N =     3

AT TIME N =    3 THE MEAN QUEUE LENGTH EQUALS      .3723

THE DISTRIBUTION OF THE QUEUE LENGTH AT TIME N =     3

   0  .87232   1  .90174   2  .91707   3  .95615   4  .98700   5  .99680   6  .99797   7  .99904   8  .99970   9  .99994
  10  .99998  11  .99999  12 1.00000  13 1.00000  14 1.00000  15 1.00000  16 1.00000  17 1.00000  18 1.00000  19 1.00000
  20 1.00000
```

```
THE QUEUE CHARACTERISTICS AT TIME N =     4

AT TIME N =   4 THE MEAN QUEUE LENGTH EQUALS      .4633

THE DISTRIBUTION OF THE QUEUE LENGTH AT TIME N =    4

  0  .84073   1  .87621   2  .90047   3  .94792   4  .98299   5  .99432   6  .99650   7  .99838   8  .99940   9  .99984
 10  .99994  11  .99997  12  .99999  13 1.00000  14 1.00000  15 1.00000  16 1.00000  17 1.00000  18 1.00000  19 1.00000
 20 1.00000

THE MEAN WAITINGTIME AT TIME N =   4 IS       1.1819

THE DISTRIBUTION OF THE WAITINGTIME AT TIME N =    4

  0  .84073   1  .85538   2  .86709   3  .88087   4  .89378   5  .90744   6  .92041   7  .93252   8  .94350   9  .95327
 10  .96180  11  .96930  12  .97562  13  .98084  14  .98504  15  .98838  16  .99103  17  .99311  18  .99474  19  .99599
 20  .99695  21  .99768  22  .99823  23  .99865  24  .99897  25  .99921  26  .99940  27  .99954  28  .99965  29  .99974
 30  .99980  31  .99985  32  .99989  33  .99991  34  .99993  35  .99995  36  .99996  37  .99997  38  .99998  39  .99999
 40  .99999  41  .99999  42  .99999  43 1.00000  44 1.00000  45 1.00000  46 1.00000  47 1.00000  48 1.00000  49 1.00000
 50 1.00000  51 1.00000  52 1.00000  53 1.00000  54 1.00000  55 1.00000  56 1.00000  57 1.00000  58 1.00000  59 1.00000
 60 1.00000  61 1.00000  62 1.00000  63 1.00000  64 1.00000  65 1.00000  66 1.00000  67 1.00000  68 1.00000  69 1.00000
 70 1.00000  71 1.00000  72 1.00000  73 1.00000  74 1.00000  75 1.00000  76 1.00000  77 1.00000  78 1.00000  79 1.00000
 80 1.00000  81 1.00000  82 1.00000  83 1.00000  84 1.00000  85 1.00000  86 1.00000  87 1.00000  88 1.00000  89 1.00000
 90 1.00000  91 1.00000  92 1.00000  93 1.00000  94 1.00000  95 1.00000  96 1.00000  97 1.00000  98 1.00000  99 1.00000
100 1.00000 101 1.00000 102 1.00000 103 1.00000 104 1.00000 105 1.00000 106 1.00000 107 1.00000 108 1.00000 109 1.00000
110 1.00000 111 1.00000 112 1.00000 113 1.00000 114 1.00000 115 1.00000 116 1.00000 117 1.00000 118 1.00000 119 1.00000
120 1.00000 121 1.00000 122 1.00000 123 1.00000 124 1.00000 125 1.00000 126 1.00000 127 1.00000 128 1.00000 129 1.00000
130 1.00000 131 1.00000 132 1.00000 133 1.00000 134 1.00000 135 1.00000 136 1.00000 137 1.00000 138 1.00000 139 1.00000
140 1.00000 141 1.00000 142 1.00000 143 1.00000 144 1.00000 145 1.00000 146 1.00000 147 1.00000 148 1.00000 149 1.00000
150 1.00000 151 1.00000 152 1.00000 153 1.00000 154 1.00000 155 1.00000 156 1.00000 157 1.00000 158 1.00000 159 1.00000
160 1.00000 161 1.00000 162 1.00000 163 1.00000 164 1.00000 165 1.00000 166 1.00000 167 1.00000 168 1.00000 169 1.00000
170 1.00000 171 1.00000 172 1.00000 173 1.00000 174 1.00000 175 1.00000 176 1.00000 177 1.00000 178 1.00000 179 1.00000
180 1.00000
```

```
THE QUEUE CHARACTERISTICS AT TIME N =     5

AT TIME N =    5 THE MEAN QUEUE LENGTH EQUALS      .5424

THE DISTRIBUTION OF THE QUEUE LENGTH AT TIME N =     5

   0  .81261   1  .85511   2  .88748   3  .94078   4  .97875   5  .99162   6  .99497   7  .99765   8  [illegible]   9  .99970
  10  .99987  11  .99994  12  .99998  13  .99999  14 1.00000  15 1.00000  16 1.00000  17 1.00000  18 [illegible]  19 1.00000
  20 1.00000



THE QUEUE CHARACTERISTICS AT TIME N =     6

AT TIME N =    6 THE MEAN QUEUE LENGTH EQUALS      .6111

THE DISTRIBUTION OF THE QUEUE LENGTH AT TIME N =     6

   0  .78864   1  .83768   2  .87674   3  .93432   4  .97451   5  .98886   6  .99340   7  .99687   8  [illegible]   9  .99952
  10  .99977  11  .99990  12  .99996  13  .99999  14  .99999  15 1.00000  16 1.00000  17 1.00000  18 [illegible]  19 1.00000
  20 1.00000



