

AD-A136 580

RIA-84-U59

AD A-136 580

DRSMC/SA/MR-4

TECHNICAL
LIBRARY

**MODELS OF A SERVICE SYSTEM
FOR
PRODUCTION MACHINE MAINTENANCE**

GEORGE J. SCHLENKER

NOVEMBER 1983

Distribution Statement

Approved for public release; distribution unlimited.



**US ARMY ARMAMENT, MUNITIONS AND CHEMICAL COMMAND
SYSTEMS AND OPERATIONS ANALYSIS DIRECTORATE
ROCK ISLAND, ILLINOIS 61299**

DISPOSITION

Destroy this report when no longer needed. Do not return it to the originator.

DISCLAIMER

The findings in this report are not to be construed as an official Department of the Army position.

WARNING

Information and data contained in this document are based on input available at the time of preparation. Because the results may be subject to change, this document should not be construed to represent the official position of the US Army Materiel Development & Readiness Command unless so stated.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER DRSMC/SA/MR-4	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Models of a Service System for Production Machine Maintenance		5. TYPE OF REPORT & PERIOD COVERED Report-Final
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) George J. Schlenker		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS US Army Armament, Munitions and Chemical Command Systems and Operations Analysis Directorate Rock Island, IL 61299		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS US Army Armament, Munitions and Chemical Command Systems and Operations Analysis Directorate Rock Island, IL 61299		12. REPORT DATE November 1983
		13. NUMBER OF PAGES 95
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES Additions to/from the DISTRIBUTION LIST are invited and should be forwarded to the following address: Commander, US Army Armament, Munitions and Chemical Command, ATTN: DRSMC-SAS (R) Rock Island, IL 61299 AUTOVON 793-5041/6370		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Operations research Maintenance systems Markov processes Manufacturing operations Queueing theory Industrial operations Service systems Numerical analysis		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) The machine repair and adjustment process is one of the elements which characterizes the operations of a production system. In many cases this maintenance system can be described adequately by analytic models. This report derives some mathematical models which depict the steady-state perfor- mance of this service system. All the models are steady-state Markov models of a service system having a finite customer population with multiple, independent servers, each having a service time which is Erlang-distributed.		

20. ABSTRACT (Cont'd)

These models have been useful in verifying the maintenance portion of a stochastic production simulation. Some numerical results from the implementing computer program are presented. Results of different models are compared. Computer programs are provided.

DRSMC/SA/MR-4

MODELS OF A SERVICE SYSTEM
FOR PRODUCTION MACHINE MAINTENANCE

GEORGE J. SCHLENKER

November 1983

ACKNOWLEDGEMENT

The author gratefully acknowledges the typing of this report by Mrs. Deborah Marland. Her assistance on this and other technical reports is certainly appreciated.

CONTENTS

	Page
Background	1
Objectives	2
Scope	2
Birth-Death Processes	3
Definition of States for a Single-Server System	4
State Transitions (Single Server)	4
Kolmogorov Equations for a One-Server System	6
Statistical Properties of the System	9
Conditional State Probability Distribution	10
Mean and Variance of Conditional Waiting Time in Queue	13
Multiple Servers with Exponential Service Times	14
Distribution of Time in Queue (Exponential Services)	15
Numerical Results	17
Comparisons Between Models With Finite Customers	22
Conclusions	29
Recapitulation and Survey of Other Derivations	30
Two Servers With Erlang(2) Service	31
Kolmogorov Equations for a Two-Server System	34
Steady-State Equations (Two Servers)	35
Statistics for the Two-Server System	38
Three Servers With Erlang(2) Service	41
Steady-State Equations (Three Servers)	45
Statistics for the Three-Server System	47
References	48
Distribution List	49
Annex A. Computer Program Listings	A-1
Annex B. Heterogeneous Machine Population	B-1

MEMORANDUM REPORT

SUBJECT: Models of a Service System for Production Machine Maintenance

1. Reference:

References are designated by bracketed numbers and are included in the footnotes. The references are also listed.

2. Background

This memorandum reports on work associated with a larger study [1], concerned with the productivity of manufacturing systems. A stochastic simulation, called TANDEMT, was developed for the Ref [1] study. Part of TANDEMT characterizes a maintenance system for adjusting and repairing machines on a production line. To verify this part of the simulation, analytic studies of service systems were performed. The models in this study draw from and extend classical queueing theory, such as in Ref [2]. Whereas the analytic models were used to verify a simulation, their use is not limited to that purpose. It has been found that one can forego the simulation and use only analytic models in designing a machine maintenance system*. Recognizing

[1] DD1498, HQ, US ARRCOM, DRSAR-SA, March 1983, title: Manufacturing Productivity Study.

[2] Gross, Donald and Harris, Carl, Fundamentals of Queueing Theory, John Wiley and Sons, c. 1974.

* The analytic models are included in a subroutine which can be called from TANDEMT. This arrangement is convenient since these programs share input data. In using the routine for system design, the minimum number of repairmen is selected which satisfies constraints on system performance measures.

the general* importance of analytic models of this sort, I have chosen to present certain derivations and results of interest here.

3. Objectives

One purpose of this memorandum is to derive some results which may be useful in modeling a maintenance system which serves a finite population of machines. Another objective is to present parametric studies using these models. A comparison of results of alternative models will also be made. Of interest to analysts are some numerical approximations.

4. Scope

In the language of queueing theory the (finite) population of machines are "customers" and the repairmen who perform maintenance are the "servers". Similarly, the failures of machines are regarded as demands for service, which if unmet are placed in a "service queue". I will use these terms to suggest that these models are not limited to a production setting. The "system" of interest here is the machine maintenance system, which is characterized by the number of repairmen (or servers) and by the probability distribution of service time. These two factors are the distinguishing elements between models in this memorandum. Note that no distinction is made between types of repairmen. For this application any repairman can render the requisite service. These repairmen are not specialists. Further, they are assumed to work as individuals, rather than in teams.

* Other applications of the models considered here are manifold. Finite queueing models have been used in describing the process by which artillery targets are detected and "served" and in describing the FM communications process in tactical fire control nets.

5. Ordinarily, the random arrival of a "demand" for service is regarded as occurring at a rate equal to the product of a unit rate times the number of customers outside* the service system. The time-to-next-demand is conditionally an exponential random variable. For most mature manufacturing systems operating in steady-state, this is a valid description of demand for maintenance. This demand description is used in all models in this report. Actual maintenance experience often shows that the time to complete a maintenance action (time to repair) is a lognormal random variable. To describe a lognormally distributed event in terms of Markov processes involves approximation. For service times which have a coefficient of variation near unity, an exponential approximation is used. However, if the standard deviation of service time is significantly smaller than the mean--as is often the case--an Erlang distribution can be used to describe the service time. The Erlang distribution, or gamma distribution with integer shape parameter p can be represented as the distribution of a sum of p random variables each of which is exponential. For example, in the case of an Erlang distribution with shape parameter 2 and mean $\bar{x}$, an equivalent stochastic model consists of two serial elements both of which are exponential with common mean $\bar{x}/2$. Generally, the coefficient of variation of a gamma distribution with shape parameter p is $1/\sqrt{p}$. Thus, if the standard deviation of service time is about 0.7 times the mean, an Erlang distribution with shape parameter 2 would be an appropriate statistical model. The models considered here have either exponential or gamma(2) service times.

6. Birth-Death Processes

Consider a service system having a single server whose service times are gamma(2) i.e., are random variables from a gamma probability distribution with shape parameter 2. This is sometimes written $\Gamma(2)$. Let the customer population be m machines. Thus, at most m machines may reside in the maintenance system with at most $m-1$ in the queue. Machine failures occur at inter-event times which are exponential with a rate $(m-k)\lambda$, where k machines are in the system and where

* The term "inside the system" refers to residence in either the service queue or in a service channel being served. After being served a customer leaves the system and is then "outside the system".

the unit rate λ is the unit rate of entry or "birth" rate for a machine into the system. Notationally, let MTBF be the mean time between failures for the population of machines.* Then,

$$\lambda = 1/\text{MTBF}. \quad (1)$$

To model the gamma(2) service times, two stages of service are assumed, each of which is exponentially distributed with rate parameter μ . After the second stage of service is completed the machine leaves the maintenance system. Thus, if μ_1 is the rate of exodus, or "death" rate, from the system

$$\mu = 2\mu_1. \quad (2)$$

With a mean time to repair MTTR,

$$\mu_1 = 1/\text{MTTR}. \quad (3)$$

7. Definitions of States for a Single-Server System

To characterize the possible states of the system, define and number event E_j as follows:

$$E_j = 2j-1, \quad j \geq 1, \quad (4a)$$

with $j-1$ customers in the queue, where the customer being served is in the first stage of service; and

$$E_j = 2j, \quad (4b)$$

with $j-1$ customers in the queue, where the customer being served is in the second stage of service. E_0 denotes the null state. Notationally, let

$$p_n(t) = P\{\text{system is in state } E_n \text{ at time } t\}, \quad 0 \leq n \leq 2m. \quad (5)$$

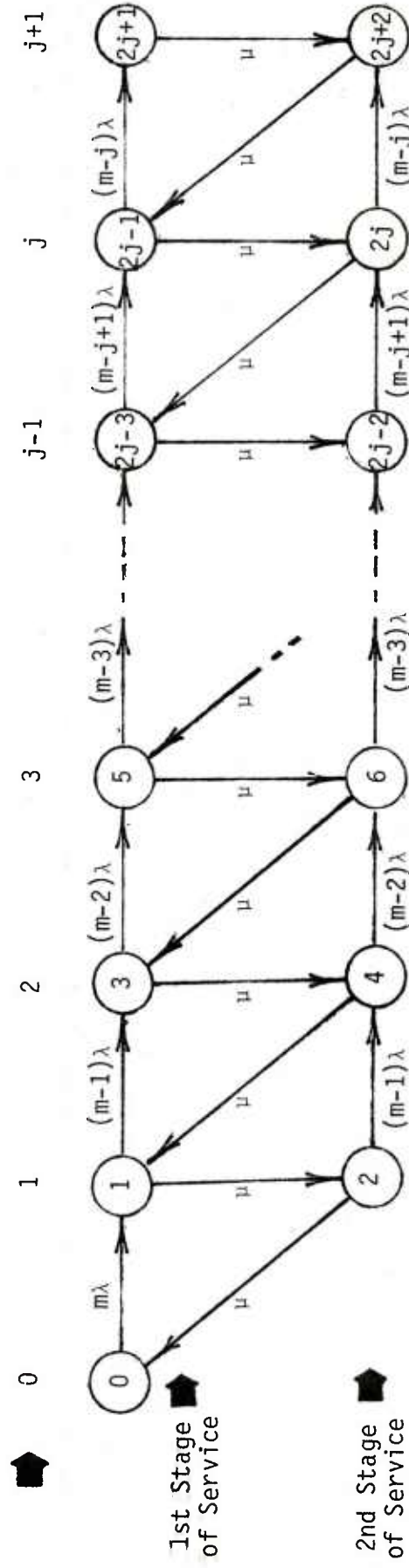
The object of the following derivation is to show how $p_n(t)$ is obtained, for the admissible values of n , when time becomes large, i.e., when the system is in stochastic steady state.

8. State Transitions (Single Server)

It is helpful to examine the ways in which the system can transition between states. Figure 1 displays the states as nodes and the transitions as directed arcs. This state transition diagram also shows the Markov rate constant for

* See Annex B for treatment of a heterogeneous population of machines.

Customers in the System



Definition of States

State E_j is defined by the number of customers in the system and by the stage of service as follows:

$$E_j = 2j-1 \text{ with } j \text{ customers in the system and with the customer being attended in the 1st stage of service.}$$

$$= 2j \text{ with } j \text{ customers in the system and with the customer being attended in the 2nd stage of service,}$$

$$1 \leq j \leq m.$$

Figure 1. State Transition Diagram for a Single-Server Service System With Erlang (2) Service Times

each indicated transition as a label on the arc. Thus, the conditional probability of transition from state 0 to state 1 in an incremental time interval h is $m\lambda h$. Similarly, given the system is in state 1, a transition to state 2 in time increment h occurs with probability μh , and a transition to state 3 in time increment h occurs with probability $(m-1)\lambda h$. Following the standard procedure for Markov processes, one writes the probability of being in state n at time $t+h$ in terms of the probabilities of occupying all of the states at time t . For example,

$$\begin{aligned} P\{E_0 \text{ at } t+h\} &= P\{E_0 \text{ at } t\} \cdot \\ &P\{\text{no transition in } (t, t+h)\} \\ &+ P\{E_2 \text{ at } t\}P\{\text{trans. from } E_2 \text{ to } E_0 \text{ in } (t, t+h)\} . \end{aligned} \quad (6)$$

In our notation,

$$p_0(t+h) = p_0(t) (1 - m\lambda h) + p_2(t)\mu h . \quad (7)$$

Then,

$$\frac{1}{h} [p_0(t+h) - p_0(t)] = -m\lambda p_0(t) + \mu p_2(t) . \quad (8)$$

Taking the limit as h approaches zero yields

$$\dot{p}_0(t) = -m\lambda p_0(t) + \mu p_2(t) . \quad (9)$$

Note that this result can also be obtained by inspection from the state transition diagram. To form the general right hand side: All rates from the k th node are summed; and, the negative of this sum is the coefficient of p_k . All other terms of the r.h.s. have positive coefficients corresponding to the rates on arcs entering the k th node. The j th coefficient multiplies $p_j(t)$, where j is a node leading directly into the k th node.

9. Kolmogorov Equations for a One-Server System

Following the above procedure for all states gives rise to the differential-difference (Kolmogorov) equations which describe this system. Suppressing the notation for time dependence, one has:

$$\dot{p}_0 = -m\lambda p_0 + \mu p_2 .$$

$$\dot{p}_1 = m\lambda p_0 - (\mu + (m-1)\lambda)p_1 + \mu p_4 .$$

For $2 \leq j \leq m-1$,

$$\dot{p}_{2j-1} = (m-j+1)\lambda p_{2j-3} - (\mu + (m-j)\lambda)p_{2j-1} + \mu p_{2j+2}$$

$$\dot{p}_{2j} = (m-j+1)\lambda p_{2j-2} + \mu p_{2j-1} - (\mu + (m-j)\lambda)p_{2j} .$$

And, for the last two states,

$$\dot{p}_{2m-1} = \lambda p_{2m-3} - \mu p_{2m-1}$$

$$\dot{p}_{2m} = \lambda p_{2m-2} + \mu p_{2m-1} - \mu p_{2m} . \quad (10)$$

In stochastic steady state, the time derivatives of all probabilities are zero. Setting dp_k/dt to zero for all k in (10) yields a matrix-vector equation for the state probability vector $\underline{p}$, with

$$\underline{p}' = (p_1, p_2, \dots, p_{2m}) .$$

The $2m+1$ equations in the set (10) contain one redundant equation since the probabilities are related by

$$p_0 = 1 - \sum_{j=1}^{2m} p_j . \quad (11)$$

One of the equations can be deleted--say, the first--and (11) substituted for p_0 in the second equation (the only other place in which p_0 appears). This process yields

$$-m\lambda \sum_{j=1}^{2m} p_j - (\mu + (m-1)\lambda)p_1 + \mu p_3 = -m\lambda$$

$$(m-1)\lambda p_1 - (\mu + (m-2)\lambda)p_3 + \mu p_5 = 0$$

$$(m-1)\lambda p_2 + \mu p_3 - (\mu + (m-2)\lambda)p_4 = 0 ,$$

and so forth.

The equivalent matrix-vector formulation is

$$A \underline{p} = \underline{b} \quad (12)$$

with A ($2m \times 2m$) and $\underline{p}$ and $\underline{b}$ ($2m \times 1$), where

$$\underline{b}' = (-m\lambda, 0, 0, \dots, 0) , \quad (12b)$$

and

$$A = \{a_{ij}\} . \quad (12c)$$

The elements of A are:

$$a_{11} = -m\lambda - (\mu + (m-1)\lambda)$$

$$a_{1j} = -m\lambda, \quad 2 \leq j \neq 4 \leq 2m$$

$$a_{14} = \mu - m\lambda$$

$$a_{21} = \mu$$

$$a_{22} = -(\mu + (m-1)\lambda)$$

$$a_{2j} = 0, \quad j > 2.$$

For the remaining non-zero terms

$$\text{of } \{a_{ij}\}, \quad 2 \leq k \leq m-1,$$

$$a_{2k-1, 2k-3} = (m-k+1)\lambda$$

$$a_{2k-1, 2k-1} = -(\mu + (m-k)\lambda)$$

$$a_{2k-1, 2k+2} = \mu$$

$$a_{2k, 2k-2} = (m-k+1)\lambda$$

$$a_{2k, 2k-1} = \mu$$

$$a_{2k, 2k} = -(\mu + (m-k)\lambda);$$

$$a_{2m-1, 2m-3} = \lambda$$

$$a_{2m-1, 2m-1} = -\mu$$

$$a_{2m, 2m-2} = \lambda$$

$$a_{2m, 2m-1} = \mu$$

$$a_{2m, 2m} = -\mu.$$

(13)

Since A is of full rank, $\underline{p}$ may be obtained from

$$\underline{p} = A^{-1} \underline{b} . \quad (14)$$

The probability of the null state is obtained from (11).

Then, the probability of the maintenance system having k customers in steady state is given by

$$\pi_k = p_{2k-1} + p_{2k} , \quad 1 \leq k \leq m . \quad (15)$$

10. Statistical Properties of the System

From this probability distribution one obtains the mean and variance* of the number of customers in the system.

$$E[N_{sys}] \equiv \bar{N}_{sys} = \sum_{k=1}^m k \pi_k \quad (16)$$

$$V(N_{sys}) = \sum_{k=1}^m k^2 \pi_k - (\bar{N}_{sys})^2 . \quad (17)$$

The (mathematically) expected number of customers in the queue ($\bar{N}_q$) is obtained from

$$E[N_q] = \sum_{k=2}^m (k-1) \pi_k$$

or

$$E[N_q] = \bar{N}_{sys} - (1-p_0) \equiv \bar{N}_q \quad (18)$$

The variance of the number in the queue

$$V[N_q] = \sum_{k=2}^m (k-1)^2 \pi_k - (\bar{N}_q)^2 . \quad (19)$$

Related properties are the mean time a customer spends in the system (W) and the mean time spent in the queue (W_q). For all service systems the value of W is greater than W_q by the mean service time. In this case,

$$W = W_q + 1/\mu_1 . \quad (20)$$

* Operator notation is used in displaying the mean and variance. The expectation operator is denoted $E[\cdot]$, and the variance operator $V[\cdot]$.

Given W , equation (20) is used to obtain W_q .

With exponential arrivals and a finite customer population the mean arrival rate is $\lambda(m - \bar{N}_{sys})$. Using this result in Little's formula (p. 120, [2]) yields

$$\bar{N}_{sys} = \lambda(m - \bar{N}_{sys})W ,$$

from which

$$W = \bar{N}_{sys} / [\lambda(m - \bar{N}_{sys})] . \quad (21)$$

Then, W_q can be obtained from (20). Alternatively, W_q is given by another version of Little's formula:

$$W_q = \bar{N}_q / [\lambda(m - \bar{N}_{sys})] . \quad (22)$$

11. Conditional State Probability Distribution

The following derivation is to obtain the probability distributions of waiting time in queue and in the system, given an arriving customer must queue. In doing this, we first obtain the conditional probability distribution that the system is in the n th state, given an arrival. Denote, q_n as the (steady-state) probability that an arrival finds the system in the n th state. From Bayes' theorem, q_n is proportional to the product of two probabilities: (a) $P\{\text{that an arrival occurs, given that the system is in state } n\}$ and (b) $P\{\text{system is in state } n\}$. Thus,

$$\begin{aligned} q_0 &\propto m\lambda p_0 \\ q_1 &\propto (m-1)\lambda p_1 \\ q_2 &\propto (m-1)\lambda p_2 \\ q_3 &\propto (m-2)\lambda p_3 \\ q_4 &\propto (m-2)\lambda p_4 \\ &\vdots \\ q_{2m-3} &\propto (m-1)\lambda p_{2m-3} \\ q_{2m-2} &\propto (m-1)\lambda p_{2m-2} . \end{aligned} \quad (23)$$

[2] Gross, D. and Harris C. Op.Cit., 1974.

Generally, denoting the integer portion of x by $\text{int}[x]$,

$$q_n = (m - \text{int}[(n+1)/2])p_n/K , \quad 0 \leq n \leq 2m-2 , \quad (24)$$

where the normalization constant K is given by

$$K = mp_0 + \sum_{n=1}^{2m-2} (m - \text{int}[(n+1)/2])p_n . \quad (25)$$

The conditional probability density that an arrival finds k other customers in the system is denoted by ρ_k :

$$\begin{aligned} \rho_0 &= q_0 \\ \rho_k &= q_{2k-1} + q_{2k} , \quad 1 \leq k \leq m-1 . \end{aligned} \quad (26)$$

This function is analogous to the unconditional probability density π_k , given by equation (15).

12. Notationally, denote $F_q(t)$ as the distribution of waiting time in queue, T_q :

$$F_q(t) = P\{T_q \leq t\} . \quad (27)$$

The expected value of time in queue from this distribution has been denoted above as W_q . Denote the conditional expectation of time in queue, given a queue as $W_{q|w}$. For a single-server system, queueing occurs whenever the system is not in the null state. Therefore,

$$W_{q|w} = W_q / (1 - q_0) . \quad (28)$$

An alternative -- to equation (22)--means of calculating W_q in this case is the following. This expression is based on the fact--as observed in the state transition diagram (Figure 1)--that an odd state with k customers (E_{2k-1}) involves $2k$ stage transitions of service to leave the system; and, an even state with k customers (E_{2k}) involves $2k-1$ stage transitions of service to leave the system. Each service-stage transition requires an expected time of $1/\mu$. Thus,

$$W_q = \frac{1}{\mu} \sum_{k=1}^{m-1} (2k q_{2k-1} + (2k-1)q_{2k}) . \quad (29)$$

Altho it is not obvious in this case, the mean of waiting time in queue given by (29) equals the value given by the expression (22) from Little's formula. Clearly, equation (22) is preferable computationally; however, development of (22) has motivated the method for deriving $F_q(t)$.

13. Define an integer n^* such that

$$\begin{aligned} n^* &= n+1 & \text{for } n \text{ odd} \\ n^* &= n-1 & \text{for } n \text{ even,} \end{aligned} \quad (30)$$

with n an index over the system states: $1 \leq n \leq 2m-2$.

Then,

$$F_q(t) = q_0 + \sum_{n=1}^{2m-2} q_n P\{n^* \text{ stages of service are completed in time } \leq t, \text{ given an arrival finds the system in the } n \text{ th state}\} . \quad (31)$$

Since each stage of service requires an exponentially-distributed time, with mean μ^{-1} , the time to complete n^* stages has a cumulative distribution which is the n^* -fold convolution of this, namely the gamma cumulative distribution function (c.d.f.):

$$G(n^*, \mu, t) = \int_0^t \frac{\mu (\mu x)^{n^*-1}}{(n^*-1)!} e^{-\mu x} dx . \quad (32)$$

Using (32), one can write (31) as

$$F_q(t) = q_0 + \sum_{n=1}^{2m-1} q_n G(n^*, \mu, t) . \quad (33)$$

With a change in the summation index and using (30), equation (33) becomes

$$F_q(t) = q_0 + \sum_{k=1}^{m-1} (q_{2k-1} G(2k, \mu, t) + q_{2k} G(2k-1, \mu, t)) . \quad (34)$$

A computationally useful form of (34) may be derived using Molina's theorem ([3] and [4], p. 86):

$$\int_0^x \frac{u^{n-1} e^{-u}}{(n-1)!} du = 1 - \sum_{j=0}^{n-1} \frac{x^j e^{-x}}{j!} ; \quad (35a)$$

$$= G(n, 1, x) . \quad (35b)$$

[3] Molina, E.C. Poisson's Exponential Binomial Limit, Van Nostrand, c. 1942.

[4] Schlenker, G. Numerical Methods in Renewal Theory, (AD 828276), Hdqts USA WECOM, February 1968.

Molina's theorem relates a standardized gamma c.d.f. to a sum of Poisson terms. Note that the c.d.f. in (32) with parameters n^* and μ and with argument t is, via a change of variable, equivalent to $G(n^*, 1, \mu t)$. Thus, the gamma c.d.f.'s in (34) can be replaced by the sums required by (35) using an argument x , where

$$x = \mu t . \quad (36)$$

This substitution yields the formula for calculating $F_q(t)$:

$$F_q(t) = 1 - \sum_{k=1}^{m-1} (q_{2k-1} \sum_{j=0}^{2k-1} \frac{x^j e^{-x}}{j!} + q_{2k} \sum_{j=0}^{2k-2} \frac{x^j e^{-x}}{j!}) , \quad (37)$$

or

$$\begin{aligned} F_q(t) &= 1 - q_1 (e^{-x} + x e^{-x}) - q_2 e^{-x} \\ &- e^{-x} \sum_{k=2}^{m-1} (q_{2k-1} \sum_{j=0}^{2k-1} \frac{x^j}{j!} + q_{2k} \sum_{j=0}^{2k-2} \frac{x^j}{j!}) . \end{aligned} \quad (38)$$

This unconditional probability distribution for time in queue can also be used to get the conditional c.d.f., given an arrival must queue. The latter is denoted by $F_{q|w}(t)$. Since q_0 is the probability that an arrival does not need to queue,

$$F_{q|w}(t) = (F_q(t) - q_0) / (1 - q_0) . \quad (39)$$

Using (34) and (39),

$$F_{q|w}(t) = \frac{1}{1 - q_0} \sum_{k=1}^{m-1} (q_{2k-1} G(2k, \mu, t) + q_{2k} G(2k-1, \mu, t)) . \quad (40)$$

Note that the conditional c.d.f. of time in queue is a weighted sum of gamma distributions.

14. Mean and Variance of Conditional Waiting Time in Queue

The j th origin moment of $G(n, \mu, t)$ is given, from p. 33, [4], as

$$a_j = \mu^{-j} \prod_{i=1}^j (n+i-1) . \quad (41)$$

[4] Schlenker, G. Op. Cit.

Specifically, the first and second origin moments are:

$$\begin{aligned} a_1 &= n\mu^{-1} \\ a_2 &= n(n+1)\mu^{-2} . \end{aligned} \quad (42)$$

Using these results, one can obtain expressions for the first and second origin moments for the conditional time in queue ($T_{q|w}$).

From (40),

$$E[T_{q|w}] = \frac{1}{1-q_0} \sum_{k=1}^{m-1} q_{2k-1} 2k\mu^{-1} + q_{2k} (2k-1)\mu^{-1} , \quad (43)$$

and

$$E[T_{q|w}^2] = \frac{1}{1-q_0} \sum_{k=1}^{m-1} q_{2k-1} 2k(2k+1)\mu^{-2} + q_{2k} 2k(2k-1)\mu^{-2} . \quad (44)$$

Then, the variance of the conditional time in queue is obtained from

$$V[T_{q|w}] = E[T_{q|w}^2] - (E[T_{q|w}])^2 . \quad (45)$$

The expression $E[T_{q|w}]$ was earlier denoted by $W_{q|w}$, and was related to W_q by (28). Thus, equation (28) together with (22) provide a consistency check for the value given by (43). This check is performed in the attached computer program FINITE.ME2.Q.

15. Multiple Servers With Exponential Service Times

The formulas derived at this point pertain to the case in which a single server exhibits service times which are Erlang(2). The Kolmogorov equations for this case, tho simple enough, did not lead to a solution of the state probabilities (p_n) in closed form. All equations involved numerical solution with a degree of complexity requiring computer assistance. Another model of a finite-customer population which does lend itself to a somewhat simpler solution is one to which I now turn. In this model the service times are exponential random variables--i.e., Erlang(1) distributed. Further, the number of servers is generalized to an arbitrary integer c . This model has been developed in some detail by Gross and Harris [2]. For this model p_n represents the probability that n customers are found in the service system in stochastic steady state. From p. 118 of Ref [2],

[2] Gross, D. and Harris, C. Op. Cit.

$$p_n = \binom{m}{n} \left(\frac{\lambda}{\mu}\right)^n p_0, \quad 0 \leq n < c \quad (46a)$$

$$p_n = \binom{m}{n} \frac{n!}{c^{n-c} c!} \left(\frac{\lambda}{\mu}\right)^n p_0, \quad c \leq n \leq m. \quad (46b)$$

The value of the probability for the null state is found by setting the sum of the state probabilities to unity.

$$p_0^{-1} = \sum_{n=0}^{c-1} \binom{m}{n} \left(\frac{\lambda}{\mu}\right)^n + \sum_{n=c}^m \binom{m}{n} \frac{n!}{c^{n-c} c!} \left(\frac{\lambda}{\mu}\right)^n. \quad (47)$$

Equations for the mean and variance of the number of customers in the system--analogous to (16) and (17)--are:

$$\bar{N}_{sys} = \sum_{k=1}^m k p_k \quad (48)$$

$$V(N_{sys}) = \sum_{k=1}^m k^2 p_k - (\bar{N}_{sys})^2. \quad (49)$$

Since queueing occurs when $N_{sys} > c$, the expected number of customers in queue is given by

$$\bar{N}_q = \sum_{k=c}^m (k-c) p_k$$

or

$$\bar{N}_q = \bar{N}_{sys} - c + \sum_{k=0}^{c-1} (c-k) p_k. \quad (50)$$

Little's formula is also valid for this model; hence, the mean system waiting time and mean waiting time in queue are given by (21) and (22), respectively.

16. Distribution of Time in Queue (Exponential Services)

The notational conventions used in pgf 11 for the model with Erlang(2) service times are also used here. The conditional probability that an arriving customer finds the system in the n th state is denoted q_n . Bayes' rule requires that

$$q_n = (m-n)p_n/K, \quad (51)$$

where the normalization constant K is found by requiring that

$$\sum_{n=0}^{m-1} q_n = 1.$$

Thus,

$$K = 1/\sum_{n=0}^{m-1} (m-n)p_n. \quad (52)$$

The following derivation for the c.d.f. of time in queue for a c-server system is a generalization of a result found on p. 125 of [2].

Let $F_q(t)$ be the cumulative (unconditional) distribution of waiting time in queue as in (27). Then,

$$F_q(t) = \sum_{n=c}^{m-1} q_n P\{n-c+1 \text{ job completions occur in time } \leq t, \text{ given an arrival finds } n \text{ in the system}\} + P\{\text{system is open}\}, \quad (53a)$$

where

$$P\{\text{system is open}\} = \sum_{n=0}^{c-1} q_n. \quad (53b)$$

In this model a job completion occurs in exponentially distributed time with rate constant $c\mu$, when the system is full.

Therefore, the

$$P\{n^* \text{ completions occur in time } \leq t\}$$

is given by

$$G(n^*, c\mu, t),$$

with the gamma c.d.f. defined by (32).

Thus,

$$F_q(t) = \sum_{n=c}^{m-1} q_n G(n-c+1, c\mu, t) + \sum_{n=0}^{c-1} q_n. \quad (54)$$

Using Molina's theorem (35), a computationally useful form of (54) is obtained.

$$F_q(t) = 1 - q_c e^{-c\mu t} - e^{-c\mu t} \sum_{n=c+1}^{m-1} q_n \sum_{j=0}^{n-c} \frac{(c\mu t)^j}{j!}. \quad (55)$$

From (54) it is seen that the unconditional expected time in queue is the weighted sum of the means of gamma distributions having different shape parameters. With (42),

$$E[T_q] = \sum_{n=c}^{m-1} q_n (n-c+1)/(c\mu). \quad (56)$$

Similarly,

$$E[T_q^2] = \sum_{n=c}^{m-1} q_n (n-c+1)(n-c+2)/(c\mu)^2. \quad (57)$$

Then, the unconditional variance of time in queue is

$$V(T_q) = E[T_q^2] - (E[T_q])^2. \quad (58)$$

[2] Gross, D. and Harris, C. Op. Cit.

The conditional probability distribution of queueing time, given an arrival must wait, is obtained from $F_q(t)$ by

$$F_{q|w}(t) = (F_q(t) - P\{\text{open}\}) / (1 - P\{\text{open}\}) , \quad (59)$$

where $P\{\text{open}\}$ is given by (53b).

This expression is analogous to (39) for the single-server model.

The first two origin moments of the conditional probability distribution of queueing time are obtained from the corresponding unconditional moments (56, 57) by dividing each by $1 - P\{\text{open}\}$. The variance of this conditional c.d.f. is obtained from the general expression in (45).

17. Numerical Results

Before proceeding with the derivation of other analytic relationships, it is interesting to apply the formulas developed at this point to specific examples. The numerical values chosen are motivated by the author's experience with maintenance of machines for producing metal parts of conventional ammunition. In performing the calculations two computer programs have been useful: FINITE.ME2.Q and BEST.SERVICE. These subprograms are listed in Annex A together with an executive driver. The first program implements the model with conditional Poisson* arrivals and Erlang(2) services. The second program calculates all the statistics for the multiserver model having conditionally Poisson arrivals and exponential services. Both programs treat the population of customers as finite. In order to contrast results from a finite-population model with a model having an infinite customer population, I also evaluate the latter model with an arrival rate parameter (λ_∞) chosen to yield the maximum arrival rate in the finite models. Thus,

$$\lambda_\infty = m\lambda . \quad (60)$$

* Poisson arrivals implies that the time between arriving customers is an exponentially distributed random variable and the arrival rate is a constant parameter. In this case, I mean by "conditionally Poisson" that the arrival rate per customer outside the system is a constant.

The model with infinite customer population which is chosen for comparison with the models of this report has an Erlang(2) service time distribution with a single server. Certain mean-value statistics for this case are presented in [2], starting on page 163. This model is referred to as $(M/E_2/1)$ by Gross and Harris [2].

The expected waiting time-- W_q , in the notation of [2]--for $(M/E_2/1)$ is

$$W_q = \frac{3\lambda_\infty}{4\mu_\infty(\mu_\infty - \lambda_\infty)}$$

where μ_∞ is the stage service rate, denoted elsewhere as μ .

The expected waiting time in the system (W) is given by (20), since this equation is independent of population size. Little's formula yields the expected number in the queue (L_q) and the expected number in the system (L):

$$L_q = \lambda_\infty W_q \quad (62a)$$

$$L = \lambda_\infty W. \quad (62b)$$

The probability that the service system is empty (p_0) is also given in [2]:

$$P\{N=0\} \equiv p_0 = 1 - \lambda_\infty/\mu_\infty. \quad (63)$$

Equations (60) thru (63) were evaluated for comparison with their counterparts of the finite-population models. A numerical comparison of the infinite-population model with the finite population models is made in Table 1. The notational designations for the three models used in Table 1 are Infinite $\Gamma(2)$, Finite $\Gamma(2)$, and Finite Exponential to indicate the customer population size and the service-time distribution function, respectively. The time between arrivals (in minutes) and the number of customers in the population are treated as parameters. One notes that certain output variables agree fairly well across models, such as p_0 , whereas others, such as W_q , do not. Due to the fact that λ_∞ exceeds the average arrival rate in the finite models, the average number in the system and in the queue exceeds their finite-population counterparts. However, it is noteworthy that when the time between arrivals is large (or λ is small) and the customer population is large, the values of $E[N]$ and $E[Q]$ from the infinite model approximate the corresponding values in the finite $\Gamma(2)$ model.

TABLE 1

COMPARISON OF THE STOCHASTIC PROPERTIES OF
SEVERAL MODELS OF MAINTENANCE SERVICE SYSTEMS

Assumptions:

Distribution of time between arrivals is exponential for each customer in a finite population of customers.

There is a single server with mean service time of 20 minutes.

$1/\lambda$ (1) (min)	Number Customers	System (2) Property	Type of Model (3)		
			Inf., $\Gamma(2)$	Finite, $\Gamma(2)$	Finite, Exp
120	2	E[N]	0.4583	0.3128	0.3200
		SD[N]		0.5274	0.5455
		E[Q]	0.1250	0.0316	0.0400
		P{N=0}	0.6667	0.7188	0.7200
		Wq	7.500	2.249	2.857
		Wq W		14.800	20.000
	3	E[N]	0.8750	0.5191	0.5410
		SD[N]		0.6934	0.7372
		E[Q]	0.3750	0.1057	0.1311
		P{N=0}	0.5000	0.5865	0.5902
		Wq	15.000	5.111	6.400
		Wq W		17.181	22.857
	5	E[N]	3.9582	1.0936	1.1624
		SD[N]		1.0479	1.1507
		E[Q]	3.1249	0.4425	0.5228
		P{N=0}	0.1667	0.3489	0.3604
		Wq	74.998	13.593	16.348
		Wq W		24.164	30.820
	10	E[N]	∞	4.1705	4.2588
		SD[N]		1.9462	2.1538
		E[Q]		3.1989	3.3020
		P{N=0}		0.0284	0.0431
		Wq		65.849	69.017
		Wq W		68.824	74.625
	20	E[N]	∞	14.0000	14.0000
		SD[N]		2.1354	2.4494
		E[Q]		13.0000	13.0000
		P{N=0}		0.0000	0.0000
		Wq		260.000	260.001
		Wq W		260.000	260.004

TABLE 1 (CONT)

COMPARISON OF THE STOCHASTIC PROPERTIES OF
SEVERAL MODELS OF MAINTENANCE SERVICE SYSTEMS

Assumptions:

Distribution of time between arrivals is exponential for each customer in a finite population of customers.

There is a single server with mean service time of 20 minutes.

$1/\lambda$ (1) (min)	Number Customers	System (2) Property	Type of Model (3)		
			Inf., $\Gamma(2)$	Finite, $\Gamma(2)$	Finite, Exp
1200	20	E[N]	0.4583	0.4336	0.4668
		SD[N]		0.7210	0.8012
		E[Q]	0.1250	0.1075	0.1413
		P{N=0}	0.6667	0.6739	0.6744
		Wq	7.500	6.591	8.679
		Wq W		21.138	28.049
	30	E[N]	0.8750	0.7998	0.8928
		SD[N]		1.0634	1.2358
		E[Q]	0.3750	0.3131	0.4076
		P{N=0}	0.5000	0.5133	0.5149
		Wq	15.000	12.868	16.806
		Wq W		27.186	35.809

Notes for Table 1.

(1) The mean time between arrivals per customer is $1/\lambda$. Thus, the maximum arrival rate is λ times the number of customers.

(2) The following notation is used to denote system properties:

$E[N]$	Statistically expected number of customers in the service system.
$SD[N]$	Standard deviation of the number of customers in the service system.
$E[Q]$	Expected value of the number in the service queue.
$P\{N=\emptyset\}$	Probability that the system is empty.
W_q	Expected value of the waiting time (minutes) in the service queue.
$W_q W$	Expected waiting time (minutes) given a customer must queue.

(3) Models are characterized here by the population of customers and by the form of the service-time distribution. Thus, "Inf., $\Gamma(2)$ " signifies an infinite population of customers and a service-time distribution which is gamma (or Erlang) with shape parameter 2. The form of the c.d.f. of time t is

$$F(t) = 1 - (1 + \beta t)e^{-\beta t},$$

with parameter β . The mean of t with this distribution is $2/\beta$.

The other model has a finite customer population and exponentially distributed service times.

18. Comparisons Between Finite Models

The model with exponential service times is computationally much easier than the $\Gamma(2)$ model. Therefore, it is useful to ask for what statistics the models are in reasonable agreement. For these, the model with exponential service will do. Some observations can be made regarding the similarity of results from the finite $\Gamma(2)$ model versus the finite exponential model. Generally, very good agreement exists between these models for the values of $E[N]$, $SD[N]$, and $P\{N=0\}$. Poorer agreement exists for $E[Q]$, and very poor agreement exists for W_q and $W_{q|w}$. Clearly, the difference in dispersion of service times has the greatest effect upon mean waiting time (or conditional mean waiting time). Differences in the conditional waiting time distributions are displayed explicitly in Figures 2 and 3. These figures differ in only one parameter--customer population size. Figures 4 and 5 illustrate the parametric effect of population size for the model with exponential services. Using this model, the parametric effect of number of servers on the c.d.f. of waiting time in queue is shown in Figure 6. These probability distributions are plotted on Weibull probability paper for convenience, since extreme variation in waiting time is shown for any value of a parameter. The following interesting observation can be made from Figure 6. When the number of servers increases--indicating improving service--the conditional distribution function of time in queue closely approximates an exponential distribution. An exponential distribution plots as a straight line on Weibull paper with a slope of unity (Note scale difference of abscissa and ordinate.) Even tho the mathematical form of (55) is clearly not that of an exponential c.d.f., nevertheless an exponential form is a good approximation under certain conditions. After examining a variety of parametric variations with this model, I conclude that a condition under which the exponential approximation is pragmatically adequate is the following: when the mean value of conditional waiting time in queue does not exceed the mean service time. Examples illustrating this point are given in Table 2. The exponential distribution, with single parameter θ , which approximates the conditional c.d.f. of waiting time in queue is denoted by $\tilde{F}(t, \theta)$. The exact c.d.f. is denoted $F(t)$.