THE QUEUE CHARACTERISTICS AT TIME N =     7

AT TIME N =    7 THE MEAN QUEUE LENGTH EQUALS      .6711

THE DISTRIBUTION OF THE QUEUE LENGTH AT TIME N =     7

   0  .76817   1  .82319   2  .86772   3  .92839   4  .97032   5  .98612   6  .99184   7  .99606   8  [illegible]   9  .99930
  10  .99966  11  .99985  12  .99994  13  .99998  14  .99999  15 1.00000  16 1.00000  17 1.00000  18 [illegible]  19 1.00000
  20 1.00000
```

```
THE QUEUE CHARACTERISTICS AT TIME N =    5

AT TIME N =   5 THE MEAN QUEUE LENGTH EQUALS      .5424

THE DISTRIBUTION OF THE QUEUE LENGTH AT TIME N =    5

  0  .81261   1  .85511   2  .88748   3  .94078   4  .97875   5  .99162   6  .99497   7  .99765   8  [illegible]   9  .99970
 10  .99987  11  .99994  12  .99998  13  .99999  14 1.00000  15 1.00000  16 1.00000  17 1.00000  18 [illegible]  19 1.00000
 20 1.00000


THE QUEUE CHARACTERISTICS AT TIME N =    6

AT TIME N =   6 THE MEAN QUEUE LENGTH EQUALS      .6111

THE DISTRIBUTION OF THE QUEUE LENGTH AT TIME N =    6

  0  .78864   1  .83768   2  .87674   3  .93432   4  .97451   5  .98886   6  .99340   7  .99687   8  [illegible]   9  .99952
 10  .99977  11  .99990  12  .99996  13  .99999  14  .99999  15 1.00000  16 1.00000  17 1.00000  18 [illegible]  19 1.00000
 20 1.00000


THE QUEUE CHARACTERISTICS AT TIME N =    7

AT TIME N =   7 THE MEAN QUEUE LENGTH EQUALS      .6711

THE DISTRIBUTION OF THE QUEUE LENGTH AT TIME N =    7

  0  .76817   1  .82319   2  .86772   3  .92839   4  .97032   5  .98612   6  .99184   7  .99606   8  .99837   9  .99930
 10  .99966  11  .99985  12  .99994  13  .99998  14  .99999  15 1.00000  16 1.00000  17 1.00000  18 1.00000  19 1.00000
 20 1.00000
```

```
THE QUEUE CHARACTERISTICS AT TIME N =   12

AT TIME N =  12 THE MEAN QUEUE LENGTH EQUALS      .8743

THE DISTRIBUTION OF THE QUEUE LENGTH AT TIME N =   12

   0  .70563   1  .77884   2  .83818   3  .90603   4  .95319   5  .97470   6  .98510   7  .99213   8  .99604   9  .99796
  10  .99694  11  .99947  12  .99975  13  .99988  14  .99994  15  .99997  16  .99999  17  .99999  18 1.00000  19 1.00000
  20 1.00000

THE MEAN WAITINGTIME AT TIME N =  12 IS       2.2182

THE DISTRIBUTION OF THE WAITINGTIME AT TIME N =   12

   0  .70563   1  .73513   2  .76120   3  .78801   4  .81280   5  .83687   6  .85901   7  .87930   8  .89747   9  .91351
  10  .92747  11  .93965  12  .95005  13  .95885  14  .96618  15  .97227  16  .97731  17  .98146  18  .98488  19  .98769
  20  .98998  21  .99184  22  .99337  23  .99461  24  .99562  25  .99645  26  .99712  27  .99767  28  .99812  29  .99848
  30  .99877  31  .99901  32  .99920  33  .99936  34  .99949  35  .99959  36  .99967  37  .99974  38  .99979  39  .99983
  40  .99987  41  .99989  42  .99991  43  .99993  44  .99995  45  .99996  46  .99997  47  .99997  48  .99998  49  .99998
  50  .99999  51  .99999  52  .99999  53  .99999  54  .99999  55 1.00000  56 1.00000  57 1.00000  58 1.00000  59 1.00000
  60 1.00000  61 1.00000  62 1.00000  63 1.00000  64 1.00000  65 1.00000  66 1.00000  67 1.00000  68 1.00000  69 1.00000
  70 1.00000  71 1.00000  72 1.00000  73 1.00000  74 1.00000  75 1.00000  76 1.00000  77 1.00000  78 1.00000  79 1.00000
  80 1.00000  81 1.00000  82 1.00000  83 1.00000  84 1.00000  85 1.00000  86 1.00000  87 1.00000  88 1.00000  89 1.00000
  90 1.00000  91 1.00000  92 1.00000  93 1.00000  94 1.00000  95 1.00000  96 1.00000  97 1.00000  98 1.00000  99 1.00000
 100 1.00000 101 1.00000 102 1.00000 103 1.00000 104 1.00000 105 1.00000 106 1.00000 107 1.00000 108 1.00000 109 1.00000
 110 1.00000 111 1.00000 112 1.00000 113 1.00000 114 1.00000 115 1.00000 116 1.00000 117 1.00000 118 1.00000 119 1.00000
 120 1.00000 121 1.00000 122 1.00000 123 1.00000 124 1.00000 125 1.00000 126 1.00000 127 1.00000 128 1.00000 129 1.00000
 130 1.00000 131 1.00000 132 1.00000 133 1.00000 134 1.00000 135 1.00000 136 1.00000 137 1.00000 138 1.00000 139 1.00000
 140 1.00000 141 1.00000 142 1.00000 143 1.00000 144 1.00000 145 1.00000 146 1.00000 147 1.00000 148 1.00000 149 1.00000
 150 1.00000 151 1.00000 152 1.00000 153 1.00000 154 1.00000 155 1.00000 156 1.00000 157 1.00000 158 1.00000 159 1.00000
 160 1.00000 161 1.00000 162 1.00000 163 1.00000 164 1.00000 165 1.00000 166 1.00000 167 1.00000 168 1.00000 169 1.00000
 170 1.00000 171 1.00000 172 1.00000 173 1.00000 174 1.00000 175 1.00000 176 1.00000 177 1.00000 178 1.00000 179 1.00000
 180 1.00000
```