$$\tilde{F}(t, \theta) = 1 - \exp(-t/\theta) . \quad (64)$$

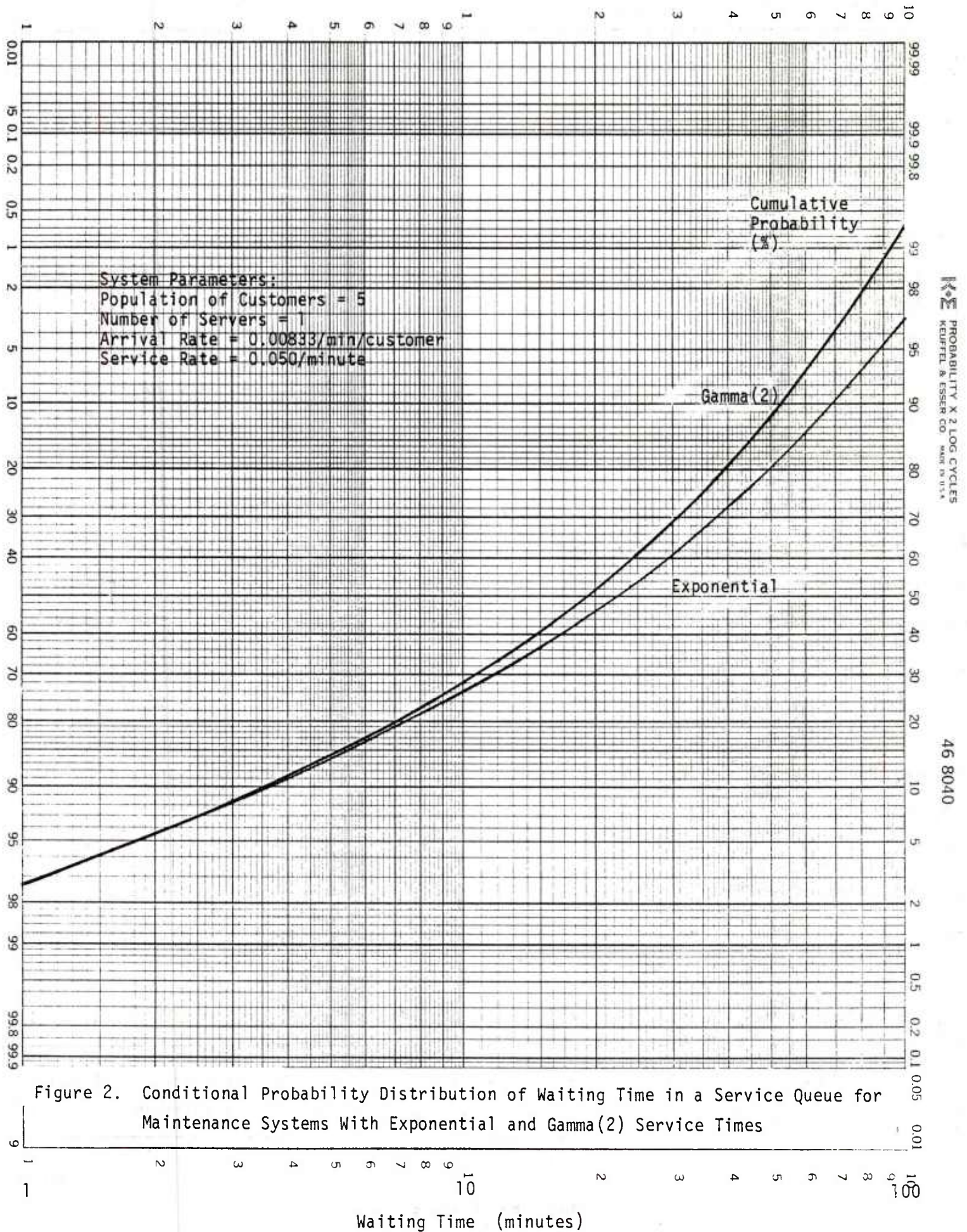
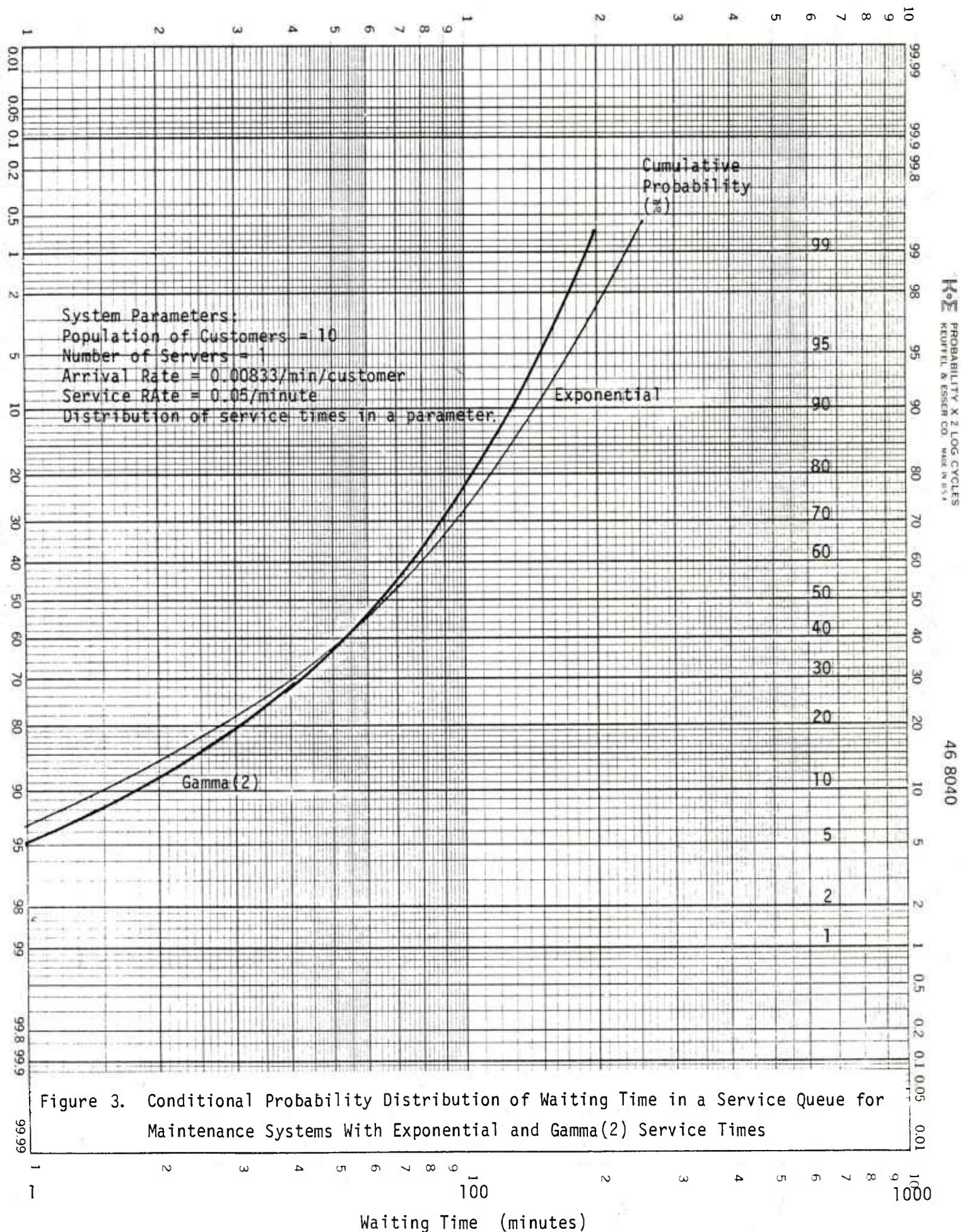
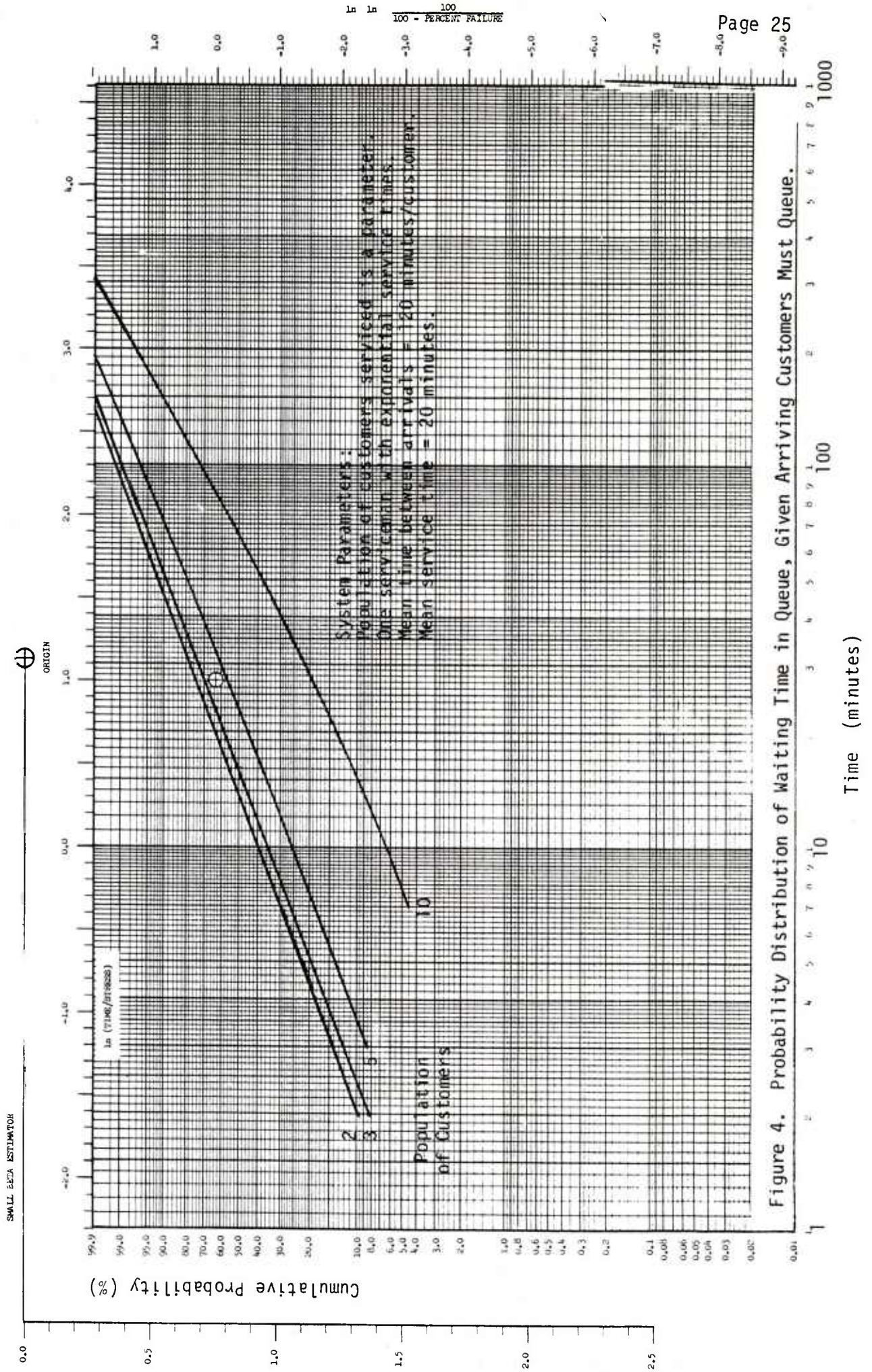


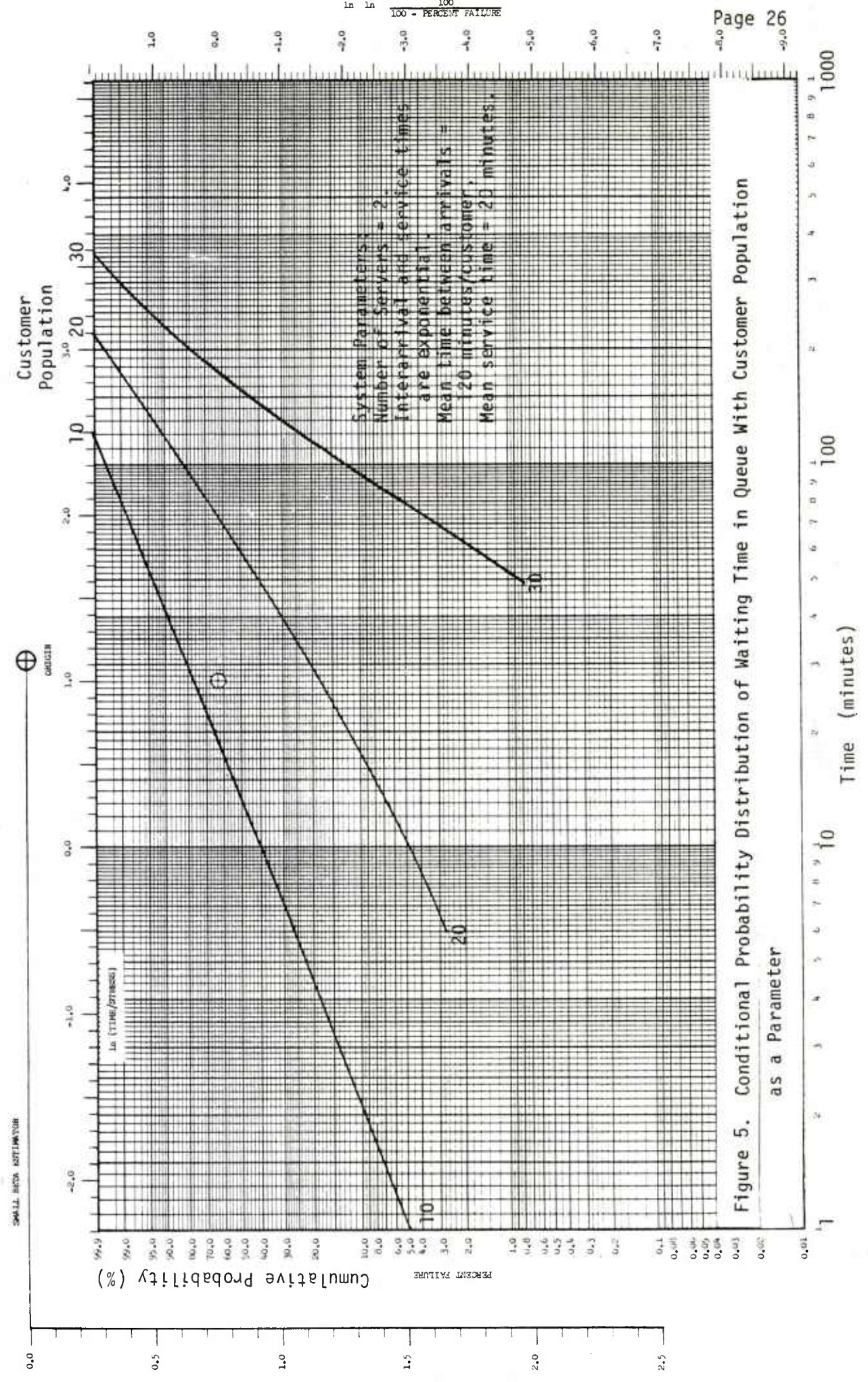
Figure 2. Conditional Probability Distribution of Waiting Time in a Service Queue for Maintenance Systems With Exponential and Gamma(2) Service Times



No. 118-2 WEIBULL PROBABILITY
(0.01 - 99.9)
x 3-CYCLE LOGARITHMIC

TECHNICAL ENGINEERING AIDS TO MANAGEMENT
104 BELOUSE AVENUE
LOWELL, MASSACHUSETTS





TECHNICAL ~ ENGINEERING AIDS TO MANAGEMENT
104 BELOUSE AVENUE
LOWELL, MASSACHUSETTS

No. 118-2 2100ULL PROBABILITY
 (0.01 - 99.99)
x 3-CYCLE LOGARITHMIC

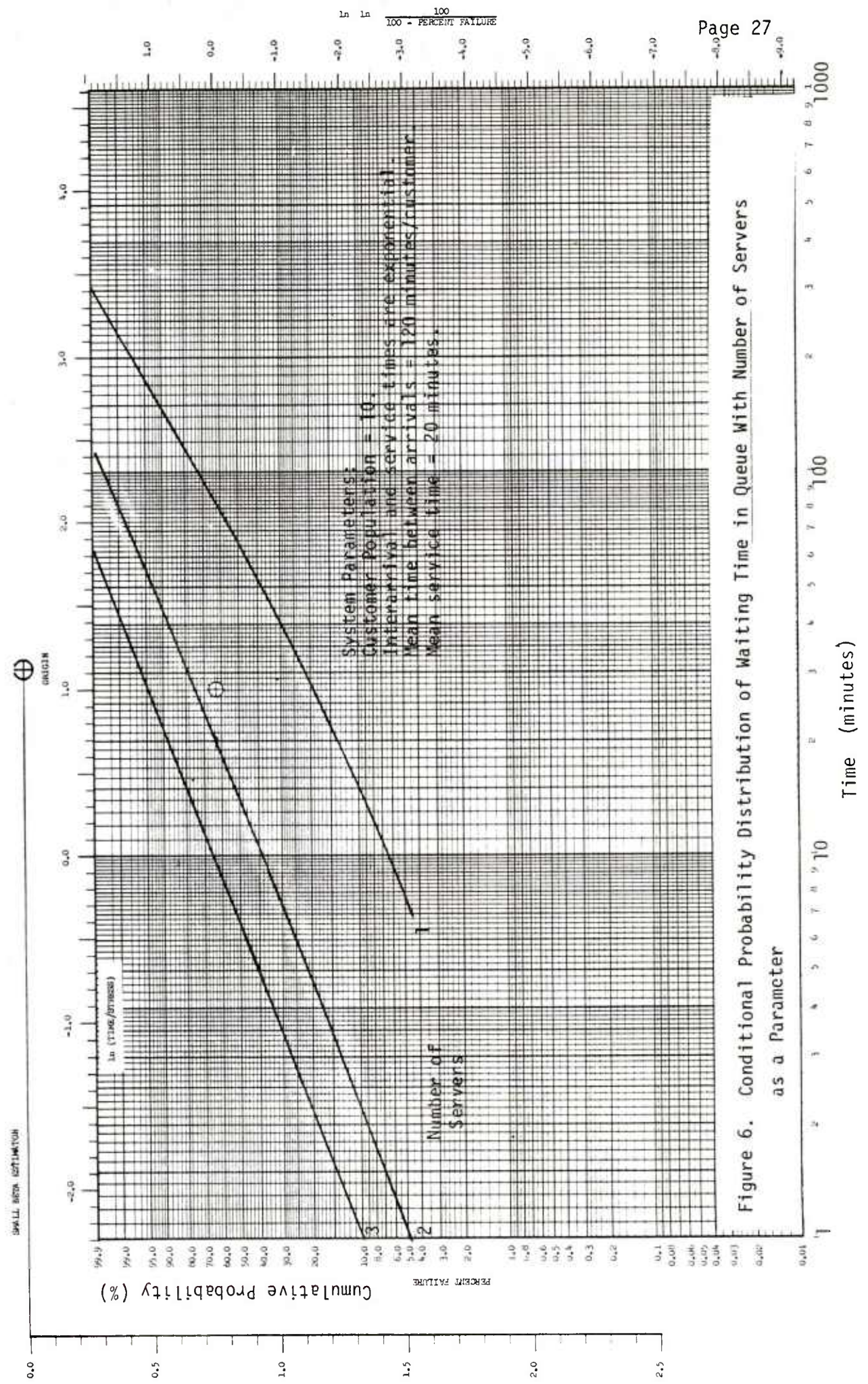


Figure 6. Conditional Probability Distribution of Waiting Time in Queue With Number of Servers as a Parameter

TABLE 2.

COMPARISON OF EXACT AND APPROXIMATE* CONDITIONAL PROBABILITY DISTRIBUTIONS OF WAITING
TIME IN QUEUE FOR A SERVICE SYSTEM⁺ WITH SEVERAL SERVERS AND WITH A FINITE CUSTOMER POPULATION

Time, t (minutes)	Number of Servers							
	1 (left t-scale)		2 (right t-scale)		3 (right t-scale)		4 (right t-scale)	
	F(t)	$\bar{F}(t, 74.6)$	F(t)	$\bar{F}(t, 18.6)$	F(t)	$\bar{F}(t, 9.41)$	F(t)	$\bar{F}(t, 6.17)$
7	0.0450	0.0896	0.0478	0.0523	0.0984	0.1008	0.1482	0.1496
14	0.0944	0.1711	0.0936	0.1019	0.1874	0.1915	0.2747	0.2768
21	0.1472	0.2454	0.2199	0.2357	0.4063	0.4122	0.5530	0.5553
28	0.2025	0.3129	0.3964	0.4159	0.6505	0.6545	0.8017	0.8022
56	0.4308	0.5279	0.6467	0.6588	0.8815	0.8806	0.9617	0.9609
98	0.7156	0.7312	0.7989	0.8007	0.9609	0.9587	0.9928	0.9923
140	0.8823	0.8469	0.8883	0.8836	0.9874	0.9857	0.9987	0.9985
196	0.9711	0.9277	0.9393	0.9320	0.9960	0.9951	0.9998	0.9997
280	0.9975	0.9766	0.9677	0.9603	0.9988	0.9983	1.0000	1.0000

* The approximation is an exponential distribution of the form $\bar{F}(t, \theta) = 1 - \exp(-t/\theta)$. The exact distribution is denoted by $F(t)$.

+ Each customer has an exponentially distributed interarrival time. Service times are exponential.

Mean time between interarrivals = 120 minutes/customer.

Customer population = 10.

Mean service time = 20 minutes.

The parameter θ is assigned the exact conditional mean waiting time ($W_{q|w}$). This value can be obtained from easily computed quantities as follows. The system mean waiting time is calculated from (21). Then, W_q is calculated via (20). Both of these relations exploit Little's formula. Then, Bayes' theorem requires

$$W_{q|w} = W_q / (1 - p_q) , \quad (65)$$

where p_q is the probability that an arrival must queue. With c servers,

$$p_q = 1 - \sum_{n=0}^{c-1} q_n . \quad (66)$$

For a single server,

$$q_0 = 1 - p_q \quad (67)$$

$$q_0 = mp_0 / \sum_{n=1}^{m-1} (m-n) p_n . \quad (68)$$

Four pairs of c.d.f.'s are shown in Table 2. The parameter which distinguishes each pair is the number of servers, which varies from 1 to 4. The constants across comparisons are customer population size, mean interarrival time per customer, and mean service time. The mean service time is 20 minutes. Note that the conditional mean waiting time is less than 20 minutes in the last three comparisons, but not the first. In the first case, with a single server the approximation is poor; whereas, in the other cases the approximation is quite good.

19. Conclusions

The inferences drawn from the numerical examples are summarized here.

- o Unless the activity parameter ($m\lambda/\mu$) is smaller than about 1/3, use of queueing theory for an infinite customer population to describe the behavior of a finite queueing model is quite inexact. Further, there are no compelling numerical reasons for using results applicable to infinite systems.
- o Numerical problems in calculating the system steady-state probability vector were non-existent using double-precision arithmetic on the PRIME 750 minicomputer.
- o If expected values of service system properties such as number of customers in the system and/or in the queue are of primary concern, the differences between finite models having exponential and $r(2)$ service times seem insignificant in practice.

- o One must qualify the previous conclusion. The form of the probability distribution of service time should be selected carefully when the conditional mean time in queue must be accurate. Even with the same mean, different service time c.d.f.'s may yield quite different mean waiting times. This observation holds for exponential versus $r(2)$ service-time distributions.
- o The functional form of the conditional probability distribution of waiting time in queue is generally quite complex. But, a simple approximation exists under certain conditions. For a model with exponential service times, the approximation is valid when the conditional average waiting time in queue does not exceed the mean service time.

20. Recap and Survey of Other Derivations

A review at this point is appropriate. I have discussed two mathematical models used to describe the maintenance system for production machines. Because the population of machines is finite, these queueing models treat a finite customer population. In both cases machine failure is assumed to occur stochastically at a constant rate during machine operation. A failure constitutes the birth of a service demand. Similarly, the completion of service constitutes the death of the service action. With this perspective a Markov birth-death process was used to describe a particular service system probabilistically. This single-server system completes service in a time that has an Erlang(2)-- or $r(2)$ --probability distribution. For contrast, a second model was described which exhibits exponentially distributed service times. The mathematical development of the latter model was nearly complete in the literature on queueing. Only the conditional c.d.f. of waiting time in queue needed to be derived for the multiple-server case. This was done next. Without further mathematical developments, I presented some numerical results. Inferences were made from these results. In the remainder of this memo I will generalize the analytic results of the finite $r(2)$ model to treat several servers. This generalization proceeds in steps. First, two servers are considered. The system state-transition diagram is constructed. From this the Kolmogorov equations are written. Then the method of numerical solution is indicated and certain statistical properties are derived. In deriving these results, equations for the general multiple server case are presented. Finally, a finite $r(2)$ model with three servers is considered. The same procedure for deriving results is followed here as was used for the one- and two-server cases.

21. Two Servers With Erlang(2) Service

The equations derived in paragraphs 6 thru 14 above apply to a service system with a single server. This case was selected initially because of its mathematical simplicity. Frequently, several repairmen work independently and in parallel on machines of (a segment of) a production system. To introduce the treatment of multiple servers, consider the case where 2 servers are present. Each is assumed to have an identical service time distribution which (again) is Erlang(2). The derivation of results follows the pattern used for a single server, with appropriate modification in the definition of system states. In labeling the states of the system, it is convenient to start with a notation which uses double subscripts. Later, the doubly subscripted state variable is replaced with a variable having a single subscript. The procedure followed in this case is easily generalized to cases having more servers. An identical approach was used in an analytic communications model for artillery with CLGP Copperhead (p. 7, [5]).

22. The state of a two-server system is characterized by the number of customers in the system and by the stage of service of each of the servers. The first server is arbitrarily considered to take the first customer. Thus, when only one customer is in the system, the first server is either in the first stage of service or the second, with the other (2nd) server idle. This situation can be represented iconically by the two system configurations:

$$\begin{array}{cc} 0 & 1 & 0 \\ & 0 & 0 \end{array} \quad \text{or} \quad \begin{array}{cc} 0 & 0 & 1 \\ & 0 & 0 \end{array},$$

where the first integer of the first row is the number of customers in queue and where a 1 is placed in the active stage of service for each server.

States are symbolized by the label E_{ij} , where the index i equals the number of customers (k) in the system and the second index j takes on values from 1 to 3 indicating service configuration. All of the state configurations and indices are shown in Table 3.

[5] Schlenker, G. "Field Artillery Communication Studies with Application to CLGP," Systems Analysis Directorate Activities Summary for April 1977, May 1977.

TABLE 3

STATES OF THE SERVICE SYSTEM WITH
TWO SERVERS AND TWO STAGES OF SERVICE

Label (i,j)	Q	Configuration		Customers in the System
		Stage 1	Stage 2	
0	0	0	0	0
		0	0	
1,1	0	1	0	1
		0	0	
1,2	0	0	1	1
		0	0	
2,1	0	1	0	2
		1	0	
2,2	0	0	1	2
		1	0	
2,3	0	0	1	2
		0	1	
3,1	1	1	0	3
		1	0	
3,2	1	0	1	3
		1	0	
3,3	1	0	1	3
		0	1	
Generally, for $2 \leq k \leq m$.				
k,1	k-2	1	0	k
		1	0	
k,2	k-2	0	1	k
		1	0	
k,3	k-2	0	1	k
		0	1	

Customers in
the System

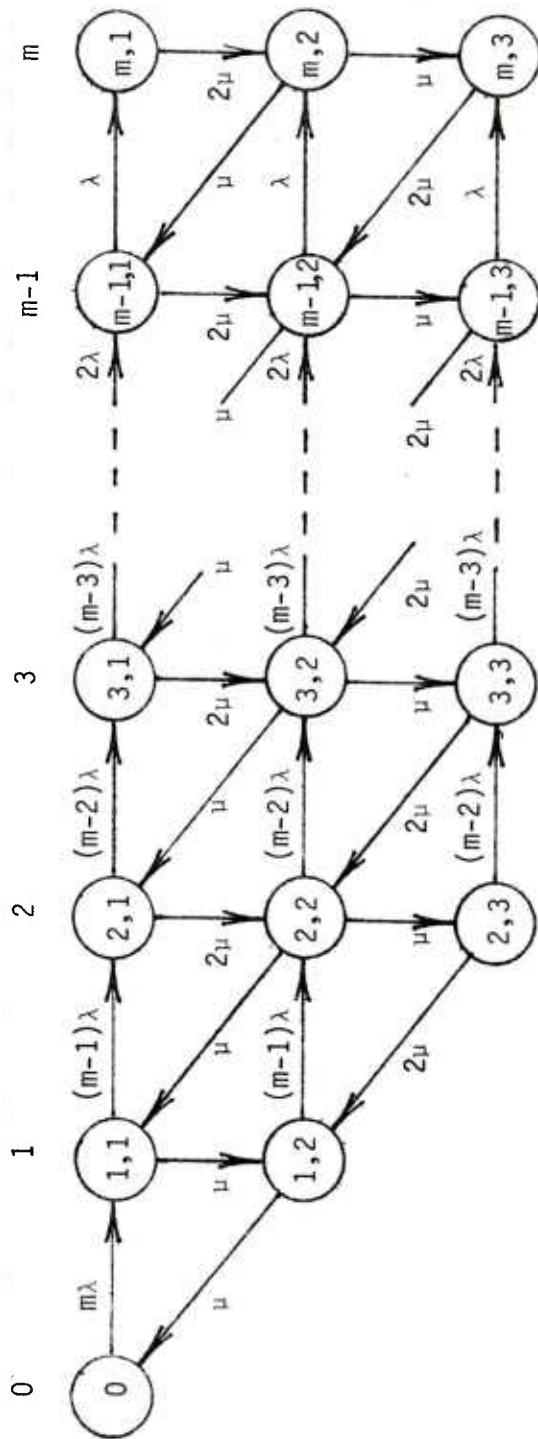


Figure 7. State Transition Diagram for a Two-Server System with Erlang (2) Service Times

23. Kolmogorov Equations for a Two-Server System

As was done for the single-server system, one can construct a state transition diagram for this system preparatory to writing the Kolmogorov equations. However, the states in this diagram are labeled with the double-index notation. This diagram is shown in Figure 7. We use the same notation for the unit birth rate (λ) into the system and the unit death rate (μ) out of the system. The definitions of λ and μ are given in equations (1) and (3). As assumed earlier, the time between arrivals for customers outside the system is exponential. Since each server is assumed to act independently, the transition rate from the $E_{1,1}$ state to the $E_{1,2}$ state is μ (the stagewise service rate for one server). If both servers worked together, the mean stage service time at this point would be less than μ^{-1} .

24. For this system define

$$P_{ij}(t) = P\{\text{system is in state } E_{ij} \text{ at time } t\} . \quad (69)$$

Then, using the state transition diagram, one can write the Kolmogorov equations by inspection. Notational dependence on time is suppressed. For the non-queueing states,

$$\begin{aligned} \dot{P}_0 &= -m\lambda P_0 + \mu P_{12} \\ \dot{P}_{11} &= m\lambda P_0 - ((m-1)\lambda + \mu)P_{11} + \mu P_{22} \\ \dot{P}_{12} &= \mu P_{11} - ((m-1)\lambda + \mu)P_{12} + 2\mu P_{23} \\ \dot{P}_{21} &= (m-1)\lambda P_{11} - ((m-2)\lambda + 2\mu)P_{21} + \mu P_{32} \\ \dot{P}_{22} &= (m-1)\lambda P_{12} + 2\mu P_{21} - ((m-2)\lambda + 2\mu)P_{22} + 2\mu P_{33} \\ \dot{P}_{23} &= \mu P_{22} - ((m-2)\lambda + 2\mu)P_{23} . \end{aligned} \quad (70)$$

For i customers in the system with

$$\begin{aligned} 3 \leq i \leq m-1 , \\ \dot{P}_{i,1} &= (m-i+1)\lambda P_{i-1,1} - ((m-i)\lambda + 2\mu)P_{i,1} + \mu P_{i+1,2} \\ \dot{P}_{i,2} &= (m-i+1)\lambda P_{i-1,2} + 2\mu P_{i,1} - ((m-i)\lambda + 2\mu)P_{i,2} + 2\mu P_{i+1,3} \\ \dot{P}_{i,3} &= (m-i+1)\lambda P_{i-1,3} + \mu P_{i,2} - ((m-i)\lambda + 2\mu)P_{i,3} . \end{aligned} \quad (71)$$

The last three states are described by

$$\begin{aligned}
 \dot{P}_{m,1} &= \lambda P_{m-1,1} - 2\mu P_{m,1} \\
 \dot{P}_{m,2} &= \lambda P_{m-1,2} + 2\mu P_{m,1} - 2\mu P_{m,2} \\
 \dot{P}_{m,3} &= \lambda P_{m-1,3} + \mu P_{m,2} - 2\mu P_{m,3} .
 \end{aligned} \tag{72}$$

25. Steady-State Equations (Two Servers)

To obtain the steady-state solution to the above equations, all derivatives are set to zero. At this point, the double-subscript notation is changed to a single subscript notation in denoting states. Let the probability of the k th state, in stochastic steady state, be denoted by p_k , so that

$$\begin{aligned}
 p_0 &= P_{00}(\infty) \\
 p_1 &= P_{11}(\infty) \\
 p_2 &= P_{12}(\infty) \\
 p_3 &= P_{21}(\infty) \\
 p_4 &= P_{22}(\infty) \\
 p_5 &= P_{23}(\infty)
 \end{aligned} \tag{73a}$$

In general, for $2 \leq i \leq m$ and $1 \leq j \leq 3$,

$$k = 3(i-1) + j-1$$

and

$$p_{3i+j-4} = P_{ij}(\infty) . \tag{73b}$$

Note that $\max(k) = 3m-1$.

With this notational change, the steady-state system equations are, from (70, 71, 72, 73),

$$-\lambda p_0 + \mu p_2 = 0$$

$$m\lambda p_0 - ((m-1)\lambda + \mu)p_1 + \mu p_4 = 0$$

$$\mu p_1 - ((m-1)\lambda + \mu)p_2 + 2\mu p_5 = 0$$

$$(m-1)\lambda p_1 - ((m-2)\lambda + 2\mu)p_3 + \mu p_7 = 0$$

$$(m-1)\lambda p_2 + 2\mu p_3 - ((m-2)\lambda + 2\mu)p_4 + 2\mu p_8 = 0$$

$$\mu p_4 - ((m-2)\lambda + 2\mu)p_5 = 0 . \quad (74)$$

Since the probabilities $\{p_k\}$ must sum to unity, the first (redundant) equation in (74) can be discarded and p_0 eliminated from the second via the substitution

$$p_0 = 1 - \sum_{k=1}^{3m-1} p_k . \quad (75)$$

This operation permits the equations for $\{p_k\}$ to be written in matrix-vector form:

$$A \underline{p} = \underline{b} , \quad (76)$$

where

$\underline{p}$ and $\underline{b}$ are $(3m-1 \times 1)$ and A is $(3m-1 \times 3m-1)$.

From the second equation in (74), with (75),

$$a_{11} = -2m\lambda + \lambda - \mu$$

$$a_{14} = \mu - m\lambda$$

$$a_{1j} = -m\lambda , \quad 1 < j < 3m-1 \text{ and } j \neq 4 . \quad (77)$$

Also

$$\underline{b}' = (-m\lambda , 0, 0, \dots, 0) . \quad (78)$$

The elements of $A--\{a_{rc}\}--$ can be assigned in groups of 3 rows each as follows:

For $6 \leq k \leq 3m-4$, corresponding to $3 \leq i \leq m-1$ and $1 \leq j \leq 3$ in the double-subscript notation,

$$\text{let } r = 3(i-1) . \quad (79a)$$

Then,

$$a_{r,3(i-2)} = (m-i+1)\lambda$$

$$a_{r,3(i-1)} = -((m-i)\lambda + 2\mu)$$

$$a_{r,3(i+1)} = \mu . \quad (79b)$$

Now assign

$$r = 3(i-1) + 1 \quad (80a)$$

in

$$a_{r,3(i-2) + 1} = (m-i+1)\lambda$$

$$a_{r,3(i-1)} = 2\mu$$

$$a_{r,3(i-1) + 1} = -((m-i)\lambda + 2\mu)$$

$$a_{r,3i+2} = 2\mu . \quad (80b)$$

Finally, for

$$r = 3(i-1) + 2 , \quad (81)$$

$$a_{r,3(i-2) + 2} = (m-i+1)\lambda$$

$$a_{r,3(i-1) + 1} = \mu$$

$$a_{r,3(i-1) + 2} = -((m-1)\lambda + 2\mu) . \quad (82)$$

For the last three rows in A,

$$a_{3m-3,3m-6} = \lambda$$

$$a_{3m-3,3m-3} = -2\mu$$

$$a_{3m-2,3m-5} = \lambda$$

$$a_{3m-2,3m-3} = 2\mu$$

$$a_{3m-2,3m-2} = -2\mu$$

$$a_{3m-1,3m-4} = \lambda$$

$$a_{3m-1,3m-2} = \mu$$

$$a_{3m-1,3m-1} = -2\mu . \quad (83)$$

As in the case with a single server, the state probability vector ($\underline{p}$) is obtained via (14), with $\underline{b}$ and A given in (78) thru (83).

26. Statistics for the Two-Server System

To calculate the mean and variance of number of customers in the system requires aggregation of the system-state probabilities. Recall that π_k is defined as the probability that k customers are in the system. Referring to Table 3 for the definition of states,

$$\begin{aligned}\pi_0 &= P_0 \\ \pi_1 &= P_{11} + P_{12} \\ \pi_k &= \sum_{j=1}^3 P_{kj} \quad , \quad 2 \leq k \leq m .\end{aligned}\tag{84a}$$

Using the single-subscript notation, from (72 and 73),

$$\begin{aligned}\pi_0 &= p_0 \\ \pi_1 &= p_1 + p_2 \\ \pi_k &= p_{3k-3} + p_{3k-2} + p_{3k-1} \quad , \quad 2 \leq k \leq m .\end{aligned}\tag{84b}$$

The mean and variance of number of customers in the system are obtained from (16) and (17). To generalize (84b) for a system with c servers, note that

$$\begin{aligned}\pi_0 &= p_0 \\ \pi_1 &= p_1 + p_2 \\ \pi_2 &= p_3 + p_4 + p_5 \\ &\vdots \\ \pi_{c-1} &= \sum_{j=\ell-c+1}^{\ell} p_j\end{aligned}\tag{85a}$$

with

$$\ell = c(c+1)/2 - 1 .\tag{85b}$$

For $c \leq k \leq m$,

$$\pi_k = \sum_{j=(c+1)k-\ell-1}^{(c+1)k-\ell-1+c} p_j .\tag{85c}$$

With multiple servers, the expected value and variance of the number of busy servers (N_{busy}) are:

$$E[N_{\text{busy}}] = \sum_{k=1}^c k \pi_k + c \sum_{k=c+1}^m \pi_k \quad (86)$$

and

$$V[N_{\text{busy}}] = \sum_{k=1}^c k^2 \pi_k + c^2 \sum_{k=c+1}^m \pi_k - (E[N_{\text{busy}}])^2. \quad (87)$$

With c servers, a queue forms when the number of customers in the system N_{sys} exceeds c . The number in queue

$$N_q = N_{\text{sys}} - c, \quad N_{\text{sys}} \geq c. \quad (88)$$

From this

$$\bar{N}_q = \sum_{k=c+1}^m (k-c) \pi_k \quad (89)$$

and

$$V[N_q] = \sum_{k=c+1}^m (k-c)^2 \pi_k - \bar{N}_q^2. \quad (90)$$

In all these cases Little's formula applies. Hence, the mean waiting time in the system (W) and the mean waiting time in the queue (W_q) are obtained via (21) and (22), respectively.

27. For the specific case of c equal to 2 (with which we are dealing), the conditional state probability density (q_k) is obtained from p_k via a procedure similar to that followed for the single-server case (pgf. 11). In employing Bayes' theorem, states having the same number of customers in the system are grouped. Thus,

$$q_0 = m p_0 / K$$

$$q_1 = (m-1) p_1 / K$$

$$q_2 = (m-1) p_2 / K ;$$

and, in general for $2 \leq k \leq m-1$ and $1 \leq j \leq 3$,

$$q_{3(k-1)+j-1} = K^{-1} (m-k) p_{3(k-1)+j-1}. \quad (91)$$

This set of equations contrasts with (24) for a single server. The constant K is obtained by summing over all states, equating the sum to unity. In the general multiple-server case, the conditional state p.d.f. is obtained from

$$q_0 = mp_0/K$$

$$q_1 = (m-1)p_1/K$$

$$q_2 = (m-1)p_2/K$$

$$q_3 = (m-2)p_3/K$$

$$q_4 = (m-2)p_4/K$$

$$q_5 = (m-2)p_5/K$$

...thru all states with less than c customers in the system. The last such state has index ℓ , where

$$\ell = (c-1)(c+2)/2$$

or

$$\ell = c(c+1)/2 - 1 .$$

Thus,

(92a)

$$q_\ell = (m-c+1)p_\ell/K .$$

(92b)

Additionally, for the states in which the number of customers (k) in the system $\geq c$:

$$q_i = (m-k)p_i/K ,$$

(93a)

with

$$i = (c+1)(k-c) + j + \ell ,$$

(93b)

$$1 \leq j \leq c+1 , \quad c \leq k \leq m-1 .$$

For 2 servers, the conditional probability that an arriving customer finds k others in the system is given by

$$\rho_0 = q_0$$

$$\rho_1 = q_1 + q_2$$

$$\rho_k = \sum_{j=1}^3 q_{3(k-1)+j-1} , \quad 2 \leq k \leq m-1 .$$

(94)

This result corresponds to equation (26) for a single-server system. A generalization of this result to a c-server system is the following:

$$\begin{aligned}
 \rho_0 &= q_0 \\
 \rho_1 &= q_1 + q_2 \\
 \rho_2 &= q_3 + q_4 + q_5 \\
 &\vdots \\
 \rho_{c-1} &= \sum_{j=1}^c q_{\ell-c+j} .
 \end{aligned} \tag{95a}$$

And,

$$\rho_k = \sum_{j=1}^{c+1} q_{(c+1)(k-c)+\ell+j} , \quad c \leq k \leq m-1 . \tag{95b}$$

28. Three Servers With Erlang(2) Service

Derivation of the state equations for the three-server system with Erlang(2) service times follows the pattern used for the two-server system (pgf. 21 ff). To facilitate recognition of states, double subscript notation is also used here. The state E_{ij} signifies that i customers are in the system, with the service stages for the three channels occupied indicated by subscript j . The max value of j is $c+1$ or 4, in this case. These states are depicted iconically and are labeled in Table 4. Note that the last state for which no queueing occurs is $E_{3,4}$. The state transition diagram for this system is shown in Figure 8. As indicated earlier, one can immediately write the Kolmogorov equations by inspection from the state-transition diagram. Parenthetically, the transient version of this model was solved for a communication system having the same transition-state diagram. See p. 14 of [5]. The Kolmogorov equations, in double-subscript notation, yield the time-dependent probabilities $P_{ij}(t)$, for admissible values of i and j . Since the steady-state solution is of interest here, time derivatives are set to zero. Further, the development of a matrix-vector equation for the steady-state probabilities requires conversion to single-subscript notation.

[5] Schlenker, G. Op. Cit., May 1977.

TABLE 4

STATES OF THE SERVICE SYSTEM WITH
THREE SERVERS AND TWO STAGES OF SERVICE

Label (i,j)	Configuration			Customers in the System
	Q	Stage 1	Stage 2	
0	0	0	0	0
		0	0	
		0	0	
1,1	0	1	0	1
		0	0	
		0	0	
1,2	0	0	1	1
		0	0	
		0	0	
2,1	0	1	0	2
		1	0	
		0	0	
2,2	0	0	1	2
		1	0	
		0	0	
2,3	0	0	1	2
		0	1	
		0	0	
3,1	0	1	0	3
		1	0	
		1	0	
3,2	0	0	1	3
		1	0	
		1	0	
3,3	0	0	1	3
		0	1	
		1	0	
3,4	0	0	1	3
		0	1	
		0	1	

TABLE 4 (Cont)

Label (i,j)	Q	Configuration		Customers in the System
		Stage 1	Stage 2	
4,1	1	1	0	4
		1	0	
		1	0	
4,2	1	0	1	4
		1	0	
		1	0	
4,3	1	0	1	4
		0	1	
		1	0	
4,4	1	0	1	4
		0	1	
		0	1	
For k customers in the system, $3 \leq k \leq m$.				
k,1	k-3	1	0	k
		1	0	
		1	0	
k,2	k-3	0	1	k
		1	0	
		1	0	
k,3	k-3	0	1	k
		0	1	
		1	0	
k,4	k-3	0	1	k
		0	1	
		0	1	

Customers in
the System

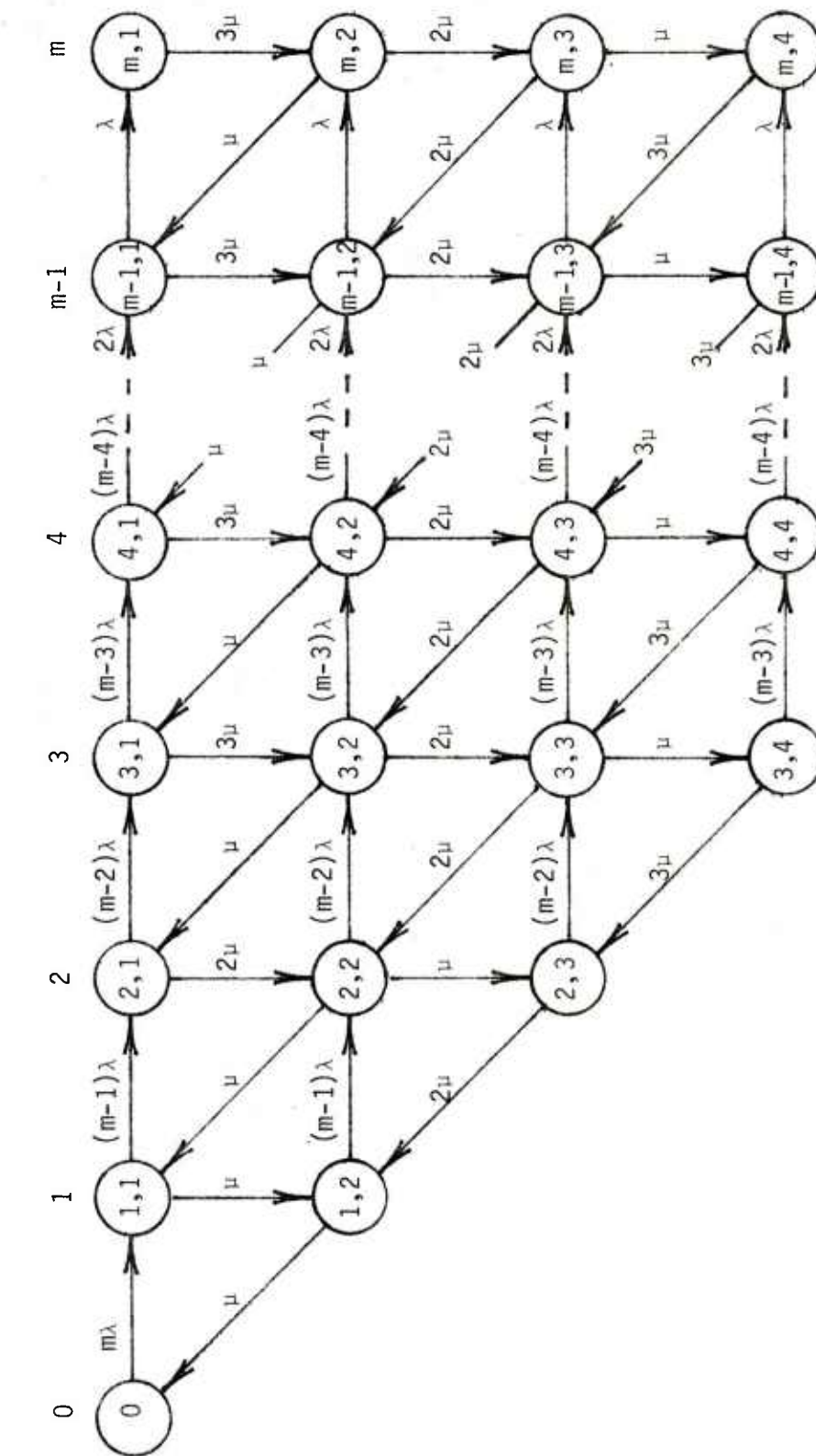


Figure 8. State Transition Diagram for a Three-Server System with Erlang (2) Service Times

29. Steady-State Equations (Three Servers)

The transformation from double to single subscript notation in (73a) is also used for this case. Additionally,

$$p_6 = P_{31}^{(\infty)}$$

$$p_7 = P_{32}^{(\infty)}$$

$$p_8 = P_{33}^{(\infty)}$$

$$p_9 = P_{34}^{(\infty)} .$$

In general, for $3 \leq n \leq m$,

$$p_{4n-6} = P_{n,1}^{(\infty)}$$

$$p_{4n-5} = P_{n,2}^{(\infty)}$$

$$p_{4n-4} = P_{n,3}^{(\infty)}$$

$$p_{4n-3} = P_{n,4}^{(\infty)} .$$

(96)

With these substitutions, the Kolmogorov equations in [5] become:

$$-m\lambda p_0 + \mu p_2 = 0$$

$$m\lambda p_0 - [(m-1)\lambda + \mu]p_1 + \mu p_4 = 0$$

$$\mu p_1 - [(m-1)\lambda + \mu]p_2 + 2\mu p_5 = 0$$

$$(m-1)\lambda p_1 - [(m-2)\lambda + 2\mu]p_3 + \mu p_7 = 0$$

$$(m-1)\lambda p_2 + 2\mu p_3 - [(m-2)\lambda + 2\mu]p_4 + 2\mu p_8 = 0$$

$$\mu p_4 - [(m-2)\lambda + 2\mu]p_5 + 3\mu p_9 = 0$$

$$(m-2)\lambda p_3 - [(m-3)\lambda + 3\mu]p_6 + \mu p_{11} = 0$$

$$(m-2)\lambda p_4 + 3\mu p_6 - [(m-3)\lambda + 3\mu]p_7 + 2\mu p_{12} = 0$$

$$(m-2)\lambda p_5 + 2\mu p_7 - [(m-3)\lambda + 3\mu]p_8 + 3\mu p_{13} = 0$$

$$\mu p_8 - [(m-3)\lambda + 3\mu]p_9 = 0 .$$

For $4 \leq n \leq m-1$,

$$\begin{aligned}
(m-n+1)\lambda p_{4n-10} - [(m-n)\lambda + 3\mu]p_{4n-6} + \mu p_{4n-1} &= 0 \\
(m-n+1)\lambda p_{4n-9} + 3\mu p_{4n-6} - [(m-n)\lambda + 3\mu]p_{4n-5} + 2\mu p_{4n} &= 0 \\
(m-n+1)\lambda p_{4n-8} + 2\mu p_{4n-5} - [(m-n)\lambda + 3\mu]p_{4n-4} + 3\mu p_{4n+1} &= 0 \\
(m-n+1)\lambda p_{4n-7} + \mu p_{4n-4} - [(m-n)\lambda + 3\mu]p_{4n-3} &= 0 .
\end{aligned}$$

The last four states of the system are described by:

$$\begin{aligned}
\lambda p_{4m-10} - 3\mu p_{4m-6} &= 0 \\
\lambda p_{4m-9} + 3\mu p_{4m-6} - 3\mu p_{4m-5} &= 0 \\
\lambda p_{4m-8} + 2\mu p_{4m-5} - 3\mu p_{4m-4} &= 0 \\
\lambda p_{4m-7} + \mu p_{4m-4} - 3\mu p_{4m-3} &= 0 .
\end{aligned} \tag{97}$$

This set of equations contains a redundancy since the state probabilities must sum to unity. To eliminate this redundancy one can substitute

$$p_0 = 1 - \sum_{k=1}^{4m-3} p_k \tag{98}$$

into the first equation in set (97), yielding

$$m\lambda \sum_{k=1}^{4m-3} p_k + \mu p_2 = m\lambda . \tag{99}$$

The second equation in (97), which also involves p_0 , is deleted from the equations to be solved. The final solution set has the following matrix-vector form:

$$A\mathbf{p} = \mathbf{b} , \tag{100a}$$

where $\mathbf{p}$ and $\mathbf{b}$ are $(4m-3 \times 1)$ and A is $(4m-3 \times 4m-3)$.