```
THE QUEUE CHARACTERISTICS AT TIME N =   13

AT TIME N =  13 THE MEAN QUEUE LENGTH EQUALS     .9014

THE DISTRIBUTION OF THE QUEUE LENGTH AT TIME N =   13

  0  .69837   1  .77355   2  .83421   3  .90269   4  .95058   5  .97295   6  .98399   7  .99141   8  .99558   9  .99768
 10  .99878  11  .99939  12  .99970  13  .99985  14  .99993  15  .99997  16  .99998  17  .99999  18 1.00000  19 1.00000
 20 1.00000


THE QUEUE CHARACTERISTICS AT TIME N =   14

AT TIME N =  14 THE MEAN QUEUE LENGTH EQUALS     .9256

THE DISTRIBUTION OF THE QUEUE LENGTH AT TIME N =   14

  0  .69222   1  .76894   2  .83061   3  .89963   4  .94821   5  .97133   6  .98293   7  .99071   8  .99514   9  .99741
 10  .99863  11  .99930  12  .99965  13  .99982  14  .99991  15  .99996  16  .99998  17  .99999  18 1.00000  19 1.00000
 20 1.00000


THE QUEUE CHARACTERISTICS AT TIME N =   15

AT TIME N =  15 THE MEAN QUEUE LENGTH EQUALS     .9474

THE DISTRIBUTION OF THE QUEUE LENGTH AT TIME N =   15

  0  .68696   1  .76486   2  .82733   3  .89685   4  .94605   5  .96984   6  .98193   7  .99003   8  .99470   9  .99715
 10  .99847  11  .99920  12  .99959  13  .99979  14  .99990  15  .99995  16  .99998  17  .99999  18  .99999  19 1.00000
 20 1.00000
```

```
THE QUEUE CHARACTERISTICS AT TIME N =   16

AT TIME N =  16 THE MEAN QUEUE LENGTH EQUALS      .9672

THE DISTRIBUTION OF THE QUEUE LENGTH AT TIME N =   16

   0  .68240   1  .76121   2  .82435   3  .89432   4  .94408   5  .96844   6  .98097   7  .98939   8  .99429   9  .99689
  10  .99831  11  .99911  12  .99954  13  .99976  14  .99988  15  .99994  16  .99997  17  .99999  18  .99999  19 1.00000
  20 1.00000

THE MEAN WAITINGTIME AT TIME N =  16 IS      2.4514

THE DISTRIBUTION OF THE WAITINGTIME AT TIME N =   16

   0  .68240   1  .71412   2  .74217   3  .77063   4  .79690   5  .82229   6  .84566   7  .86709   8  .88633   9  .90339
  10  .91831  11  .93139  12  .94265  13  .95225  14  .96033  15  .96710  16  .97276  17  .97748  18  .98141  19  .98466
  20  .98736  21  .98958  22  .99142  23  .99294  24  .99419  25  .99523  26  .99608  27  .99678  28  .99736  29  .99784
  30  .99824  31  .99856  32  .99882  33  .99904  34  .99922  35  .99936  36  .99948  37  .99958  38  .99966  39  .99972
  40  .99978  41  .99982  42  .99985  43  .99988  44  .99990  45  .99992  46  .99994  47  .99995  48  .99996  49  .99997
  50  .99997  51  .99998  52  .99998  53  .99999  54  .99999  55  .99999  56  .99999  57  .99999  58 1.00000  59 1.00000
  60 1.00000  61 1.00000  62 1.00000  63 1.00000  64 1.00000  65 1.00000  66 1.00000  67 1.00000  68 1.00000  69 1.00000
  70 1.00000  71 1.00000  72 1.00000  73 1.00000  74 1.00000  75 1.00000  76 1.00000  77 1.00000  78 1.00000  79 1.00000
  80 1.00000  81 1.00000  82 1.00000  83 1.00000  84 1.00000  85 1.00000  86 1.00000  87 1.00000  88 1.00000  89 1.00000
  90 1.00000  91 1.00000  92 1.00000  93 1.00000  94 1.00000  95 1.00000  96 1.00000  97 1.00000  98 1.00000  99 1.00000
 100 1.00000 101 1.00000 102 1.00000 103 1.00000 104 1.00000 105 1.00000 106 1.00000 107 1.00000 108 1.00000 109 1.00000
 110 1.00000 111 1.00000 112 1.00000 113 1.00000 114 1.00000 115 1.00000 116 1.00000 117 1.00000 118 1.00000 119 1.00000
 120 1.00000 121 1.00000 122 1.00000 123 1.00000 124 1.00000 125 1.00000 126 1.00000 127 1.00000 128 1.00000 129 1.00000
 130 1.00000 131 1.00000 132 1.00000 133 1.00000 134 1.00000 135 1.00000 136 1.00000 137 1.00000 138 1.00000 139 1.00000
 140 1.00000 141 1.00000 142 1.00000 143 1.00000 144 1.00000 145 1.00000 146 1.00000 147 1.00000 148 1.00000 149 1.00000
 150 1.00000 151 1.00000 152 1.00000 153 1.00000 154 1.00000 155 1.00000 156 1.00000 157 1.00000 158 1.00000 159 1.00000
 160 1.00000 161 1.00000 162 1.00000 163 1.00000 164 1.00000 165 1.00000 166 1.00000 167 1.00000 168 1.00000 169 1.00000
 170 1.00000 171 1.00000 172 1.00000 173 1.00000 174 1.00000 175 1.00000 176 1.00000 177 1.00000 178 1.00000 179 1.00000
 180 1.00000
```