In this case,

$$\mathbf{b}' = (m\lambda, 0, 0, \dots, 0) . \tag{100b}$$

These equations are analogous to (76), (77), and (78) in the two-server case. In the general c -server case, the dimension of $\mathbf{p}$ is

$$(c+1)m - (c-1)c/2 .$$

30. Statistics for the Three-Server System

Recall that π_k is defined as the probability that k customers are in the system. Elements of the system state probability vector $\underline{p}$ determine the elements of $\underline{\pi}$. The general relationship for a system with c servers is given in (85). Particularizing this for 3 servers,

$$\pi_0 = p_0$$

$$\pi_1 = p_1 + p_2$$

$$\pi_2 = p_2 + p_4 + p_5$$

and, for $3 \leq k \leq m$,

$$\pi_k = \sum_{j=4k-6}^{4k-3} p_j . \quad (101)$$

Other statistics for this case are obtained from their c -server generalizations. Specifically, the mean and variance of busy servers are obtained from (86) and (87). The mean and variance of number of customers in the queue derives from (89) and (90), with mean and variance of number in the system given by (16) and (17). The conditional probability distribution for an arrival finding k other customers in the system is obtained from (95). Since Little's formula holds for a c -server system, the mean waiting time in the system (W) and in the queue (W_q) are obtained from (21) and (22), respectively. For a system with c servers, the probability that an arriving customer must wait in queue is

$$P\{\text{arrival must wait}\} = \sum_{k=c}^{m-1} \rho_k$$

or

$$= 1 - \sum_{k=0}^{c-1} \rho_k . \quad (102)$$

Each of the above statistics is calculated for $c = 1, 2$, or 3 servers in the computer program FINITE.ME2.Q, listed in Annex A.

REFERENCES

1. DD1498, HQ US ARRCOM, DRSAR-SA, March 1983, title: Manufacturing Productivity Study.
2. Gross, Donald and Harris, Carl. Fundamentals of Queueing Theory, John Wiley and Sons, c. 1974.
3. Molina, E.C. Poisson's Exponential Binomial Limit, VanNostrand, c. 1942.
4. Schlenker, G. Numerical Methods in Renewal Theory, (AD 828276), HQ USA WECOM, February 1968.
5. Schlenker, G. "Field Artillery Communication Studies with Application to CLGP," Systems Analysis Directorate Activities Summary for April 1977, DRSAR/SA/N-66, (AD A042792), US ARRCOM, May 1977.

DISTRIBUTION LIST

No. of Copies

1

HQDA (DAMO-ZA)
Wash DC

Commander
USA DARCOM
5001 Eisenhower Ave.
Alexandria, VA 22333

1

ATTN: DRCRE-IP
DRCPA-S

1

Director, US AMSAA
Aberdeen PG, MD 21005
ATTN: DRXSY-DL
DRXSY-R

1

1

Commander, US Army Logistics Center
Fort Lee, VA 23801
ATTN: ATCL-S

1

Director, US Army Ammo Center
Savanna, IL 61074
ATTN: SARAC-DO

1

12

Defense Technical Information Center
Cameron Station
Alexandria, VA 22314

Commander US AMCCOM (D)
Dover, NJ 07801

1

ATTN: DRSMC-QAS (D)

1

DRSMC-LCU-M (D)

1

DRSMC-LCM-E (D)

1

DRSMC-LCE-D (D)

1

DRSMC-RAA (D)

1

DRSMC-LCA-P (D)

Commander US AMCCOM (A)
Aberdeen PG, MD 21005
ATTN: DRSMC-CLY (A)

1

Commander
USA Communications and Electronics Command
Ft. Monmouth, NJ 07703
ATTN: DRSEL-PL-SA

1

Commander USA MICOM
Redstone Arsenal, AL 35809
ATTN: DRSMI-DS

1

DISTRIBUTION LIST (Cont'd)

No. of Copies

1	Commander USA TACOM Warren, MI 48090 ATTN: DRSTA-V
1	Commander USA TECOM Aberdeen PG, MD 21005 ATTN: DRSTE-SY
1	Commander USA Production Base Modernization Agency Dover, NJ 07801 ATTN: Mr. S. Karlin
1	Director US AMETA Rock Island, IL 61299 ATTN: DRXOM-QA
1	Director USA IBEA Rock Island, IL 61299 ATTN: DRXIM-MT/Mr. S. McGlone
1	Director US Army Inventory Research Office Room 800, Custom House 2nd and Chestnut Streets Philadelphia, PA 19160 ATTN: DRXMC-IRO
1	Commander US AMCCOM (R) Rock Island, IL 61299 ATTN: DRSMC-LC (R)/Mr. L. Ambrosini
1	DRSMC-IRD (R)/Mr. G. Cowan
1	DRSMC-IRI (R)/Mr. J. Kilar
1	DRSMC-PP (R)/Mr. M. Craighead
1	DRSMC-SA (R)/Mr. B. Witherspoon
1	DRSMC-SAS (R)/Mr. D. Eckman
1	Mr. F. Northey
1	Mr. J. Wasson
1	DRSMC-SAL (R)/Mr. N. Trier
1	DRSMC-SAA (R)/Mr. B. White
1	DRSMC-OPP (R)/Mr. P. Woodhouse
1	DRSMC-OPI (R)/Mr. R. Heider
2	DRSMC-LEP-L (R)/Ms. Sandi Paul
1	Chief Benet Weapons Lab, LCWCL Armament R&D Center US AMCCOM Watervliet, NY 12189 ATTN: DRSMC-LCB-TL

ANNEX A

COMPUTER SOURCE PROGRAMS

The computer programs in this annex are used to calculate a variety of steady-state statistical properties of queueing systems having a finite customer population and a multi-server service system. Calculations are made for each of two assumptions regarding the probability distribution of service times. The random service time is treated as derived from an Erlang distribution with shape parameter 2 in the program `FINITE.ME2.Q`. The service time is treated as an exponential random variable in the routine `BEST.SERVICE`. Each of these programs call utility routines for inverting and multiplying matrices: `MAT.INVERSE`, `MAT.MPY`, and `MAT.VEC.MPY`. All of these programs are written in SIMSCRIPT 2.5 for the PRIME 750 minicomputer. However, the programs do not employ features unique to this computer. Cross reference lists are included with program statements to facilitate the identification of variable type and to facilitate the location of variables in a program.

A brief (50 lines of executable code) driver program `TEST.FINITE.ME2.Q` reads data interactively from the terminal and echoes these inputs. This program calculates certain queueing statistics for a service system with an infinite population of customers having Poisson arrivals and with Erlang(2) service times. These quantities can be compared with statistics from systems having a finite customer population. The driver program in turn calls `FINITE.ME2.Q` and `BEST.SERVICE`. The first of these programs is largest with 415 lines of executable code. The statistics listed in the program's beginning comments are optionally printed by the routine.

The routine `BEST.SERVICE` has 174 lines of executable code. This program has two calculational modes: one to produce queueing statistics for the parameters supplied and another to sequentially optimize the number of servers, subject to prescribed constraints. In the present application only the calculational mode is invoked.

The major problem segments of all programs are announced via comment statements, which are identified with starting double quote marks. Inputs to the driver program are read from the terminal in response to prompting messages. Output is sent directly to the terminal for display. Since the output is lengthy, it is recommended that a `COMO` file be established to display or print it.

Options = SEQUENCE,IO,SUBCHK,XREF,NOEXPLIST,TRACE3 CACJ SIMSCRIPT 11.5 for PRIME Systems, Release 2.1 06 JAN 1983 15:25:50

```
1  **TEST.FINITE.ME2.G
2  **
3  **DRIVER PROGRAM FOR ROUTINE FINITE.ME2.G
4  **SOLVES THE FINITE QUEUEING PROBLEM WITH POISSON ARRIVALS PER
5  **UNSERVED INDIVIDUAL IN THE POPULATION AND WITH ERLANG(2) SERVICE TIME.
6  **
7  PREAMBLE
8  NORMALLY MODE IS REAL
9  END **OF PREAMBLE
```

Options = SEQUENCE, ID, SUBCHK, XREF, NOEXPLIST, TRACE3
CACI SIMSCRIPT 11.5 for PRIME Systems, Release 2.1
06 JAN 1983 15:25:50

```

1  MAIN
2  DEFINE N, MODE, MPOP, AND NSERVE AS INTEGER VARIABLES
3  DEFINE MODE.BEST AS A TEXT VARIABLE
4  DEFINE P.NSTATE AS A 1-DIMENSIONAL ARRAY
5  LET MODE.BEST="CALCULATE"
6  **
7  **GET INPUTS FROM THE TERMINAL.
8  **
9  **START, START NEW PAGE
10 PRINT 2 LINES THUS
11 INPUT THE ARRIVAL RATE (PER MIN.) PER INDIVIDUAL IN THE POPULATION.
12 IF YOU WISH TO STOP, ENTER ZERO (0).
13 READ LAMBDA
14 IF LAMBDA LE 0.0
15 STOP
16 OTHERWISE
17 PRINT 1 LINE THUS
18 INPUT THE SERVICE RATE (PER MINUTE).
19 READ MUI
20 PRINT 1 LINE THUS
21 INPUT THE NUMBER OF INDIVIDUALS IN THE POPULATION.
22 READ MPOP
23 RESERVE P.NSTATE(*) AS MPOP
24 PRINT 1 LINE THUS
25 INPUT THE NUMBER OF SERVERS.
26 READ NSERVE
27 IF NSERVE GE MPOP
28 PRINT 1 LINE THUS
29 INPUT ERROR. SERVERS > OR = CUSTOMERS.
30 GO TO START
31 OTHERWISE
32 SKIP 4 LINES
33 **
34 **ECHO INPUT PARAMETERS.
35 **
36 LET LAMBDA.INF=MPOP*LAMBDA
37 PRINT 7 LINES WITH LAMBDA, LAMBDA.INF, MUI, MPOP, AND NSERVE
38 THUS
39 INPUT PARAMETERS FOR A STEADY-STATE, FINITE QUEUE
40
41 LAMBDA ***** PER MINUTE PER CUSTOMER
42 MPOP(LAMBDA) ***** PER MINUTE
43 MUI ***** PER MINUTE
44 NSERVE ***** CUSTOMERS
45
46 SKIP 2 LINES
47 IF NSERVE > 3
48 GO TO 11
49 OTHERWISE
50
51 **
52 **CALCULATE EXPECTED VALUES FOR SYSTEM VARIABLES FOR AN INFINITE
53 **POPULATION WITH ERLANG(2) SERVICE.
54 **
55 IF LAMBDA.INF < MUI
56 LET PUE=1.0-LAMBDA.INF/MUI

```

```

45 LET WGE=75*LAMBDA.INF/MU1/(MU1-LAMBDA.INF)
46 LET W=WC+1.0/MU1
47 LET L=LAMBDA.INF*W **LITTLE'S FORMULA
48 LET LG=L-1.0+P0
49 SKIP 2 LINES
50 PRINT 6 LINES WITH P0, WQ, W, L, LG THUS
    EXPECTED, STEADY-STATE VALUES FOR AN INFINITE POPULATION FOR ARRIVALS:
-----
PROBABILITY OF AN EMPTY SYSTEM      *****
WAITING TIME IN QUEUE (MINUTES)      *****
WAITING TIME IN SYSTEM (MINUTES)     *****
NUMBER OCCUPYING THE SYSTEM          *****
NUMBER OCCUPYING THE QUEUE           *****
-----SKIP 4 LINES-----
51 ALWAYS
52
53 **CALL FOR STEADY-STATE, FINITE QUEUEING CALCULATIONS.
54 **
55 CALL FINITE.ME2.G(LAMBDA,MU1,MPOP,NSERVE,MODE) YIELDING P.NULL,P.SYS.FULL,
56 P.CUST.WAIT,ENSYS,SDNSYS,ENG,ESYS,WAIT,EG.WAIT,EG.SDQ,WAIT.GQ,
57 E.BUSY.SERVERS, SD.BUSY.SERVERS, AND P.NSTATE(*)
58
59 **CALCULATE THE SYSTEM STATE FOR THE CASE WHERE SERVICE TIMES ARE EXPONENTIAL.
60 **
61 **CALL BEST.SERVICE(LAMBDA,MU1,MPOP,NSERVE,A,B,1,MODE,BEST) YIELDING P.NULL,
62 P.SYS.FULL,P.NACH.WAIT,ENO.DOWN,SDNO.DOWN,ENO.QUEUED,ESYS.WAIT,EG.WAIT,
63 EG.WAIT.GQ,SDQ.WAIT.GQ,E.BUSY.SERVERS,SD.BUSY.SERVERS, AND P.NSTATE(*)
64 GO TO START
65
66 END **OF TEST.FINITE.ME2.G

```


Options = SEQUENCE, LV, SUBCHK, XREF, NOEXPLIST, TRACE, CACI SIMSCRIPT 11.5 for PRIME Systems, Release 2.1 06 JAN 1983 15:25:50

```

1 ROUTINE FOR FINITE ME2, Q (LAMBDA, MU1, MPOP, NSERVE, MODE) YIELDING P, NULL,
2 P, SYS, FULL, P, CUST, WAIT, ENSYS, SUNSYS, ENG, ESYS, WAIT, EG, WAIT,
3 EQ, WAIT, GQ, SDG, WAIT, GQ, E, BUSY, SERVERS, SD, BUSY, SERVERS, P, NSTATE
4
5 ** THIS ROUTINE CALCULATES THE STEADY-STATE, SYSTEM STATE-PROBABILITY
6 ** VECTOR FOR A SERVICE SYSTEM HAVING EXPONENTIAL INTERARRIVAL TIMES
7 ** FOR EACH MEMBER OF A FINITE CUSTOMER POPULATION, AND HAVING NSERVE SERVERS
8 ** WITH SERVICE TIMES DISTRIBUTED AS ERLANG WITH SHAPE PARAMETER 2.
9
10 ** INPUT:
11 ** LAMBDA
12 ** MU1
13 ** MPOP
14 ** NSERVE
15 ** MODE
16
17 ** OUTPUT:
18 ** P, NSTATE
19
20 ** VECTOR OF PROBABILITY ELEMENTS IN WHICH THE N TH
21 ** ELEMENT IS THE PROBABILITY THAT N INDIVIDUALS ARE
22 ** IN THE SERVICE SYSTEM IN THE STEADY STATE.
23
24 ** P, NULL
25 ** P, SYS, FULL
26 ** P, CUST, WAIT
27 ** ENSYS
28 ** SUNSYS
29 ** ENG
30 ** EG, WAIT
31 ** EG, WAIT, GQ
32
33 ** SDG, WAIT, GQ
34
35 ** E, BUSY, SERVERS
36 ** SD, BUSY, SERVERS
37
38 ** OPTIONAL PRINTING OF THE QUEUE WAITING TIME DISTRIBUTION IS PROVIDED.
39
40 ** ENDOGENOUS VARIABLES:
41 ** KOLMOGOROV STATE-TRANSITION MATRIX
42 ** INVERSE OF AM
43 ** VECTOR OF PROBABILITIES IN WHICH THE N TH
44 ** ELEMENT IS THE PROBABILITY OF FINDING THE
45 ** SYSTEM IN THE N TH STATE, WHERE
46 ** N=2*(NO CUSTOMERS-1)+STAGE OF SERVICE, NO>0.
47 ** VECTOR OF PROBABILITIES IN WHICH THE N TH
48 ** ELEMENT IS THE PROBABILITY THAT AN ARRIVING CUSTOMER
49 ** FINDS THE SYSTEM IN THE N TH STATE. THIS VECTOR
50 ** IS USED TO CALCULATE THE DISTRIBUTION OF
51 ** WAIT IN QUEUE.
52
53 ** I, J, MPOP, TWOM, MAX, MODE, N, AND NSERVE AS INTEGER VARIABLES
54 ** RV, PV, QV, P, NSTATE, AND Q, NSTATE AS REAL, 1-DIMENSIONAL ARRAYS
55 ** DEFINE AM AND ANINV AS A REAL, 2-DIMENSIONAL ARRAYS
56 ** LET TWOM=2*MPOP
57 ** LET MU=2.0*MU1
58 ** SERVICE RATE FOR EACH OF THE 2 STAGES
59 ** LET NL=REAL.F(MPOP)*LAMBDA
60 ** MAX ARRIVAL RATE
61 ** RESERVE P, NSTATE(*) AS MPOP
62 ** RESERVE G, WSTAIL(*) AS MPOP

```

```

58 IF NSERVE LE 0
59 RETURN
60 OTHERWISE
61 IF NSERVE>1
62 GO TO L0
63 OTHERWISE
64 RESERVE BV(1) AS TWOM
65 RESERVE PV(1) AS TWOM
66 RESERVE GV(1) AS TWOM
67 RESERVE AM(1) AS TWOM BY TWOM
68 RESERVE AMINV(1) AS TWOM BY TWOM
69 FOR J=1 TO TWOM DO
70 LET BV(1)=0.0
71 LET GV(1)=0.0
72 FOR J=1 TO TWOM DO
73 LET AM(1,J)=0.0
74 LOOP **OVER J
75 LOOP **OVER I TO INITIALIZE
76 **FILL NON-ZERO ELEMENTS OF THE REDUCED STATE-TRANSITION MATRIX (AM)
77 **AND THE CONSTANT VECTOR (BV).
78 **
79 FOR J=1 TO TWOM DO
80 LET AM(1,J)=ML
81 LOOP **OVER COLUMNS
82 SUBTRACT MU*REAL.F(MPOP-1)*LAMBDA FROM AM(1,1)
83 ADD MU TO AM(1,4)
84 LET AM(2,1)=MU
85 LET AM(2,2)=MU-ML+LAMBDA
86 FOR N=2 TO MPOP-1 DO
87 LET L=2*N-1
88 LET AM(1,1-2)=(MPOP-N+1)*LAMBDA
89 LET AM(1,1)=MU-(MPOP-N)*LAMBDA
90 LET AM(1,1+3)=MU
91 LET AM(1+1,1-1)=AM(1,1-2)
92 LET AM(1+1,1)=MU
93 LET AM(1+1,1+1)=AM(1,1)
94 LET AM(1+1,1+1)=AM(1,1)
95 LOOP **OVER N
96 **FINALLY
97 LET AM(TWOM-1,TWOM-3)=LAMBDA
98 LET AM(TWOM-1,TWOM-1)=MU
99 LET AM(TWOM,TWOM-2)=LAMBDA
100 LET AM(TWOM,TWOM-1)=MU
101 LET AM(TWOM,TWOM)=MU
102 LET BV(1)=-ML
103 **OBTAIN THE INVERSE OF AM(1,1).
104 **
105 ** CALL MAT.INVERSE(AM(1,1),TWOM) YIELDING AMINV(1,1)
106 **
107 **OBTAIN THE SOLUTION VECTOR OF THE MATRIX EQUATION.
108 **
109 ** CALL MAT.VECMPLY(AMINV(1,1),BV(1),TWOM) YIELDING PV(1)
110 **
111 **SOLVE FOR THE EMPTY-SYSTEM PROBABILITY (P.NULL) AND CALCULATE
112 **THE STATE-PROBABILITY VECTOR (P.NSTATE).
113 **
114

```

ROUTINE FINITE.ME2.G
Options = SEQUENCE, ID, SUBCHK, XREF, NOEXPLIST, TRACES

```

115 LET ENSYS=0.0
116 LET VNSYS=0.0
117 LET ENG=0.0
118 LET ESO=0.0
119 LET SUM=0.0
120 FOR N=1 TO MPOP DO
121 LET I=2*N-1
122 LET P=NSTATE(N)=PV(I)+PV(I+1)
123 ADD P*NSTATE(N) TO SUM
124 ADD N*P*NSTATE(N) TO ENSYS
125 ADD N**2*P*NSTATE(N) TO VNSYS
126 ADD (N-1)*P*NSTATE(N) TO ENG
127 ADD (N-1)**2*P*NSTATE(N) TO ESO
128 LOOP **OVER NON-ZERO SYSTEM STATES
129 **CHECK VALIDITY OF PROBABILITY SUM.
130 IF SUM<0.0 OR SUM>1.0
131 PRINT 1 LINE WITH SUM THUS
132 IN ROUTINE FINITE.ME2.G. PARTIAL SUM OF STATE PROBABILITIES = *****
133 ERROR
134 OTHERWISE
135 LET P=NULL=1.0-SUM
136 LET P.SYS=FULL=SUM
137 LET ENG.CHECK=ENSYS-SUM **FOR A SINGLE SERVER
138 **CALCULATE THE VARIANCE AND STD DEV OF THE NUMBER IN THE SYSTEM.
139 **
140 LET VNSYS=VNSYS-ENSYS**2
141 LET SDNSYS=SQRT.F(VNSYS)
142 LET VNG=ESO-ENG**2
143 LET SDNG=SQRT.F(VNG)
144 **CALCULATE THE MEAN AND STANDARD DEVIATION OF BUSY SERVERS.
145 **
146 LET E.BUSY.SERVERS=1.0-P.NULL **FOR A SINGLE SERVER
147 LET VAR.BUSY.SERVERS=1.0-P.NULL-E.BUSY.SERVERS**2
148 LET SD.BUSY.SERVERS=SQRT.F(VAR.BUSY.SERVERS)
149 **CALCULATE MEAN WAITING TIMES USING LITTLE'S FORMULA.
150 **
151 LET E.SYS.WAIT=ENSYS/LAMBDA/(REAL.F(MPOP)-ENSYS)
152 LET E.G.WAIT=ESYS.WAIT-1.0/MUI
153 IF MPOP=1
154 RETURN
155 OTHERWISE **CALCULATE AND PRINT WAITING TIME DISTRIBUTIONS
156 **
157 **CALCULATE THE PROBABILITY THAT AN ARRIVAL FINDS THE SYSTEM
158 **IN THE N TH STATE (GV(N)).
159 **
160 LET NORM.CONST=MPOP*P.NULL
161 FOR N=1 TO MPOP-1 DO
162 ADD (MPOP-N)*P.NSTATE(N) TO NORM.CONST
163 LOOP **TO CALCULATE THE NORMALIZATION CONSTANT
164 LET G.NULL=REFOF*P.NULL/NORM.CONST
165 LET F.COST.WAIT=1.0-G.NULL
166 LET E.G.WAIT=G.NULL
167