```
THE QUEUE CHARACTERISTICS AT TIME N =   17

AT TIME N =  17 THE MEAN QUEUE LENGTH EQUALS      .9851

THE DISTRIBUTION OF THE QUEUE LENGTH AT TIME N =   17

   0  .67841   1  .75794   2  .82164   3  .89203   4  .94228   5  .96715   6  .98007   7  .98877   8  .99389   9  .99664
  10  .99816  11  .99902  12  .99948  13  .99973  14  .99986  15  .99993  16  .99996  17  .99998  18  .99999  19 1.00000
  20 1.00000



THE QUEUE CHARACTERISTICS AT TIME N =   18

AT TIME N =  18 THE MEAN QUEUE LENGTH EQUALS     1.0014

THE DISTRIBUTION OF THE QUEUE LENGTH AT TIME N =   18

   0  .67489   1  .75499   2  .81918   3  .88994   4  .94063   5  .96594   6  .97922   7  .98819   8  .99350   9  .99640
  10  .99801  11  .99892  12  .99943  13  .99970  14  .99984  15  .99992  16  .99996  17  .99998  18  .99999  19 1.00000
  20 1.00000



THE QUEUE CHARACTERISTICS AT TIME N =   19

AT TIME N =  19 THE MEAN QUEUE LENGTH EQUALS     1.0163

THE DISTRIBUTION OF THE QUEUE LENGTH AT TIME N =   19

   0  .67177   1  .75233   2  .81694   3  .88803   4  .93911   5  .96481   6  .97843   7  .98764   8  .99313   9  .99616
  10  .99786  11  .99883  12  .99937  13  .99966  14  .99982  15  .99991  16  .99995  17  .99998  18  .99999  19 1.00000
  20 1.00000
```

```
THE QUEUE CHARACTERISTICS AT TIME N =   20

AT TIME N =  20 THE MEAN QUEUE LENGTH EQUALS     1.0299

THE DISTRIBUTION OF THE QUEUE LENGTH AT TIME N =    20

   0  .66898   1  .74992   2  .81491   3  .88630   4  .93771   5  .96377   6  .97768   7  .98711   8  .99276   9  .99593
  10  .99771  11  .99874  12  .99932  13  .99963  14  .99980  15  .99990  16  .99995  17  .99997  18  .99999  19 1.00000
  20 1.00000

THE MEAN WAITINGTIME AT TIME N =  20 IS       2.6088

THE DISTRIBUTION OF THE WAITINGTIME AT TIME N =   20

   0  .66898   1  .70155   2  .73043   3  .75964   4  .78664   5  .81271   6  .83674   7  .85882   8  .87869   9  .89637
  10  .91189  11  .92555  12  .93735  13  .94747  14  .95604  15  .96326  16  .96934  17  .97445  18  .97873  19  .98230
  20  .98529  21  .98777  22  .98984  23  .99156  24  .99299  25  .99419  26  .99519  27  .99601  28  .99670  29  .99727
  30  .99775  31  .99814  32  .99847  33  .99874  34  .99896  35  .99915  36  .99930  37  .99942  38  .99953  39  .99961
  40  .99968  41  .99974  42  .99979  43  .99983  44  .99986  45  .99989  46  .99991  47  .99992  48  .99994  49  .99995
  50  .99996  51  .99997  52  .99997  53  .99998  54  .99998  55  .99999  56  .99999  57  .99999  58  .99999  59  .99999
  60 1.00000  61 1.00000  62 1.00000  63 1.00000  64 1.00000  65 1.00000  66 1.00000  67 1.00000  68 1.00000  69 1.00000
  70 1.00000  71 1.00000  72 1.00000  73 1.00000  74 1.00000  75 1.00000  76 1.00000  77 1.00000  78 1.00000  79 1.00000
  80 1.00000  81 1.00000  82 1.00000  83 1.00000  84 1.00000  85 1.00000  86 1.00000  87 1.00000  88 1.00000  89 1.00000
  90 1.00000  91 1.00000  92 1.00000  93 1.00000  94 1.00000  95 1.00000  96 1.00000  97 1.00000  98 1.00000  99 1.00000
 100 1.00000 101 1.00000 102 1.00000 103 1.00000 104 1.00000 105 1.00000 106 1.00000 107 1.00000 108 1.00000 109 1.00000
 110 1.00000 111 1.00000 112 1.00000 113 1.00000 114 1.00000 115 1.00000 116 1.00000 117 1.00000 118 1.00000 119 1.00000
 120 1.00000 121 1.00000 122 1.00000 123 1.00000 124 1.00000 125 1.00000 126 1.00000 127 1.00000 128 1.00000 129 1.00000
 130 1.00000 131 1.00000 132 1.00000 133 1.00000 134 1.00000 135 1.00000 136 1.00000 137 1.00000 138 1.00000 139 1.00000
 140 1.00000 141 1.00000 142 1.00000 143 1.00000 144 1.00000 145 1.00000 146 1.00000 147 1.00000 148 1.00000 149 1.00000
 150 1.00000 151 1.00000 152 1.00000 153 1.00000 154 1.00000 155 1.00000 156 1.00000 157 1.00000 158 1.00000 159 1.00000
 160 1.00000 161 1.00000 162 1.00000 163 1.00000 164 1.00000 165 1.00000 166 1.00000 167 1.00000 168 1.00000 169 1.00000
 170 1.00000 171 1.00000 172 1.00000 173 1.00000 174 1.00000 175 1.00000 176 1.00000 177 1.00000 178 1.00000 179 1.00000
 180 1.00000
```

270

REPRINTED FROM

NAVAL RESEARCH LOGISTICS QUARTERLY

JUNE 1973
VOL. 20, NO. 2

DEPARTMENT OF THE NAVY
OFFICE OF NAVAL RESEARCH

OFFICE OF NAVAL RESEARCH

NAVSO P-1278

THE SINGLE SERVER QUEUE IN DISCRETE TIME–NUMERICAL ANALYSIS I

Marcel F. Neuts*

Purdue University

ABSTRACT

This is the first of a sequence of papers dealing with the computational aspects of the transient behavior of queues in discrete time.

It is shown that for a substantial class of queues of practical interest, a wealth of numerical information may be obtained by relatively unsophisticated methods.

This approach should prove useful in the analysis of unstable queues which operate over a limited time interval, but is by no means limited to such queues.

Mathematically the service unit is modeled in terms of a multivariate Markov chain, whose particular structure is used in iterative computation. Many important queue features may then be derived from the n-step transition probabilities of this chain.

1. INTRODUCTION

The theory of queues, and more generally that of stochastic models, suffer from the insufficient development of the interface between structural-analytic results on one hand and directly applicable numerical methods on the other hand.

The practical queue analyst tends to use of the extensive theoretical work on service systems only those rare steady-state results which are analytically simple. This is often done with little regard for the mathematical assumptions underlying these results. Moreover such results commonly do not answer the real questions one is facing in the design of a service facility, and in rare cases the measures of performance based on steady-state assumptions may actually be misleading. An example of a stable queue with rare arrivals of large groups of customers in which this is the case, is discussed in the third paper in this sequence.

While simulation techniques are widely used, their implementation requires a thorough understanding of the probability structure of the queue as well as very substantial computing funds. In many instances of simulation studies familiar to this author, the structure of the queue was incorrectly or insufficiently used and exorbitant computing times were reported. One recognizes that simulation is often the only resort in studying a complex system. However, for those models whose structure is mathematically well understood, it is desirable that algorithms making use of the existing theory be developed. It is obviously intellectually pleasant to be able to use one's understanding of the mathematical structure to obtain numerical results by efficient algorithms. There are however also many practical reasons for investigating exact, rather than Monte Carlo algorithms, wherever possible. Since this is not the place to discuss these at length, we mention the study of relatively rare events as an example. To obtain a good estimate of the probability that a very stable queue exceeds a certain bound may require long simulation runs, because the simulated paths will only rarely exhibit the event

* This research was supported by the National Science Foundation, grant GP 28650.

of interest. Queues which are nearly critical are also difficult to simulate, because of the substantial stochastic variability of the queue length and the related quantities. For these again, exact algorithms prove to be a useful alternative method of analysis.

The time involved in the preparation and testing of an exact algorithm is probably much greater than in a corresponding simulation study. This aspect must be taken into account in comparisons of the cost and the effectiveness of both approaches. For this reason only models of wide potential applicability should initially be studied by means of exact algorithms. There is furthermore a return which has particular value to the theoretical queue analyst. An algorithm is similar to a theorem in that its applicability is usually far greater than the original problem from which it sprung. As is the case for theorems, algorithms are also interesting because of their limitations. When the latter are recognized, they usually stimulate many questions related to approximations, algorithmic efficiency or structural theorems.

The purpose of this paper and the subsequent ones is to investigate a useful single server queue in detail. Before we consider the model specifically, we first discuss some difficulties and desiderata related to the numerical analysis of a much wider class of queueing problems.