```

ROUTINE FINITE-ME2-Q CACI SIMSCRIPT 11.5 for PRIME Systems, Release 2.1
Options = SEQUENCE,10,SUCHK,XREF,NOEXPLIST,TRACE3 06 JAN 1983 15:25:50

```

171 LET VQ.WAIT.GG=0.0
172 FOR NE1 TO MPOF-1 DO
173 LET I=2*N-1
174 LET QV(I)=(MPOP-N)*PV(I)/NORM.CONST
175 LET QV(I+1)=QV(I)+QV(I+1)
176 LET G.NSTATE(N)=QV(I)+QV(I+1)
177 ADD (I+1)*QV(I)+I*QV(I+1) TO EQ.WAIT.GG
178 ADD (I+1)*(I+2)*QV(I+1)+I*QV(I+1) TO VQ.WAIT.GG
179 LOOP ** OVER THE NUMBER OF ARRIVING CUSTOMERS
180 LET EG.WAIT.CHECK=EQ.WAIT.GG/MU
181 LET EG.WAIT.CHECK=EQ.WAIT.CHECK+1.0/MU1
182 LET EG.WAIT.GG=EG.WAIT.GG/MU/(1.0-Q.NULL)
183 LET VQ.WAIT.GG=VQ.WAIT.GG/MU/MU/(1.0-Q.NULL)-EQ.WAIT.GG**2
184 LET SUC.WAIT.GG=SUC.WAIT.GG/(VQ.WAIT.GG)
185 IF NQLE NE 1
186 RETURN
187 OTHERWISE
188 *LI*SKIP 4 LINES
189 PRINT 21 LINES WITH NSERVE,P,SYS,FULL,ENYS,SDNSYS,ENQ,SDNQ,E,BUSY,SERVERS,
190 SD,BUSY,SERVERS,ESYS,WAIT,EG,WAIT,EG,WAIT.GG,P.NULL,P.NULL,Q.NULL,Q.NULL
191 THUS
192 STATE OF THE FINITE SYSTEM IN STEADY STATE WITH GAMMA(2) SERVICE FOR ** SERVERS

```

PROBABILITY: SYSTEM IS FULL *****
 EXPECTED NUMBER IN THE SYSTEM *****
 STD DEV NUMBER IN THE SYSTEM *****
 EXPECTED NUMBER IN THE QUEUE *****
 STD DEV NUMBER IN THE QUEUE *****
 EXPECTED NUMBER BUSY SERVERS *****
 STD DEV NUMBER BUSY SERVERS *****
 MEAN WAITING TIME (MIN) IN SYSTEM *****
 MEAN WAITING TIME (MIN) IN QUEUE *****
 COND*AL MEAN TIME (MIN) IN QUEUE *****

PROBABILITY DISTRIBUTIONS FOR NUMBER OF CUSTOMERS IN THE SYSTEM

NO IN SYSTEM	UNCONDITIONAL			GIVEN ARRIVAL		
	PROB	CUML	PROB	PROB	CUML	PROB
0	*****	*****	*****	*****	*****	*****
192	*****	*****	*****	*****	*****	*****
193	*****	*****	*****	*****	*****	*****
194	*****	*****	*****	*****	*****	*****
195	*****	*****	*****	*****	*****	*****
196	*****	*****	*****	*****	*****	*****
197	*****	*****	*****	*****	*****	*****
198	*****	*****	*****	*****	*****	*****
199	*****	*****	*****	*****	*****	*****
200	*****	*****	*****	*****	*****	*****
201	*****	*****	*****	*****	*****	*****
202	*****	*****	*****	*****	*****	*****
203	*****	*****	*****	*****	*****	*****

ROUTINE FINITE.FL2.Q CACI SIMSCRIPT 11.5 for PRIME Systems, Release 2.1
Options = SEQUENCE, I, SUBCHK, XREF, NOEXPLIST, TRACE3 06 JAN 1983 15:25:50

```

204 LET DELT=MAX.F(1.0,TRUNC.F(EQ.WAIT.60/10.0))
205 LIT MAX=60
206 LIT MAX=60
207 SKIP 2 LINES
208 **PRINT HEADINGS FOR THE QUEUE WAITING TIME DISTRIBUTION.
209 **
210 PRINT 8 LINES WITH MPOP AND 0-NULL THUS
    CUMULATIVE AND CONDITIONAL CUM PROB DISTRIBUTIONS OF WAITING TIME IN QUEUE
    EXPONENTIAL INTERARRIVALS FOR EACH OF ** CUSTOMERS. ERLANG(2) SERVICE.

```

```

TIME CUML COND
(MIN) PROB PROB
-----
211 **START TIME LOOP.
212 **
213 **
214 **
215 **
216 **
217 **
218 **
219 **
220 **
221 **
222 **
223 **
224 **
225 **
226 **
227 **
228 **
229 **
230 **
231 **
232 **
233 **
234 **
235 **
236 **
237 **
238 **
239 **

    LET TM=1*DELT
    LET ARG=TM*MU
    LET CUM.DIST=1.0-EXP.F(-ARG)*(QV(1)*(1.0+ARG)+QV(2))
    LET SUMN=0.0
    FOR N=2 TO MPOP-1 DO
        LET ARGJ=1.0
        LIT JFACT=1.0
        FOR J=1 TO 2*(N-1) DO
            LET ARGJ=ARGJ*ARG
            LET JFACT=JFACT*J
        END FOR
        LET SUMN=SUMN+JFACT*ARGJ
    END FOR
    LET CUM.DIST=CUM.DIST+SUMN
    LET CON.DIST=(CUM.DIST-Q.NULL)/(1.0-Q.NULL)
    PRINT 1 LINE WITH TM, CUM.DIST, AND CON.DIST THUS

** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** *
** ** ** ** ** ** ** ** ** ** ** ** *
    LOOP **OVER 1 TIME SLIPS
    PRINT 10 LINES WITH EQ.WAIT.60, SDQ.WAIT.60, EQ.WAIT.CHECK,
    ESYS.WAIT.CHECK, Q.NULL, E.BUSY.SERVERS, SD.BUSY.SERVERS
    THUS

MEAN WAITING TIME IN QUEUE, GIVEN A WAIT ***** MINUTES
STD DEV WAITING TIME IN QUEUE, GIVEN A WAIT ***** MINUTES
MEAN WAITING TIME IN QUEUE (UNCONDITIONAL) ***** MINUTES
MEAN WAITING TIME IN SYS (UNCONDITIONAL) ***** MINUTES
PROBABILITY THAT AN ARRIVAL FINDS SYS EMPTY *****
MEAN NUMBER OF BUSY SERVERS *****
STD DEV NUMBER OF BUSY SERVERS *****

240 RETURN
241 *LC,IF NSEIVE>2

```

```

242 GO TO L3
243 OTHERWISE
244 **RESERVE ARRAYS FOR THE CASE: NSERVE=2
245 **
246 LET MAXM=3*MPOP-1
247 RESERVE BV(*) AS MAXM
248 RESERVE PV(*) AS MAXM
249 RESERVE QV(*) AS MAXM
250 RESERVE AM(*) AS MAXM BY MAXM
251 RESERVE AMINV(*) AS MAXM BY MAXM
252 **FILL THE REDUCED STATE-TRANSITION MATRIX (AM) AND THE CONSTANT
253 **VECTOR (BV).
254 **
255 FOR I=1 TO MAXM DO
256   LET BV(I)=0.0
257   FOR J=1 TO MAXM DO
258     LET AM(I,J)=0.0
259     LOOP **OVL I
260     FOR J=1 TO MAXM DO
261       LET AM(I,J)=AM(I,J)+MU
262       LET AM(I,J)=AM(I,J)-MU
263       LET AM(I,J)=AM(I,J)+MU
264       LET AM(I,J)=AM(I,J)-MU
265       LET AM(I,J)=AM(I,J)+MU
266       LET AM(I,J)=AM(I,J)-MU
267       LET AM(I,J)=AM(I,J)+MU
268       LET AM(I,J)=AM(I,J)-MU
269       LET AM(I,J)=AM(I,J)+MU
270       LET AM(I,J)=AM(I,J)-MU
271       LET AM(I,J)=AM(I,J)+MU
272       LET AM(I,J)=AM(I,J)-MU
273       LET AM(I,J)=AM(I,J)+MU
274       LET AM(I,J)=AM(I,J)-MU
275       LET AM(I,J)=AM(I,J)+MU
276       LET AM(I,J)=AM(I,J)-MU
277       LET AM(I,J)=AM(I,J)+MU
278       LET AM(I,J)=AM(I,J)-MU
279       LET AM(I,J)=AM(I,J)+MU
280       LET AM(I,J)=AM(I,J)-MU
281       LET AM(I,J)=AM(I,J)+MU
282       LET AM(I,J)=AM(I,J)-MU
283       LET AM(I,J)=AM(I,J)+MU
284       LET AM(I,J)=AM(I,J)-MU
285       LET AM(I,J)=AM(I,J)+MU
286       LET AM(I,J)=AM(I,J)-MU
287       LET AM(I,J)=AM(I,J)+MU
288       LET AM(I,J)=AM(I,J)-MU
289       LET AM(I,J)=AM(I,J)+MU
290       LET AM(I,J)=AM(I,J)-MU
291       LET AM(I,J)=AM(I,J)+MU
292       LET AM(I,J)=AM(I,J)-MU
293       LET AM(I,J)=AM(I,J)+MU
294       LET AM(I,J)=AM(I,J)-MU
295       LET AM(I,J)=AM(I,J)+MU
296       LET AM(I,J)=AM(I,J)-MU
297       LET AM(I,J)=AM(I,J)+MU
298       LET AM(I,J)=AM(I,J)-MU
299       LET AM(I,J)=AM(I,J)+MU

```

```

299 LET AM(1,3*MPOP-3)=-2.0*MU
300 LET AM(1,1)=-2.0*MU
301 ADD 1 TO I
302 LET AM(1,3*MPOP-4)=LAMBDA
303 LET AM(1,3*MPOP-2)=MU
304 LET AM(1,1)=-2.0*MU
305 LET BV(1)=ML
306
307 **OBTAIN THE INVERSE OF AM(**).
308
309 ** CALL MAT-INVERSE (AM(**),MAXM) YIELDING AMINV(**)
310
311 **OBTAIN THE SOLUTION VECTOR OF THE MATRIX EQUATION.
312
313 ** CALL MAT-VEC-MPY (AMINV(**),BV(**),MAXM) YIELDING PV(**)
314
315 **SOLVE FOR THE EMPTY-SYSTEM PROBABILITY (P.NULL) AND CALCULATE THE
316 **CUSTOMER STATE PROBABILITY VECTOR (P.NSTATE).
317
318 LET ENSYS=P.V(1)+P.V(2)
319 LET P.NSTATE(1)=ENSY
320 LET VNSYS=ENSY
321 LET ENQ=0.0
322 LET ESQ=0.0
323 LET SUM=ENSY
324 FOR N=2 TO MPOP DO
325 LET I=3*(N-1)
326 LET P.NSTATE(N)=P.V(I)+P.V(I+1)+P.V(I+2)
327 ADD P.NSTATE(N) TO SUM
328 ADD P.NSTATE(N) TO ENSYS
329 ADD (N-2)*P.NSTATE(N) TO ENQ
330 ADD (N-2)*P.NSTATE(N) TO ESQ
331 ADD N**2*P.NSTATE(N) TO VNSYS
332 LOOP **OVER THE OTHER NON-ZERO SYSTEM STATES
333
334 **CHECK VALIDITY OF THE PROBABILITY SUM.
335
336 IF SUM<0.0 OR SUM>1.0
337 PRINT 1 LINE WITH SUM THUS
338 IN ROUTINE FINITE.ME2.G. PARTIAL SUM OF STATE PROBABILITIES = *****
339 STOP
340 OTHERWISE
341 LET P.NULL=1.0-SUM
342 LET P.SYS.FULL=1.0-P.NULL-P.NSTATE(1) **FOR 2 SERVERS
343
344 **CALCULATE THE VARIANCE AND STANDARD DEV OF THE NUMBER IN THE SYSTEM.
345
346 LET VNSYS=VNSYS-ENSY**2
347 LET SDNSYS=SQRT.F(VNSYS)
348 LET VQ=ESQ-ENQ**2
349 LET SDNG=SQRT.F(VQ)
350
351 **CALCULATE THE MEAN AND STANDARD DEVIATION OF BUSY SERVERS.
352
353 LET E-BUSY.SERVERS=P.NSTATE(1)+2.0*(1.0-P.NULL-P.NSTATE(1))
354 LET VAR-BUSY.SERVERS=P.NSTATE(1)+4.0*(1.0-P.NULL-P.NSTATE(1))
355 IF-BUSY.SERVERS**2

```

CACI SIMSCRIPT II.5 for PRIME Systems, Release 2.1
06 JAN 1983 15:25:50

ROUTINE FINITE.ME2.Q
OPTIONS = SEQUENCE, I, SUBCHK, XREF, NOEXPLIST, TRACE3

```

355 LET SD.BUSY.SERVERS=SQRT.F(VAR.BUSY.SERVERS)
356
357 **CALCULATE MEAN WAITING TIME USING LITTLE'S FORMULA.
358
359 LET ESYS.WAIT=ENSY/LAMBDA/(REAL.F(MPOP)-ENSY)
360 LET EG.WAIT=ENG/LAMBDA/(REAL.F(MPOP)-ENSY)
361 IF MPOP LE 2
362 RETURN
363 OTHERWISE **CALCULATE AND PRINT CONDITIONAL PROB DISTRIBUTIONS
364
365 **CALCULATE THE PROBABILITY THAT AN ARRIVAL FINDS THE SYSTEM IN STATE N.
366
367 LET NORM.CONST=MPOP*P.NULL
368 FOR N=1 TO MPOP-1 DO
369 LET G.NULL=(MPOP-N)*P.NSTATE(N) TO NORM.CONST
370 LOOP **TO CALCULATE THE NORMALIZATION CONSTANT
371 LET G.NULL=XPOP*P.NULL/NORM.CONST
372 LET GV(1)=(MPOP-1)*PV(1)/NORM.CONST
373 LET GV(2)=(MPOP-1)*PV(2)/NORM.CONST
374 LET G.NSTATE(1)=GV(1)+GV(2)
375 FOR N=2 TO MPOP-1 DO
376 LET I=3*(N-1)
377 LET GV(I)=(MPOP-N)*PV(1)/NORM.CONST
378 LET GV(I+1)=(MPOP-N)*PV(I+1)/NORM.CONST
379 LET GV(I+2)=(MPOP-N)*PV(I+2)/NORM.CONST
380 LET G.NSTATE(N)=GV(I)+GV(I+1)+GV(I+2)
381 LOOP **OVER NUMBER OF ARRIVING CUSTOMERS
382 *L2 LET P.CUST.WAIT=1.0-G.NULL-G.NSTATE(1)
383 *L2 LET EG.WAIT=EG.WAIT/P.CUST.WAIT
384 *L2 LET SG.WAIT=EG.WAIT.GG **APPROXIMATELY
385 IF MORE NE 1 RETURN
386 OTHERWISE
387 GO TO L1
388 *L3 IF MPOP LE 3
389 PRINT 1 LINE WITH MPOP AND NSERVE THUS
390 IN FINITE.ME2.G. MPOP = **. NSERVE = **.
391 STOP
392 OTHERWISE
393 **RESERVE ARRAYS FOR THE CASE: NSERVE = 3.
394
395 LET MAXM=4*MPOP-3
396 RESERVE RV(*) AS MAXM
397 RESERVE PV(*) AS MAXM
398 RESERVE GV(*) AS MAXM
399 RESERVE AN(**) AS MAXM BY MAXM
400 RESERVE AMINV(**) AS MAXM BY MAXM
401
402 **FILL THE REDUCED STATE TRANSITION MATRIX (AM) AND THE CONSTANT VECTOR (RV).
403
404 FOR I=1 TO MAXM DO
405 LET BV(I)=0.0
406 FOR J=1 TO MAXM DO
407 LET AP(I,J)=0.0
408 LOOP **OVER J
409 LOOP **OVER I
410 FOR J=1 TO MAXM DO

```

ROUTINE FINITE.ME2.Q CACI SINGSCRIPT 11.5 for PRIME Systems, Release 2.1
Options = SEQUENCE, IL, SUMCHK, XREF, NOEXPLIST, TRACE3 06 JAN 1983 15:25:50

```

411 LET AM(1,J)=ML
412 LOOP **OVER J
413 LET AM(1,2)=AM(1,2)+MU
414 LET AM(2,1)=MU
415 LET AM(2,2)=((MPOP-1)*LAMBDA+MU)
416 LET AM(2,3)=2.0*MU
417 LET AM(3,1)=(MPOP-1)*LAMBDA
418 LET AM(3,3)=((MPOP-2)*LAMBDA+2.0*MU)
419 LET AM(3,7)=MU
420 LET AM(4,2)=(MPOP-1)*LAMBDA
421 LET AM(4,3)=2.0*MU
422 LET AM(4,4)=AM(3,3)
423 LET AM(4,8)=2.0*MU
424 LET AM(5,4)=MU
425 LET AM(5,5)=AM(4,4)
426 LET AM(5,9)=3.0*MU
427 LET AM(6,3)=(MPOP-2)*LAMBDA
428 LET AM(6,6)=((MPOP-3)*LAMBDA+3.0*MU)
429 LET AM(6,11)=MU
430 LET AM(7,4)=AM(6,3)
431 LET AM(7,6)=3.0*MU
432 LET AM(7,7)=AM(6,6)
433 LET AM(7,12)=2.0*MU
434 LET AM(8,5)=AM(7,4)
435 LET AM(8,7)=2.0*MU
436 LET AM(8,8)=AM(7,7)
437 LET AM(8,13)=3.0*MU
438 LET AM(9,8)=MU
439 LET AM(9,9)=AM(8,8)
440 FOR N=4 TO MPOP-1 DO
441 I=4*N-6 **AS THE ROW INDEX
442 LET AM(1,4*N-10)=(MPOP-N+1)*LAMBDA
443 LET AM(1,1)=(MPOP-N)*LAMBDA+3.0*MU)
444 LET AM(1,4*N-1)=MU
445 ADD 1 TO I **I=4*N-5
446 LET AM(1,4*N-9)=(MPOP-N+1)*LAMBDA
447 LET AM(1,1-1)=3.0*MU
448 LET AM(1,1)=AM(1-1,1-1)
449 LET AM(1,4*N-2)=2.0*MU
450 ADD 1 TO I **I=4*N-4
451 LET AM(1,4*N-8)=(MPOP-N+1)*LAMBDA
452 LET AM(1,1-1)=2.0*MU
453 LET AM(1,1)=AM(1-1,1-1)
454 LET AM(1,4*N+1)=3.0*MU
455 ADD 1 TO I **I=4*N-3
456 LET AM(1,4*N-7)=(MPOP-N+1)*LAMBDA
457 LET AM(1,1-1)=MU
458 LET AM(1,1)=AM(1-1,1-1)
459 LOOP **OVER THE NUMBER OF CUSTOMERS IN THE SYSTEM
460 **
461 **EQUATIONS FOR THE LAST FOUR STATES.
462 **
463 LET I=4*MPOP-6
464 LET AM(1,4*MPOP-10)=LAMBDA
465 LET AM(1,1)=3.0*MU
466 ADD 1 TO I **I=4*MPOP-5
467 LET AM(1,4*MPOP-9)=LAMBDA

```

ROUTINE FINITE.ME2.0
Options = SEQUENCE,10,SUBCHK,XREF,NOEXPLIST,TRACE3

```

468 LET AM(I,I-1)=3.0*MU
469 LET AM(I,I)=AM(I-1,I-1)
470 ADD 1 TO I
471 LET AM(I,I-1)=LAMBDA
472 LET AM(I,I)=2.0*MU
473 LET AM(I,I-1)=AM(I-1,I-1)
474 ADD 1 TO I
475 LET AM(I,I-1)=LAMBDA
476 LET AM(I,I)=MU
477 LET AM(I,I)=AM(I-1,I-1)
478 LET BV(1)=ML
479
480 **OBTAIN THE INVERSE OF AM(**).
481 **
482 ** CALL MAT.INVERSE(AM(**),MAXM) YIELDING AMINV(**)
483 **
484 ** OBTAIN THE SOLUTION VECTOR OF THE MATRIX EQUATION.
485 **
486 ** CALL MAT.VECMPLY(AMINV(**),BV(**),MAXM) YIELDING PV(**)
487 **
488 ** SOLVE FOR THE EMPTY SYSTEM STATE PROBABILITY (P=NULL) AND CALCULATE
489 ** THE CUSTOMER STATE PROBABILITY VECTOR (P.NSTATE(**)).
490 **
491 LET P.NSTATE(1)=PV(1)+PV(2) **FOR 1 CUSTOMER
492 LET P.NSTATE(2)=PV(3)+PV(4)+PV(5)
493 LET VNSYS=P.NSTATE(1)+2.0*P.NSTATE(2)
494 LET ENG=0.0
495 LET ESC=0.0
496 LET SUM=P.NSTATE(1)+P.NSTATE(2)
497 FOR N=3 TO NPOP DO
498 LET I=N-6
499 LET P.NSTATE(N)=PV(1)+PV(I+1)+PV(I+2)+PV(I+3)
500 ADD P.NSTATE(N) TO SUM
501 ADD N*P.NSTATE(N) TO VNSYS
502 ADD (N-3)*P.NSTATE(N) TO ENG
503 ADD (N-3)**2*P.NSTATE(N) TO ESC
504 ADD N**2*P.NSTATE(N) TO VNSYS
505
506 LOOP **OVER THE OTHER NON-ZERO STATES
507
508 **CHECK VALIDITY OF THE PROBABILITY SUM.
509
510 IF SUM<1.0 OR SUM>1.0
511 PRINT LINE WITH SUM THUS
512 IN ROUTINE FINITE.ME2.G. PARTIAL SUM OF STATE PROBABILITIES = *****
513 STOP
514 OTHERWISE
515 LET P=NULL=1.0-SUM
516 LET P.SYS.FULL=1.0-P.NULL-P.NSTATE(1)-P.NSTATE(2) **FOR 3 SERVERS
517
518 **CALCULATE THE VARIANCE AND STANDARD DEVIATION OF NUMBER IN THE SYSTEM.
519
520 LET VNSYS=VNSYS-ENG**2
521 LET VNSYS=SQRT.F(VNSYS)
522 LET SUM=SUM.F(VNSYS)
523