a. The Finiteness of the Waiting Line

In reality, unbounded queues do not exist. The unbounded queue is strictly a mathematically convenient abstraction. By removing one "boundary" of the queue length process one obtains simpler stochastic processes. It further becomes possible to give an elegant treatment of the intuitive quality of stability of the queue.

In practical situations, there is either a finite waiting room, usually with loss of those customers who find the waiting space full, or else the buildup of the queue beyond a certain limit creates utter chaos and the object of the study is precisely the design of a sufficiently fast service unit or a sufficiently large waiting room to make this a very rare event.

Throughout this paper we shall limit our attention to bounded queues. In the numerical examples considered the largest value of the maximum queue length L_1 was, at most, 100.

b. Transient versus Steady-state Behavior

There are very few results on the transient behavior of queues, which are analytically explicit. Even the latter are nearly all ill-suited for direct numerical analysis. An interesting discussion of a simple transient queue and the difficulties of its computational analysis may be found in Leese and Boyd [1].

The steady-state probabilities of some simple queues are the only ones available in books written for the applied worker. Their relevance to the concrete problems is often limited; they clearly have no bearing whatsoever on the solution of unstable queues. Moreover, since the limiting process by which they are obtained has an averaging (or mixing) property, such results convey no information on the fluctuations of the queue length and the waiting times. Ignoring such fluctuations in a design may have catastrophic results.

In recent years the study of weak convergence properties of queueing processes and the resulting diffusion approximations have shed a new light on service systems of which the macroscopic time-behavior rather than the short-range fluctuations is the most important feature. This promising approach is mathematically fairly sophisticated and has not yet undergone sufficient investigation from the viewpoint of computation.

Our concern in this paper is, in a sense, with the small-scale service system whose short-range behavior is important. Therefore the models studied here are unlikely to be well approximated by a diffusion process. Nevertheless a comparison of both the direct solution and a diffusion approximation method with regards to accuracy and computing time is of interest, but will not be undertaken here.

c. Continuous versus Discrete Parameter Models

It is known that finite $M \mid G \mid 1$ and $GI \mid M \mid 1$ queues may be conveniently studied in relation to an imbedded Markov renewal process with a finite number of states. The transient behavior of such queues, as well as many related ones, may be computed in principle in terms of the successive matrix-convolution products of the transition matrix of this imbedded process. If a queue with a waiting room of size L_1 is investigated, each such a matrix-convolution product may require as many as $(L_1+1)^2$ evaluations of the convolution product of two functions. To perform this operation accurately is time-consuming, so that the computer implementation of this analysis is likely to result in considerable computing time.

In an earlier study of the single server queue in discrete time, Dafermos and Neuts [2] argued at length for the advantages of analyzing many queues in terms of a discrete time parameter. These arguments will not be repeated here. From the viewpoint of numerical analysis the most obvious advantages of a discrete time model are:

a. The ease with which one or more supplementary variables may be introduced so as to imbed the queueing process in a multivariate Markov process.

b. The fact that convolution products of sequences of numbers may be computed with much greater ease than those of functions of a continuous real variable may be evaluated accurately.

In most cases of practical interest one may discern an elementary unit of time natural to the particular queue. Many queueing analysts nevertheless insists on thinking of discrete models as approximations to continuous ones. This insistence, which may usually be traced to the prevailing attitude in applied mathematics before the advent of the computer, has some appeal for its mathematical elegance particularly where methods of analysis may be used. From a computational viewpoint, continuous parameter models are often substantially more delicate to analyze and this without yielding additional insight into the real process which is being modeled.

d. Parametric versus General Distributions

The insistence on specific parametric families of probability distributions (such as the gamma family) in stochastic models is also largely a holdover from the pre-computer era. Where a parametric assumption has an important structural consequence (e.g., the memory-less property of the negative exponential distribution) one should be very aware of this. However, in cases where the parametric assumption yields only marginal simplifications, both the theoretical analysis and the computational methods should ignore it altogether. As a case in point, the $M|E_k|1$ queue for $k > 1$ is only in some details easier to discuss than the $M|G|1$ queue. There is therefore little point in a special numerical method for the former which does not also include the latter.

In the sequel of this discussion, we shall therefore only stress the structural assumptions that are needed. Such items as service time distributions will be as general as possible.

In most specific models one may assume without loss of generality that service times and related random variables are bounded lattice random variables. This assumption, essential to our approach, is also the one which limits its range of applicability most. Our approach is not well suited to queues

in which both very short and very long jobs may arrive. Different methods of numerical analysis are needed for these. It is commonly so that a given numerical method is well suited for a certain range of problems, but fails for similar problems outside this range. The assumptions which limit the applicability of the present model are discussed in the appropriate places in the sequel.

2. THE ASSUMPTIONS OF THE MODEL

a. The Arrivals

We consider a single server queue in discrete time with a maximum queuelength L_1. The elementary time interval is chosen as our time unit. We assume that the numbers of arrivals during the successive unit time intervals are independent, identically distributed random variables. Furthermore p_ν, $\nu = 0, 1, \ldots, K$, is the probability that ν customers join the queue during a given unit time interval. ($p_0 + p_1 + \ldots + p_k = 1$). In this discussion, and in the related FORTRAN program, we shall insist that $1 \leqslant K < L_1$. The restriction $K < L_1$ is not essential and is usually satisfied in practice. Its removal requires minor modifications in the analysis and in the program.

We further assume that the p_ν are independent of time, but with minor obvious changes the recurrence relations are valid also for queues in which the arrival probabilities vary with time.

b. The Service Times

We assume that the service times of the successive customers are independent, identically distributed (integer-valued), random variables with values in the set $\{1, 2, \ldots, L_2\}$. We denote by r_ν, $\nu = 1, \ldots, L_2$, the probability that a customer requires ν units of service time. The values $L_1 = 100$ and $L_2 = 100$ appear to be practical upper limits to the computer implementation of the method suggested here.

Our assumption that every customer requires at least one unit of service time may easily be removed, but is satisfied in most all concrete applications. It suffices to introduce a quantity r_0 that a service time is equal to zero and to modify the recurrence relations accordingly.

As pointed out below, it is also easy to modify our analysis to include the case where the service time density depends on the time at which the service is initiated.

c. The Queue Discipline

Except in discussing the waiting times, the order of service is immaterial. The waiting times will be discussed for the first-come, first-served discipline.

To settle the issue of simultaneous arrivals and beginnings of services, we assume that all arrivals in $[n-1, n)$ are added to the queue at time $n-0$. If a service starts at time n and requires ν units of time, we shall consider it to start at time n and to end at time $n + \nu - 0$.

Only as many arrivals as to maintain the queue length less than or equal to L_1 are accepted. Any excessive customers are assumed to be permanently lost.

d. The Initial Conditions

We assume that at time $n = 0$, there are i_0 customers present. $0 \leqslant i_0 \leqslant L_1$. If $i_0 \geqslant 1$, the customer in service at time $n = 0$ requires j_0, $1 \leqslant j_0 \leqslant L_2$ additional units of service time. We make the convention that if $i_0 = 0$, then $j_0 = 0$, and conversely.

3. THE MARKOV CHAIN

We denote by X_n the queue length at time n and by Y_n the number of additional units of service time required by the customer in service at time n. We make the convention that $X_n = 0$ if and only if $Y_n = 0$.

It follows readily from our assumptions that the bivariate sequence $\{(X_n, Y_n), n \geqslant 0\}$ is a Markov chain with state space consisting of the point $(0, 0)$ and all points $(i, j), i=1, \ldots, L_1; j=1, \ldots, L_2$, and with initial state (i_0, j).

The transition probability matrix of the Markov chain is easily written down. We shall not do so since the recurrence relations make judicious use of its special structure.

For fixed i_0 and j_0, we define the conditional probabilities

$$(1) \qquad P_n(i, j) = P\{X_n = i, Y_n = j \mid X_0 = i_0, Y_0 = j_0\}.$$

The probabilities $P_n(i, j)$ satisfy the following recurrence relations in n for all $n \geqslant 0$:

$$(2)\text{(a)} \quad P_{n+1}(0, 0) = p_0[P_n(0, 0) + P_n(1, 1)],$$

$$\text{(b)} \quad P_{n+1}(i, j) = p_0 P_n(i, j+1) + \sum_{\nu=1}^{i-1} p_{i-\nu} P_n(\nu, j+1) + r_j \Big\{ p_i P_n(0, 0) + p_0 P_n(i+1, 1) + \sum_{\nu=1}^{i} p_{i-\nu+1} P_n(\nu, 1) \Big\},$$

for $i=1, \ldots, K$ and $j=1, \ldots, L_2-1$.

$$\text{(c)} \quad P_{n+1}(i, j) = p_0 P_n(i, j+1) + \sum_{\nu=i-K}^{i-1} p_{i-\nu} P_n(\nu, j+1) + r_j \Big\{ p_0 P_n(i+1, 1) + \sum_{\nu=i-K+1}^{i} p_{i-\nu+1} P_n(\nu, 1) \Big\},$$

for $i=K+1, \ldots, L_1-1$ and $j=1, \ldots, L_2-1$.

$$\text{(d)} \quad P_{n+1}(L_1, j) = P_n(L_1, j+1) + \sum_{\nu=1}^{K} \Big(1 - \sum_{k=0}^{\nu-1} p_k\Big) P_n(L_1-\nu, j+1) + r_j \Big\{ \sum_{\nu=1}^{K} \Big(1 - \sum_{k=0}^{\nu-1} p_k\Big) P_n(L_1-\nu+1, 1) \Big\}$$

for $j=1, \ldots, L_2-1$.

$$\text{(e)} \quad P_{n+1}(i, L_2) = r_{L_2} \Big\{ p_i P_n(0, 0) + p_0 P_n(i+1, 1) + \sum_{\nu=1}^{i} p_{i-\nu+1} P_n(\nu, 1) \Big\},$$

for $i=1, \ldots, K$.

$$\text{(f)} \quad P_{n+1}(i, L_2) = r_{L_2} \Big\{ p_0 P_n(i+1, 1) + \sum_{\nu=i-K+1}^{i} p_{i-\nu+1} P_n(\nu, 1) \Big\},$$

for $i=K+1, \ldots, L_1-1$.

$$\text{(g)} \quad P_{n+1}(L_1, L_2) = r_{L_2} \Big\{ \sum_{\nu=1}^{K} \Big(1 - \sum_{k=0}^{\nu-1} p_k\Big) P_n(L_1-\nu+1, 1) \Big\} \cdot$$

The recurrence is initialized by setting $P_0(i_0, j_0) = 1$ and $P_0(i, j) = 0$ for all other pairs (i, j). If the distribution of the random variables X_0, Y_0 is given rather than exact values, this simply amounts to a different definition of the initial array $P_0(i, j)$.

We note that the expressions in curly brackets in the formulas $(2\ b - g)$ depend on the first index only. This term corresponds to the case where the instant $n + 1$ is the beginning of a new service. This simplifying feature is used in the organization of the computer program. If the service time distribution depends on the instant of initiation of a service, only the factor r_j in the recurrence relations are affected.

The analogous, but simpler recurrence relations for the unbounded queue were examined analytically by Dafermos and Neuts [2].

The recurrence relations (2) are well suited for iterative computation. The author organized his computation so as to have at the nth iteration only the quantities $P_n(0, 0)$ and $P_n(i, j)$, $i = 1, \ldots, L_1$; $j = 1, \ldots, L_2$ in memory.

The quantities $P_n(0, 0)$ and $P_n(i, j)$, $i = 1, \ldots, L_1$; $j = 1, \ldots, L_2$ make up the (conditional) joint density of the queue length X_n and the residual service time Y_n at time n. This joint density is not, in itself, of great practical use; however from it the distribution (or the density) of the queue length and the waiting time at time n can be easily obtained. These "derived" queue features are considered next.

4. DERIVED QUEUE FEATURES

a. The Queue Length Distribution

The probability that the queue length at time n is zero, is given by $P_n(0, 0)$. For $i = 1, \ldots, L_1$, we have

$$P\{X_n = i \mid X_0 = i_o, Y = j_0\} = \sum_{j=1}^{L_2} P_n(i, j). \tag{3}$$

Lower order moments of the queue length at time n may be obtained routinely.

b. The Waiting Time Distribution

The waiting time at time n is defined to be zero if, and only if $X_n = Y_n = 0$. For $X_n = 1$, $Y_n = j$, the waiting time equals j. For $X_n > 1$, $Y_n = j$ the waiting time is the sum of j and $X_n - 1$ independent service times. The waiting time is therefore an integer-valued random variable with value between 0 and L_1L_2.