```

```

524 **CALCULATE THE MEAN AND STANDARD DEVIATION OF BUSY SERVERS.
525 **
526 LET E.BUSY.SERVERS=P.NSTATE(1)+2.0*P.NSTATE(2)+3.0*(1.0-P.NULL
527 -P.NSTATE(1))-P.NSTATE(2))
528 LET VAR.BUSY.SERVERS=P.NSTATE(1)+4.0*P.NSTATE(2)+9.0*(1.0-P.NULL
529 -P.NSTATE(1))-P.NSTATE(2))-E.BUSY.SERVERS**2
530 LET SD.BUSY.SERVERS=SQRT(VAR.BUSY.SERVERS)
531 **CALCULATE MEAN WAITING TIME USING LITTLE'S FORMULA.
532 **
533 LET ESYS.WAIT=ENQ/LAMBOA/(REAL.F(MPOP)-ENQSYS)
534 LET EQ.WAIT=ENQ/LAMBOA/(REAL.F(MPOP)-ENQSYS)
535 IF MPOP=3
536 RETURN
537 OTHERWISE **CALCULATE AND PRINT CONDITIONAL PROB DISTRIBUTIONS
538 **
539 **CALCULATE THE PROBABILITY THAT AN ARRIVAL FINDS THE SYSTEM IN STATE N.
540 **
541 LET NORM.CONST=MPOP*P.NULL
542 FOR N=1 TO MPOP-1 DO
543 LOOP **TO CALCULATE THE NORMALIZATION CONSTANT
544 LET G.NULL=MPOP*P.NULL/NORM.CONST
545 LET QV(1)=(MPOP-1)*PV(1)/NORM.CONST
546 LET QV(2)=(MPOP-2)*PV(2)/NORM.CONST
547 LET QV(3)=(MPOP-3)*PV(3)/NORM.CONST
548 LET QV(4)=(MPOP-4)*PV(4)/NORM.CONST
549 LET QV(5)=(MPOP-5)*PV(5)/NORM.CONST
550 LET G.NSTATE(1)=QV(1)+QV(2)
551 LET G.NSTATE(2)=QV(1)+QV(2)
552 LET G.NSTATE(3)=QV(3)+QV(4)+QV(5)
553 FOR N=3 TO MPOP-1 DO
554 LET I=4-N
555 LET GV(I)=(MPOP-N)*PV(I)/NORM.CONST
556 LET GV(I+1)=(MPOP-N)*PV(I+1)/NORM.CONST
557 LET GV(I+2)=(MPOP-N)*PV(I+2)/NORM.CONST
558 LET GV(I+3)=(MPOP-N)*PV(I+3)/NORM.CONST
559 LET G.NSTATE(N)=GV(I)+GV(I+1)+GV(I+2)+GV(I+3)
560 **OVER NUMBER OF ARRIVING CUSTOMERS
561 LET P.CUST.WAIT=1.0-G.NSTATE(1)-G.NSTATE(2)
562 GO TO L2
563 END **OF ROUTINE FINITE.ME2.Q
564

```

C R O S S - R E F E R E N C E

NAME	TYPE	WORD	11	(2-D)	MODE	LINE NUMBERS OF REFERENCES									
AM	RECURSIVE VARIABLE	WORD	11	(2-D)	DOUBLE	52	67*	73	81	83*	84*	85	86	87	88
						86	89	90	91	92*	93	94*	95	96	97
						97	98	99	100	101	106	107	108	109	110
						260	264	266*	267	268	269	270	271	272	273
						271	272	273	274	275*	276	277*	278	279	280
						278*	281	282	283	285*	286	287*	288	289	290
						288	290	291	292	295	296	297*	298	299	300
						299	300	302	303	304	309	310	311	312	313
						407	411	413*	414	415	416	417	418	419	420
						418	419	420	421	422*	423	424	425	426	427
						425*	426	427	428	429	430*	431	432	433	434
						432*	433	434*	435	436*	437	438*	439	440	441
						439*	442	443	444	446	447	448*	449	450	451
						449	451	452	453*	454	455	456	457	458	459
						458*	464	465	467	468	469*	470	471	472	473
						472	473*	475	476	477*	482	483	484	485	486
AMINV	RECURSIVE VARIABLE	WORD	12	(2-D)	DOUBLE	52	67*	73	81	83*	84*	85	86	87	88
ARG	RECURSIVE VARIABLE	WORD	62		DOUBLE	400*	482	486	487	488	489	490	491	492	493
ARGJ	RECURSIVE VARIABLE	WORD	68		DOUBLE	216	217*	224	228	230	233	234	235	236	237
AV	RECURSIVE VARIABLE	WORD	7	(1-D)	DOUBLE	221	224*	226	228*	230	233	234	235	236	237
						51	64*	70	102	110	110	111	112	113	114
CUN.DIST	RECURSIVE VARIABLE	WORD	74		DOUBLE	305	313	315	316	317	318	319	320	321	322
CUM.DIST	RECURSIVE VARIABLE	WORD	53		DOUBLE	234	235	197	217	233*	234	235	236	237	238
CUM.DIST.CUND	RECURSIVE VARIABLE	WORD	55		DOUBLE	192	195*	197	217	233*	234	235	236	237	238
DELT	RECURSIVE VARIABLE	WORD	57		DOUBLE	204	215	216	217	233*	234	235	236	237	238
E.RUSY.SERVERS	RECURSIVE VARIABLE	WORD	16		DOUBLE	526	529	150	189	237	238	239	240	241	242
ENG	ARGUMENT	NO.	11		DOUBLE	2	117	126*	144	189	237	238	239	240	241
ENG.CHECK	RECURSIVE VARIABLE	WORD	34		DOUBLE	347	360	495	503*	521	535	536	537	538	539
ENSYS	ARGUMENT	NO.	9		DOUBLE	138	149	150	189	237	238	239	240	241	242
EG.WAIT	RECURSIVE VARIABLE	WORD	13		DOUBLE	318	319	124*	138	142	155*	156	157	158	159
EG.WAIT.CHECK	RECURSIVE VARIABLE	WORD	48		DOUBLE	360	493	502*	519	534*	535	536	537	538	539
EG.WAIT.CO	RECURSIVE VARIABLE	WORD	14		DOUBLE	2	156	189	360	534*	535	536	537	538	539
ESG	RECURSIVE VARIABLE	WORD	19		DOUBLE	180	181	177*	180	182*	183	184	185	186	187
ESYS.WAIT	RECURSIVE VARIABLE	WORD	12		DOUBLE	204	237*	383	384	182*	183	184	185	186	187
ESYS.WAIT.CHECK	RECURSIVE VARIABLE	WORD	50		DOUBLE	118	127*	144	322	330*	347	348	349	350	351
EXP.F	ROUTINE	WORD	12		DOUBLE	504*	521	156	189	359	374	375	376	377	378
FINITE.ME2.G	ROUTINE	WORD	1		INTEGER	181	182	156	189	359	374	375	376	377	378
I	RECURSIVE VARIABLE	WORD	1		INTEGER	217	233	144	322	330*	347	348	349	350	351
						50	69*	70	71	73	88	89	90	91	92
						90*	91*	92*	93*	94*	121	122*	123	124	125
						173	174	175*	176*	177*	178*	179*	180	181	182
						215	257*	258	260	280	281	282*	283	284	285
						285	287*	288	289	287*	288	289*	290	291	292
						290	291	292*	294	295	296*	297*	298	299	300
						298	299	300*	301*	302	303*	304*	305	306	307
						325	326*	327	328	329*	330*	331*	332	333	334
						404*	405	407	441	442	443	444	445	446	447

ROUTINE FINJIE.ME2.0
Options = SEQUENCE, I, J, SURCHK, XREF, NOEXPLIST, TRACE3
CACI SIMSCRIPT II.5 for FRIME Systems, Release 2.1 06 JAN 1983 15:25:50

J	RECURSIVE VARIABLE	WORD	2	INTEGER	445* 446* 447* 448* 449* 450* 451* 452* 453* 454* 455* 456* 457* 458* 459* 460* 461* 462* 463* 464* 465* 466* 467* 468* 469* 470* 471* 472* 473* 474* 475* 476* 477* 478* 479* 480* 481* 482* 483* 484* 485* 486* 487* 488* 489* 490* 491* 492* 493* 494* 495* 496* 497* 498* 499* 500* 501* 502* 503* 504* 505* 506* 507* 508* 509* 510* 511* 512* 513* 514* 515* 516* 517* 518* 519* 520* 521* 522* 523* 524* 525* 526* 527* 528* 529* 530* 531* 532* 533* 534* 535* 536* 537* 538* 539* 540* 541* 542* 543* 544* 545* 546* 547* 548* 549* 550* 551* 552* 553* 554* 555* 556* 557* 558* 559* 560* 561* 562* 563* 564* 565* 566* 567* 568* 569* 570* 571* 572* 573* 574* 575* 576* 577* 578* 579* 580* 581* 582* 583* 584* 585* 586* 587* 588* 589* 590* 591* 592* 593* 594* 595* 596* 597* 598* 599* 600* 601* 602* 603* 604* 605* 606* 607* 608* 609* 610* 611* 612* 613* 614* 615* 616* 617* 618* 619* 620* 621* 622* 623* 624* 625* 626* 627* 628* 629* 630* 631* 632* 633* 634* 635* 636* 637* 638* 639* 640* 641* 642* 643* 644* 645* 646* 647* 648* 649* 650* 651* 652* 653* 654* 655* 656* 657* 658* 659* 660* 661* 662* 663* 664* 665* 666* 667* 668* 669* 670* 671* 672* 673* 674* 675* 676* 677* 678* 679* 680* 681* 682* 683* 684* 685* 686* 687* 688* 689* 690* 691* 692* 693* 694* 695* 696* 697* 698* 699* 700* 701* 702* 703* 704* 705* 706* 707* 708* 709* 710* 711* 712* 713* 714* 715* 716* 717* 718* 719* 720* 721* 722* 723* 724* 725* 726* 727* 728* 729* 730* 731* 732* 733* 734* 735* 736* 737* 738* 739* 740* 741* 742* 743* 744* 745* 746* 747* 748* 749* 750* 751* 752* 753* 754* 755* 756* 757* 758* 759* 760* 761* 762* 763* 764* 765* 766* 767* 768* 769* 770* 771* 772* 773* 774* 775* 776* 777* 778* 779* 780* 781* 782* 783* 784* 785* 786* 787* 788* 789* 790* 791* 792* 793* 794* 795* 796* 797* 798* 799* 800* 801* 802* 803* 804* 805* 806* 807* 808* 809* 810* 811* 812* 813* 814* 815* 816* 817* 818* 819* 820* 821* 822* 823* 824* 825* 826* 827* 828* 829* 830* 831* 832* 833* 834* 835* 836* 837* 838* 839* 840* 841* 842* 843* 844* 845* 846* 847* 848* 849* 850* 851* 852* 853* 854* 855* 856* 857* 858* 859* 860* 861* 862* 863* 864* 865* 866* 867* 868* 869* 870* 871* 872* 873* 874* 875* 876* 877* 878* 879* 880* 881* 882* 883* 884* 885* 886* 887* 888* 889* 890* 891* 892* 893* 894* 895* 896* 897* 898* 899* 900* 901* 902* 903* 904* 905* 906* 907* 908* 909* 910* 911* 912* 913* 914* 915* 916* 917* 918* 919* 920* 921* 922* 923* 924* 925* 926* 927* 928* 929* 930* 931* 932* 933* 934* 935* 936* 937* 938* 939* 940* 941* 942* 943* 944* 945* 946* 947* 948* 949* 950* 951* 952* 953* 954* 955* 956* 957* 958* 959* 960* 961* 962* 963* 964* 965* 966* 967* 968* 969* 970* 971* 972* 973* 974* 975* 976* 977* 978* 979* 980* 981* 982* 983* 984* 985* 986* 987* 988* 989* 990* 991* 992* 993* 994* 995* 996* 997* 998* 999* 1000*
---	--------------------	------	---	---------	---

NORM.CONST	RECURSIVE VARIABLE	WORD	42		DOUBLE	166*	172*	173*	174*	175	176	194*
						195	196	197*	219*	223	229	231*
						279*	280	281*	282	283	285*	286
						327	328*	329*	330*	331*	332*	325
NSERVE	RECURSIVE VARIABLE	WORD	4		INTEGER	327	376	377	378	379	380*	369*
						375*	441	442*	443	444	446*	440*
						454	456*	458*	499	500*	501*	451*
						556	504*	505*	543*	544*	545*	502*
P.CUST.WAIT P.NSTATE	RECURSIVE VARIABLE	WORD	42		DOUBLE	166*	164	166*	174	175	176	369*
						371	372	373	377	378	379	542
						556	557	558	559	560	561	551
						389	50	58	61	189	201	241
P.NULL	ARGUMENT	NO.	4		INTEGER	1	1	50	58	189	201	241
	ARGUMENT	NO.	8		DOUBLE	169	169	382	383	562	124	125
	ARGUMENT	NO.	18	(1-D)	DOUBLE	127	127	166*	195	197	319	326
						327	328	329	330	331*	341*	352*
P.SYS.FULL PV	RECURSIVE VARIABLE	WORD	6		DOUBLE	327	328	329	330	331*	341*	352*
						500	501	502	503	504	505	515*
						526*	527*	528*	529*	544	545	515*
						514	515	526	528	544	545	515*
Q.NSTATE	RECURSIVE VARIABLE	WORD	7		DOUBLE	192	136	149	150	164	168	189*
						514	515	526	528	544	545	371
	RECURSIVE VARIABLE	WORD	8	(1-D)	DOUBLE	137	137	189	195	197	319	326
						51	51	110	122*	174	377	378
Q.NULL	RECURSIVE VARIABLE	WORD	10		DOUBLE	313	318*	326*	372	373	374	380
						379	379	486	491*	492*	500*	547
						548	549	550	551	556	557	558
						559	559	560	561	562*	563	564
REAL.F S.D.BUSY.SERVERS	RECURSIVE VARIABLE	WORD	10	(1-D)	DOUBLE	57*	57*	176	196	197	374	380
						552	552	553	560	562*	563	564
	RECURSIVE VARIABLE	WORD	44		DOUBLE	169	169	182	183	189*	193	210
						234*	237	371	382	383	384	385
SUM	RECURSIVE VARIABLE	WORD	9	(1-D)	DOUBLE	51	66*	71	174	175	176*	177*
						178*	217*	231*	250*	251*	252*	253*
						377	378	379	380*	381*	382*	383*
						549	550	551	552*	553*	554*	555
SUMEJ	ROUTINE	NO.	17		INTEGER	558	559	560	561	562*	563	564
	ARGUMENT	WORD	38		DOUBLE	151	151	189	193	194	195	196
	ARGUMENT	WORD	10		DOUBLE	145	145	189	193	194	195	196
	ROUTINE	NO.	15		DOUBLE	143	143	189	193	194	195	196
SUMEJ	RECURSIVE VARIABLE	WORD	21		DOUBLE	520	522*	523*	524*	525*	526*	527*
						119	123*	124*	125*	126*	127*	128*
						510*	511	512	513	514	515	516
						220	221	222	223	224	225	226
SUMEJ	RECURSIVE VARIABLE	WORD	66		DOUBLE	230	231	232	233	234	235	236
						237	238	239	240	241	242	243
						244	245	246	247	248	249	250
						251	252	253	254	255	256	257
SUMEJ	RECURSIVE VARIABLE	WORD	60		INTEGER	258	259	260	261	262	263	264
						265	266	267	268	269	270	271
						272	273	274	275	276	277	278
						279	280	281	282	283	284	285
SUMEJ	RECURSIVE VARIABLE	WORD	3		INTEGER	286	287	288	289	290	291	292
						293	294	295	296	297	298	299
						300	301	302	303	304	305	306
						307	308	309	310	311	312	313

UIR.N	IMPLIED SUBSCRIPT	SYS	4	INTEGER	69	72	80	97*	98*	99*	100*
VAR.BUSY.SERVERS	RECURSIVE VARIABLE	WORD	40	DOUBLE	101*	106	110	355	528	530	
VNG	RECURSIVE VARIABLE	WORD	36	DOUBLE	108	206	353	348	521	522	
VNSYS	RECURSIVE VARIABLE	WORD	17	DOUBLE	144	151	347	143	320	331*	345*
VB.WAIT.60	RECURSIVE VARIABLE	WORD	46	DOUBLE	116	125*	142*	519*	520		
					346	494	505*	184			
					171	178*	183*				

```

1 ROUTINE FOR MAT.MPY (AM, BM, NELMTS) YIELDING CM
2
3 **ROUTINE TO MULTIPLY SQUARE MATRICES AM AND BM OF NELMTS BY NELMTS,
4 **YIELDING PRODUCT MATRIX CM.
5
6 DEFINE I, J, K, NELMTS AS INTEGER VARIABLES
7 DEFINE AM, BM, AND CM AS REAL, 2-DIMENSIONAL ARRAYS
8 RESERVE AM(***), AS NELMTS BY NELMTS
9 RESERVE BM(***), AS NELMTS BY NELMTS
10 RESERVE CM(***), AS NELMTS BY NELMTS
11 FOR I=1 TO NELMTS DO
12   FOR J=1 TO NELMTS DO
13     LET CM(I,J)=0.0
14     FOR K=1 TO NELMTS DO
15       ADD AM(I,K)*BM(K,J) TO CM(I,J)
16     LOOP **OVER K
17   LOOP **OVER J
18 LOOP **OVER I
19 RETURN
20 END **OF ROUTINE MAT.MPY

```

C R O S S - R E F E R E N C E					L I N E N U M B E R S O F R E F E R E N C E S				
NAME	TYPE	NO.		MODE					
AM	ARGUMENT	NO.	1	(2-D)	DOUBLE	1	7	8*	15
BM	ARGUMENT	NO.	2	(2-D)	DOUBLE	1	7	9*	15
CM	ARGUMENT	NO.	4	(2-D)	DOUBLE	1	7	10*	15*
I	RECURSIVE VARIABLE	WORD	1		INTEGER	6	11*	13	15*
J	RECURSIVE VARIABLE	WORD	1		INTEGER	6	12*	13	15*
K	RECURSIVE VARIABLE	WORD	3		INTEGER	6	14*	15*	
MAT.MPY	ROUTINE	NO.	3		INTEGER	1	6	8*	9*
NELMTS	ARGUMENT	NO.	3		INTEGER	14			11
									12

```

1 ROUTINE FOR MAT.VEC.MPY (AM, BV, NELMTS) YIELDING CV
2
3 **ROUTINE TO MULTIPLY THE SQUARE MATRIX AM, OF NELMTS BY NELMTS,
4 **BY THE VECTOR BV (NELMTS BY 1), YIELDING THE VECTOR CV (NELMTS BY 1).
5
6 DEFINE I, J, K, NELMTS AS INTEGER VARIABLES
7 DEFINE BV AND CV AS REAL, 1-DIMENSIONAL ARRAYS
8 DEFINE AM AS A REAL, 2-DIMENSIONAL ARRAY
9 RESERVE AM(**,*) AS NELMTS BY NELMTS
10 RESERVE BV(*) AS NELMTS
11 RESERVE CV(*) AS NELMTS
12 FOR I=1 TO NELMTS DO
13   LET CV(I)=0.0
14   FOR K=1 TO NELMTS DO
15     ADD AM(I,K)*BV(K) TO CV(I)
16   LOOP **OVER K
17 LOOP **OVER I
18 RETURN
19 END **OF ROUTINE MAT.VEC.MPY

```

C R O S S - R E F E R E N C E

NAME	TYPE	NO.	MODE	LINE NUMBERS OF REFERENCES
AM	ARGUMENT	NO.	1 (2-D)	1 8 9*
BV	ARGUMENT	NO.	2 (1-D)	1 7 10*
CV	ARGUMENT	NO.	4 (1-D)	1 7 11*
I	RECURSIVE VARIABLE	WORD	1	6 12*
J	RECURSIVE VARIABLE	WORD	2	6 14*
K	ROUTINE	WORD	3	6 15*
MAT.VEC.MPY	ROUTINE	NO.	3	1 6 9*
NELMTS	ARGUMENT	NO.	3	1 6 9*

Options = SEQUENCE, IC, SUBCHK, XREF, NOEXPLIST, TRACE3
 CAC1 SIMSCRIPT II.5 for PRIME Systems, 06 JAN 1983 15:25:50

```

1 ROUTINE FOR MAT.INVERSE (AM, N) YIELDING BM
2 **
3 **ROUTINE TO OBTAIN THE INVERSE OF THE N BY N MATRIX AM VIA THE
4 **COMPACT FORM OF THE GAUSS-JORDAN METHOD. INVERSE IS RETURNED
5 **AS BM. AM IS LEFT UNCHANGED.
6 **
7 DEFINE I, J, K, N AS INTEGER VARIABLES
8 DEFINE AM AND EM AS REAL, 2-DIMENSIONAL ARRAYS
9 RESERVE AM(**) AS N BY N
10 RESERVE BM(**) AS N BY N
11 **COPY AM INTO EM. BM IS USED FOR GAUSSIAN REDUCTION.
12 **
13 FOR I=1 TO N DO
14   FOR J=1 TO N DO
15     LET BM(I,J)=AM(I,J)
16   LOOP **OVER J
17   LOOP **OVER I
18   FOR I=1 TO N DO
19     LET PEM(I,1)
20   IF PEM=0
21     PRINT 2 LINES WITH 1 THUS
22     IN ROUTINE MAT.INVERSE. THE **TH DIAGONAL ELEMENT IS ZERO.
23   THE MATRIX CANNOT BE INVERTED.
24   STOP
25 OTHERWISE
26 LET BM(I,1)=1.0
27 FOR J=1 TO N DO
28   LET BM(I,J)=BM(I,J)/P
29   LOOP **OVER J
30   FOR J=1 TO N DO **THE SECOND J-LOOP
31     IF J=1
32       GO TO EOJ **END OF J-LOOP
33     OTHERWISE
34       LET PEM(J,1)
35       LET PM(J,1)=0.0
36       FOR K=1 TO N DO
37         SUBTRACT P*BM(I,K) FROM BM(J,K)
38       LOOP **OVER K
39     EOJ **LOOP **OVER J
40   LOOP **OVER I
41   RETURN
42 END **OF ROUTINE MAT.INVERSE

```

C R O S S - R E F E R E N C E

NAME	TYPE	NO. NO.	MODE	LINE NUMBERS OF REFERENCES
AM	ARGUMENT	1	(2-D) DOUBLE	1
BM	ARGUMENT	3	(2-D) DOUBLE	16
EUJ				8
I				10*
J				36*
K				34
MAT.INVERSL				38
N				14*
P				30
				15*
				34
				35*
				7
				29
				21
				9*
				10*
				36*
				16*
				33
				16*
				36*
				10*
				35
				27
				33
				26
				20
				14
				36
				19
				15
				25
				27*
				20*
				36
				27*
				22
				29*
				25*
				30