The density WT_ν, $\nu = 0, \ldots, L_1L_2$ is obtained as follows. Clearly $WT_0 = P_n(0, 0)$. The quantities WT_n for $1 \leq \nu \leq L_1L_2$ were obtained by evaluating the following convolution-polynomial.

Let $WT(\cdot)$ denote the density $\{WT_\nu,\ 1 \leq \nu \leq L_1L_2\}$ and let $R(\cdot)$ be the service time density $\{r_j,\ 1 \leq j \leq L_2\}$. Finally let $P_n(i, \cdot)$ be the density $\{P_n(i, j),\ 1 \leq j \leq L_2\}$ for $i = 1, \ldots, L_1$, then $WT(\cdot)$ is given by

$$WT(\cdot) = P_n(1, \cdot) + P_n(2, \cdot) * R(\cdot) + \ldots + P_n(L_1, \cdot) * R^{(L_1-1)}(\cdot), \tag{4}$$

where $R^{(k)}(\cdot)$ is the k-fold convolution of $R(\cdot)$ with itself.

The density $WT(\cdot)$ may be computed for each n by one of two procedures. Either the convolutions $R^{(k)}(\cdot)$, $k = 1, \ldots, L_1 - 1$ may be computed once and for all and only the convolutions with the arrays $P_n(i, \cdot)$ need to be computed at those time points n at which the density of the waiting time is

desired. This procedure results in a substantial increase in the central memory storage required by the program.

Alternatively, the density $WT(\cdot)$ may be computed by a convolution analogue of *Horner's algorithm* for the numerical evaluation of ordinary polynomials. This procedure consists in the successive evaluation of the sequences

$$WT^{(1)}(\cdot)=P_n(L_1,\cdot), \tag{5}$$

$$WT^{(k)}(\cdot)=WT^{(k-1)}(\cdot)*R(\cdot)+P_n(L_1-k+1,\cdot),$$

for $k=1,\ldots,L_1$. The sequence $WT^{(L_1)}(\cdot)$ is the desired sequence $WT(\cdot)$.

The latter procedure does not result in the use of core storage for intermediate quantities. For purposes of comparison and as a guard against rounding errors, both procedures were programmed and tested on large scale examples. The Horner convolution algorithm performed slightly better in all examples and no evidence of rounding errors were found. Its much smaller core requirements make it the more efficient of the two procedures.

5. REPORT ON COMPUTATIONAL TRIALS

A FORTRAN IV program was written by the author and tested on a variety of cases on the CDC 6500 at Purdue University. Even in the largest examples no evidence of rounding errors was found, even though all probabilities were printed to five decimal places and all computations were performed in single precision.

The full output consisted, in addition to the summary of the input data, of the following:

(a) the mean of the queue length,
(b) the cumulative distribution of the queue length,
(c) the mean waiting time,
(d) the cumulative distribution of the waiting time,
(e) the joint density of the queue length X_n and the residual service time Y_n.

All these for all values of n up to some upper value to be specified.

Since the full output is very voluminous and contains much more information than may be needed in the analysis of a given queue, options were written into the program which permit the deletion of the items (c), (d), or (e) from the printed output. A further option was created which permits the computation of the waiting time distribution to be performed at certain specified time points only.

No systematic study of the processing time was made. For smaller examples the computation times were generally below 10 seconds. The following computation times for some larger examples are given for purposes of illustration only.

$L_1=30$	(max. queue length)
$L_2=6$	(max. duration of one service)
$K=2$	(max. number of arrivals)
$NNN=250$	(number of time points computed)

In the following CP is the processing time in seconds and LL is the number of lines of output, including the program listing.

a. full printed output

$$CP = 215.159 \qquad LL = 19265$$

b. the queue length and waiting time distributions only.

$$CP = 157.860 \qquad LL = 10656$$

c. the queue length distribution for all time points and the waiting time distribution only for time points which are multiples of 25.

$$CP = 23.763 \qquad LL = 4916$$

The processing time for a large scale iteration of the type needed here can be substantially decreased by writing a program in an assembly language, rather than in FORTRAN. A reduction by 50 or 60 percent can realistically be expected.

For large problems, the computation of the waiting time distribution is by far the most time consuming part of the algorithm. For very stable queues, we recommend neglecting the higher order terms in the convolution polynomial (4). These terms contribute very little, except to the extreme upper tail of the distribution, but add very considerably to the processing time. For queues with a very rapid build-up, some savings may be accomplished by neglecting lower order terms in (4), but this is less significant. For queues which are near-critical, the waiting time algorithm ceases to be practical for queues with $L_1 > 200$, because of the large numbers of operations involved. The problem of finding good numerical approximations to the distribution of the waiting time in this case is challenging and needs further investigation.

6. CONCLUSIONS

By the use of only the most elementary structural properties, we have shown that the transient behavior of a substantial class of single server queues may be analyzed numerically. The approach presented here should prove itself to be useful in studying the build-up of unstable queues and the fluctuations of queues at traffic lights, highway merging ramps, service counters in public offices and retail outlets and many others.

This approach is well suited for many queueing processes which do not lend themselves to diffusion approximation methods. The amount of computing time used in typical examples also indicates a very substantial saving over that needed to analyze similar models by simulation methods.

Further work is currently being done to extend the applicability of this approach to longer queues and to much longer time periods. This extension, however, requires the use of mathematically more sophisticated properties of the queueing process.

A version of this paper, containing a program listing and a specific numerical example, is available as a technical report. It may be obtained from the author upon request, by writing to Professor Marcel F. Neuts, Department of Statistics, Purdue University, West Lafayette, IN. 47907.

BIBLIOGRAPHY

[1] Leese, E. L. and D. W. Boyd, "Numerical Methods of Determining the Transient Behavior of Queues with Variable Arrival Rates," J. Canadian Operations Res. Soc. *4*, 1–13 (1966).

[2] Dafermos, S. and M. F. Neuts, "A Single Server in Discrete Time," Cahiers du Centre de Recherche Opérationnelle, *13*, 23–40 (1971).