CACI SIMSCKRPT II.5 for PRIME Systems, Release 2.1
06 JAN 1983 15:25:50

Options = SEQUENCE,IN,SURCHK,XREF,NOEXPLIST,TRACES

```

1 ROUTINE FOR BEST SERVICE GIVEN LAMBDA, MU, MPOP, NSERVE, PIOLIM, FOLIM, IPRINT,
2 MODE YIELDING, F, NULL, P, SYS, FULL, P, MACH, WAIT, ENO, DOWN, SDNO, DOWN,
3 ENO, QUEUED, ESYS, WAIT, EQ, WAIT, GO, SDQ, WAIT, GO, E, BUSY, SERVERS,
4 SD, BUSY, SERVERS, AND F, NSTATE
5
6 ** AT THE OPTION OF THE USER, THIS ROUTINE ITERATIVELY DETERMINES THE
7 ** MINIMUM NUMBER OF REPAIRMEN (>= NSERVE) REQUIRED TO PROVIDE A GIVEN
8 ** QUALITY OF MAINTENANCE SERVICE. TWO MEASURES OF QUALITY ARE USED:
9 ** (A) THE PROBABILITY THAT THE FRACTION OF MACHINES DOWN EXCEEDS 10%,
10 ** (B) THE EXPECTED FRACTION OF MACHINES QUEUING FOR MAINTENANCE SERVICE.
11 ** AT LEAST ONE OF THESE MEASURES MUST SATISFY ITS ASSOCIATED REQUIREMENT.
12 ** THE STOCHASTIC MODEL ASSUMES EXPONENTIAL TIMES BETWEEN FAILURES AND
13 ** EXPONENTIAL SERVICE TIMES WITH COMMON MTR FOR ALL MACHINES.
14 ** INPUT:
15 ** LAMBDA ARRIVAL RATE FOR SERVICE PER INDIVIDUAL MACHINE.
16 ** MU SERVICE RATE PER SERVER.
17 ** MPOP MACHINE POPULATION SIZE.
18 ** NSERVE NUMBER OF SERVICE PERSONNEL. MACHINES ARE DOWN.
19 ** PIOLIM LIMIT ON PROB THAT 10% OF MACHINES THAT WAIT FOR SERVICE.
20 ** FOLIM LIMIT ON FRACTION OF THE MACHINES THAT WAIT FOR SERVICE.
21 ** IPRINT INTEGER SWITCH TO PRINT FROM THE SUBROUTINE. PRINTING
22 ** OCCURS WHEN IPRINT=1.
23 ** MODE ROUTINE MODE OF OPN. IF MODE="CALCULATE", STATISTICS
24 ** ARE CALCULATED FOR THE SYSTEM HAVING NSERVE REPAIRMEN.
25 ** NO OTHER SYSTEMS ARE CONSIDERED. IF MODE="OPTIMIZE",
26 ** THE ROUTINE WILL SELECT AN OPTIMUM NUMBER OF REPAIRMEN
27 ** TO SATISFY THE CONSTRAINTS.
28
29 ** OUTPUT: PRINTS THE OPTIMUM NUMBER OF REPAIRMEN (SERVERS) AND THE
30 ** ROUTINE PRINTS PERFORMANCE STATISTICS INCLUDING STATE PROBABILITIES.
31
32 ** P, NULL PROBABILITY THE MAINTENANCE SYSTEM IS EMPTY. I.E.,
33 ** P, SYS, FULL PROBABILITY THE MAINTENANCE SYSTEM IS FULL. I.E.,
34 ** P, MACH, WAIT PROBABILITY THAT AN ARRIVAL FOR MAINT MUST QUEUE.
35 ** ENO, DOWN EXPECTED (MEAN VALUE) NUMBER OF MACHINES DOWN.
36 ** SDNO, DOWN STANDARD DEVIATION OF NUMBER OF MACHINES DOWN.
37 ** ENO, QUEUED EXPECTED NUMBER OF MACHINES IN THE MAINT QUEUE.
38 ** ESYS, WAIT EXPECTED WAIT (MINUTES) IN THE MAINTENANCE SYSTEM.
39 ** EQ, WAIT EXPECTED WAIT (MINUTES) IN THE MAINTENANCE QUEUE.
40 ** EG, WAIT GO, EXPECTED WAIT, GIVEN A MACHINE MUST WAIT FOR SERVICE.
41 ** SDQ, WAIT GO, STD DEV OF BUSY SERVERS.
42 ** L, BUSY, SERVERS MEAN NUMBER OF BUSY SERVERS.
43 ** SD, BUSY, SERVERS STD DEV NUMBER OF BUSY SERVERS.
44 ** P, NSTATE VECTOR IN WHICH THE N TH ELEMENT IS THE PROBABILITY
45 ** THAT N MACHINES ARE DOWN.
46
47 ** MU, GENOUS VARIABLES:
48 ** GV VECTOR IN WHICH THE N TH ELEMENT IS THE PROBABILITY THAT
49 ** AN ARRIVAL FOR SERVICE FINDS THE SYSTEM IN THE N TH STATE.
50 ** P, NULL PROBABILITY THAT A MACHINE NEEDING SERVICE FINDS THE
51 ** SYSTEM EMPTY.
52
53 ** DEFINE MODE AS A TEXT VARIABLE
54 ** DEFINE I, IPRINT, J, N, MAX, MPOP, N, NSERVE, AND N, TEN, PCT AS INTEGER VARIABLES
55 ** DEFINE P, NSTATE AND GV AS REAL, 1-DIMENSIONAL ARRAYS
56 ** LET RELAND, DA, FC
57 ** LET N, TEN, PCT = MAX, F(1, INT, F(0, 10, REAL, F(MPOP)))

```

```

58 RESERVE P.NSTAIL(*) AS MPOP
59 RESERVE QV(*) AS MPOP
60
61 **
62 ** ITERATION OVER THE NUMBLR SERVING STARTS HERE.
63 **
64 ** LET C=NSERVE
65 ** IF NSERVE GE MPOP
66 ** RETURN
67 ** OTHERWISE
68 ** RHO=V/C
69 ** ENO=DOWNEN/2
70 ** E.BUSY.SERVERS=0.0
71 ** VAR.BUSY.SERVERS=0.0
72 ** SUM=1.0
73 ** LET GSUM=0
74 ** LET IFACTORIAL=1.0
75 ** IF NSERVE=1
76 ** GO TO PASS
77 ** OTHERWISE
78 ** FOR N=1 TO NSERVE-1 DO
79 ** LET IFACTORIAL=IFACTORIAL*N
80 ** LET RATIO.FACTORIALS=RATIO.FACTORIALS*(MPOP-N+1)
81 ** LET P=V/R*IFACTORIAL.RATIO.FACTORIALS**OMITTING *P=NULL
82 ** LET P.NSTAIL(N)=P.N**WITH THE NTH STATE STORED IN THE NTH ELEMENT
83 ** ADD N*P.N TO ENO.DOWN
84 ** ADD (NSERVE-N)*P.N TO QSUM
85 ** ADD P.N TO SUM
86 ** ADD N*P.N TO E.BUSY.SERVERS
87 ** ADD N*2*P.N TO VAR.BUSY.SERVERS
88 **
89 ** LOOP ** TO DEVELOP SUMS
90 **
91 ** ** CALCULATE THE LAST TERM IN THE SUM FOR P.NULL
92 **
93 ** ** PASS ** LET CFACIOIAL=C*IFACTORIAL
94 ** LET CEXPOC=C**NSERVE
95 ** LET COEF=CEXPOC/CFACIOIAL
96 ** LET SUM=0.0 ** SUM FOR PROR THAT NO IN SYSTEM > OR = NSERVE
97 ** LET VMO.DOWN=VAR.BUSY.SERVERS
98 **
99 ** FOR N=NSERVE TO MPOP DO
100 ** LET RATIO.FACTORIALS=RATIO.FACTORIALS*(MPOP-N+1)
101 ** LET P.NECDEF=RATIO.FACTORIALS*RHO**N
102 ** ADD P.N TO SUM.LAST **OMITTING *P=NULL
103 ** ADD N*P.N TO ENO.DOWN **OMITTING *P=NULL
104 ** ADD N*2*P.N TO VAR.DOWN **OMITTING *P=NULL
105 ** LET P.NSTAIL(N)=P.N**OMITTING *P=NULL
106 ** ** TO OBTAIN THE LAST TERM
107 ** ADD SUM.LAST TO SUM
108 ** LET P.NULL=1.0/SUM ** PROB OF SYSTEM NULL STATE
109 **
110 ** ** CHECK OF CALCULATED MAINTENANCE SYSTEM STATE-PROBABILITIES
111 ** IF P.NULL GE 1.0 OR P.NULL LE 0.0
112 ** PRINT 1 LINE WITH P.NULL THUS
113 ** IN CALCULATING STATE PROBABILITIES. P.NULL = *****
114 ** STOP
115 ** OTHERWISE ** CALCULATE EXPECTED VALUS

```

```

114 LET ENO=DOWN=ENO.DOWN*P.NULL
115 LET VNO=DOWN=VNO.DOWN*P.NULL-ENO.DOWN**2
116 LET SCNO=DOWN=SCNO.DOWN*P.NULL-ENO.DOWN
117 LET P.SYS=FULL=SUM*P.NULL*SUM*LAST
118 LET E.BUSY=SERVERS=SUM*P.NULL*E.BUSY.SERVERS+C*P.SYS.FULL
119 LET VAR.BUSY=SERVERS=SUM*P.NULL*VAR.BUSY.SERVERS+C**2*P.SYS.FULL
120 LET BUSY.SERVERS**2
121 LET SD.BUSY=SERVERS=SUM*P.NULL*SD.BUSY.SERVERS
122 LET ENO.GUEUED=ENO.DOWN*NSERVE*P.NULL*QSUM
123 LET ESYS.WAIT=ENO.DOWN/LAMBDA/(REAL.F(MPOP)-ENO.DOWN)
124 **ABOVE EXPRESSION IS KNOWN AS LITTLE'S FORMULA.
125 LET EQ.WAIT=ESYS.WAIT-1.0/MU
126 **CALCULATE MAINTENANCE SYSTEM STATE PROBABILITIES AND THE PROBABILITY
127 **THAT THE NUMBER OF MACHINES DOWN EXCEEDS N.IEN.PCT.
128 **
129
130 LET P10=0.0
131 FOR N=1 TO MPOP DO
132 LET P.NSTATE(N)=P.NSTATE(N)*P.NULL
133 IF N GE N.IEN.PCT
134 ADD P.NSTATE(N) TO P10
135 ALWAYS
136 LOOP **OVER ALL STATES
137 **
138 **GO TO THE PRINT-OUTPUT SECTION IF THE MODE IN THIS ROUTINE IS NOT "OPTIMIZE".
139 **
140 IF MODE NE "OPTIMIZE"
141 GO TO PRNT
142 OTHERWISE **TEST FOR CONSTRAINTS
143 IF P10 > P1.LIM AND ENO.GUEUED/REAL.F(MPOP) > FQLIM
144 ADD 1 TO NSERVE
145 IF NSERVE > MPOP
146 SUBTRACT 1 FROM NSERVE
147 GO TO PRNT
148 OTHERWISE
149 GO TO S
150 OTHERWISE **PRINT OUTPUT
151 PRNT ** IF IPRINT=1
152 START NEW PAGE
153 ALWAYS
154 **CALCULATE THE NORMALIZATION CONSTANT FOR QV(**).
155 **
156 LET NORM.CONST=MPOP*P.NULL
157 FOR N=1 TO MPOP-1 DO
158 ADD (MPOP-N)*P.NSTATE(N) TO NORM.CONST
159 LOOP **TO CALCULATE THE NORMALIZATION CONSTANT
160 **
161 **CALCULATE THE ELEMENTS OF QV(**).
162 **
163 LET Q.NULL=MPOP*P.NULL/NORM.CONST
164 LET EQ.WAIT=QE=1.0
165 LET VQ.WAIT=QV=0.0
166 LET N=1 TO MPOP-1 DO
167 LET QV(N)=(MPOP-N)*P.NSTATE(N)/NORM.CONST
168 IF N GE NSERVE
169 LET JEN=NSERVE+1
170

```

```

171 ADD J*CV(N) TO EQ*WAIT*GO
172 ADD J*(J+1)*QV(N) TO VQ*WAIT*GO
173 ALWAYS
174 LOOP **TO GET QV(*)
175 LET G.SPACEL=1.0-Q.NULL
176 IF NSERVE=1
177 GO TO 11
178 OTHERWISE
179 FOR NEXT NSERVE-1 DO
180 SUBTRACT QV(N) FROM Q.SPACEL
181 **TO FIND Q.SPACEL
182 **11.11 P.MACH*WAIT*Q.SPACEL
183 LET P.SYS*QV(N)=1.0-Q.SPACEL
184 LET NMU=NSERVE*MU **THE MAX SERVICE RATE FOR A FULL SYSTEM
185 LET EQ*WAIT*CHECK=EQ*WAIT*GO/NMU **TO PROVIDE A CALCULATIONAL CHECK
186 LET VQ*WAIT*CHECK=VQ*WAIT*GO/NMU*EQ*WAIT*CHECK**2
187 LET SDG*WAIT*CHECK=SQRT.F(VQ*WAIT*CHECK)
188 **CALCULATE THE CONDITIONAL MEAN AND STANDARD DEVIATION OF THE
189 **WAITING TIME IN QUEUE, GIVEN A MACHINE MUST WAIT.
190 **
191 LET EQ*WAIT*GO=EQ*WAIT*GO/NMU*P.MACH*WAIT
192 LET VQ*WAIT*GO=VQ*WAIT*GO/NMU*P.MACH*WAIT*EQ*WAIT*GO**2
193 LET SDG*WAIT*GO=SQRT.F(VQ*WAIT*GO)
194 IF IPRINT NE 1
195 RETURN
196 OTHERWISE
197 PPRT 4 LINES THUS
198

```

A-28

SERVICE SYSTEM PERFORMANCE STATISTICS USING A FINITE-STATE QUEUEING MODEL

```

199-----PRINT 3 LINES WITH MORE, HPOP, NSERVE, HPOP*LAMBDA, MU-----
200 THUS
201 OUTPUT OF ROUTINE TO ***** MAINTENANCE FOR A POPULATION OF ***** MACHINES
202 WITH ***** SERVERS AND HAVING A MAX ARRIVAL RATE OF ***** MAINTENANCE ACTIONS
203 PER MINUTE AND A CONSTANT SERVICE RATE OF ***** ACTIONS PER MAN PER MINUTE.
204 SKIP 2 LINES
205 PRINT 15 LINES WITH P.SYS, FULL, G.NULL, P.MACH*WAIT, N.TEN*PCT, PIU*ENO*DOWN,
206 ENO*QUEUED, E.BUSY*SERVERS, SDG*WAIT*CHECK
207 EG*WAIT*GO, SDG*WAIT*GO, SDG*WAIT*CHECK
208 THUS
209 PROBABILITY THAT THE MAINTENANCE SYSTEM IS FULL *****
210 PROBABILITY THAT A FAILED MACHINE FINDS SYSTEM EMPTY *****
211 PROBABILITY THAT A MACHINE MUST QUEUE FOR SERVICE *****
212 PROBABILITY THAT THE NO MACHINES DOWN > OR = *** *****
213 EXPECTED NUMBER OF MACHINES DOWN *****
214 STD DEV NUMBER OF MACHINES DOWN *****
215 EXPECTED NUMBER OF BUSY SERVERS *****
216 STD DEV NUMBER OF BUSY SERVERS *****
217 EXPECTED DEVIATION OF BUSY SERVERS *****
218 STD DEVIATION (MIN) A MACHINE WAITS IN THE MAINT SYSTEM *****
219 EXPECTED TIME (MIN) A MACHINE WAITS IN THE MAINT QUEUE *****
220 EXPECTED QUEUE TIME (MIN) GIVEN A QUEUE *****
221 STD DEV QUEUE TIME (MIN) GIVEN A QUEUE *****
222 UNCONDITIONALLY *****

```

206 SKIP 2 LINES
207 PRINT 7 LINES THUS
MAINTENANCE SYSTEM STATE PROBABILITY DISTRIBUTIONS

```

-----
NO MACH      PROB      CUML      GIVEN ARRIVAL
IN  SYS      DENS      PROB      DENS      PROB
-----
208 LET CUM.DIST=0
209 LET CUM.DIST.COND=0.NULL
210 LET N=0
211 PRINT 1 LINE WITH N, P.NULL, CUM.DIST, 0.NULL, CUM.DIST.COND
212 THUS *****
213 FOR N=1 TO MPOP DO *****
214   ADD P.NSTATE(N) TO CUM.DIST
215   ADD Q(N) TO CUM.DIST.COND
216   PRINT 1 LINE WITH N, P.NSTATE(N), CUM.DIST, Q(N), CUM.DIST.COND
217   THUS *****
218 LOOP ***** OVER THE SYSTEM STATES *****
219 PRINT 2 LINES THUS *****

```

```

-----
220 LET DELT=MAX.F(1.0,TRUNC.F(EQ.WAIT.60/10.0))
221 LET MAX=60
222 SKIP 2 LINES
223 ** PRINT HEADINGS FOR QUEUE WAITING TIME DISTRIBUTION.
224 **
225 PRINT 7 LINES WITH 0.NULL THUS
226 CUMULATIVE AND CONDITIONAL CUM PROB DISTRIBUTIONS OF WAITING TIME IN QUEUE

```

```

-----
TIME      CUML      COND
(MIN)     PROD     PROB
-----
227 ** START TIME LOOP FOR WAITING TIME DISTRIBUTIONS.
228 **
229 FOR I=1 TO MAX ** TIME STEPS ** DO
230   LET TM=1+DULT
231   LET ARG=TM*MMU
232   LET CUM.DIST=1.0-QV(NSERVE)*EXP.F(-ARG)
233   LET SUMN=0.0
234   IF MPOP-NSERVE<2
235     GO TO BYN
236   OTHERWISE
237     FOR N=NSERVE+1 TO MPOP-1 DO
238       LET SUMJ=1.0
239       LET ARGJ=1.0
240       LET JFACTORIAL=1.0
241       FOR J=1 TO N-NSERVE DO
242         LET ARGJ=ARGJ+ARG
243         LET JFACTORIAL=JFACTORIAL*J
244

```


ANNEX B

HETEROGENEOUS MACHINE POPULATION

The theory presented in the body of this report treats the case of a homogeneous population of machines (customers), whose reliability and maintainability (RAM) parameters are the same for all machines. Thus, equations (1), (2), and (3) do not distinguish between categories or types of machines having possibly different, intrinsic mean times between failure (MTBF) and mean times to repair (MTTR). For this ideal case, the approach presented in the body of the report is exact.

In many instances the population of machines is heterogeneous. In this case an approximation is needed for the RAM parameters which are equivalent to their homogeneous counterparts. This annex discusses an approximation which can be used in conjunction with the previous analytical results to yield queueing statistics which are in good agreement with simulated results for tandem production operations.

On a production line different machine operations may each have several machines of a common type. However, each type of machine (or operation) may have RAM parameters different from parameters of other types. Furthermore, in this setting the machines in a particular operation (n) do not operate continuously. Their failure behavior will reflect this condition. Assuming that the machine operations are asynchronous, a machine must wait: (a) if the downstream buffer is full, (b) if the upstream buffer is empty, or (c) if the machine is down for repairs. The definition of arrival rate per machine (λ) requires the use of the reciprocal of the actual--as opposed to intrinsic--mean clock time between failures. Call this the effective mean time between failures MTBF*. Similarly, the effective mean time to repair (MTTR*) must reflect the diversity of types of machines repaired, the MTTR for each type, and the relative demand for repair by type. Then, the repair parameter (μ_1) used in homogeneous queueing theory is the reciprocal of MTTR*.

The following method is a partial (and imperfect) attempt to account for the non-operating time, in addition to maintenance downtime, which the machines of a particular operation experience. Generally, the production rates of acceptable parts from the machine operations are not identical, i.e., the line

is unbalanced. The machines of an operation which does not have minimal thruput must wait a portion of the time. I define the operational ratio of operation n as the fraction of the average available machine-time which machines at this operation must work in order to meet line thruput. Reference is made to Table B-1 for a complete list of symbols and terms used in this discussion. The average number of machine-minutes actually worked during a day by machines at operation n is, then,

$$U(n)D A(n)\rho(n) , \quad (B-1)$$

where the symbols U , D , A , and ρ are defined in Table B-1. The expected number of maintenance actions required at this operation is this total operating time divided by the $MTBF(n)$. This number is denoted $v(n)$. The total of average daily maintenance actions required is denoted by X .

Formally,

$$X = \sum_{n=1}^N v(n) . \quad (B-2)$$

The total daily average maintenance time required is the following sum over all operations:

$$T = \sum_{n=1}^N v(n)MTTR(n) . \quad (B-3)$$

The effective mean time, per machine, between failures is the maximum daily machine-minutes ($D M$) divided by X :

$$MTBF^* = D M/X . \quad (B-4)$$

The effective mean time to repair is

$$MTTR^* = T/X . \quad (B-5)$$

The equivalent arrival rate of maintenance actions-- $1/MTBF^*$ --is the sum of required actions per unit time over all machine types divided by the total number of machines. The equivalent mean time to repair, or reciprocal equivalent service rate, is a weighted sum of $MTTR$'s for each machine type. The weights in this expression represent the fraction of the total maintenance actions contributed by machines of this (n) type.

At this point the reader may have noticed that one of the outputs of the queueing model (W_q) is needed to define the equivalent values of λ and μ . Availability (A) of machines of a given type in a manufacturing operation is used to estimate the thruput of that operation. This, in turn, is used to develop parameters λ and μ of the queueing model. However, A must consider

the expected waiting time in the maintenance queue (W_q) as well as the intrinsic RAM characteristics of that type of machine. Since A involves W_q , which is not yet known, an iterative solution technique is required. This process starts by assuming that W_q is zero, calculates A for each machine type, which leads to estimates of λ and μ . Use of λ and μ in the queueing model, then, yields an improved estimate of W_q . The better estimate of W_q permits better estimates of λ and μ , and so forth. To accelerate the conversion of this iterative process, it is helpful to use a value of W_q (the same for all machines) which is the average of the i th and $(i-1)$ th iterates. Convergence in some instances is not uniform. Computational experience indicated that truncation of this iterative process occurs rapidly using the following convergence criterion: the relative difference in values of W_q for the last two iterations must be less than 2%. Sound computer coding practice provides a sure escape from the iteration loop after a maximum number of iterations has occurred. In the TANDEMT program, the maximum was set to 20, altho iterative truncation has yet to occur.

TABLE B-1

NOTATION USED IN OBTAINING
EFFECTIVE ARRIVAL AND SERVICE RATES

Symbol	Description
D	length of workday (minutes)
U(n)	the number of (units of) machines of a common type in operation n
N	the number of machine types or machine operations
M	the total number of machine units , $= \sum_n^N U(n)$
MTBF(n)	the mean operating time between failures (minutes) of machines of type n
MTTR(n)	the mean time (minutes) to repair a machine of type n, given maintenance resources
WQ	the mean time (minutes) a machine spends in the repair queue
A(n)	the steady-state availability of machines of type n, $= \text{MTBF}(n) / [\text{MTBF}(n) + \text{MTTR}(n) + \text{WQ}]$
R(n)	unit machine rate (per min) at operation n
AP(n)	average daily production capacity at operation n $= U(n)D R(n)A(n)$
$\alpha(n)$	acceptance rate of parts from machines at operation n only
C(n)	cumulative acceptance rate of parts passing operation n per part starting the first operation $= \prod_{j=1}^n \alpha(j)$
CP(n)	average daily production capacity of <u>acceptable</u> parts at operation n, $= AP(n) \alpha(n)$
n*	value of n for which CP(n) is a minimum $CP(n^*) \leq CP(n) \quad , \quad 1 \leq n \leq N$
RP(n)	average daily parts production at operation n required to meet the average daily line thruput, $= AP(n^*) C(n)/C(n^*)$

TABLE B-1 (Cont'd)

NOTATION USED IN OBTAINING
EFFECTIVE ARRIVAL AND SERVICE RATES

Symbol	Description
$\rho(n)$	operational ratio for machine operation n , $= RP(n)/AP(n)$
$v(n)$	average daily number of maintenance actions for machines of common type in operation n , $= U(n)D A(n)\rho(n)/MTBF(n)$
X	total average daily maintenance actions required, $= \sum_n^N v(n)$
T	total average daily maintenance time (min) required, $= \sum_n^N v(n)MTTR(n)$
$MTBF^*$	effective mean time (min) between failures, $= D M/X$
$MTTR^*$	effective mean time (min) to repair all machines, $= T/X$
λ^*	effective arrival rate (1/minute), $= 1/MTBF^*$
μ_1^*	effective service rate (1/minute), $= 1/MTTR^*$