The Single Server Queue in Discrete Time-
Numerical Analysis II

by

Eugene M. Klimko and Marcel F. Neuts(*)

Purdue University

Department of Statistics

Division of Mathematical Sciences

Mimeograph Series No. 276

February 1972.

(*) The research of this author was supported by the National Science Foundation grant GP 28650.

Abstract

This paper deals with the numerical problems arising in the computation of higher order moments of the busy period for certain classical queues of the M|G|1 type, both in discrete and in continuous time.

The classical functional equation for the moment generating function of the busy period is used. The higher order derivatives at zero of the moment generating function are computed by repeated use of the classical differentiation formula of Fáa di Bruno. Moments of order up to fifty may be computed in this manner.

A variety of computational aspects of Fáa di Bruno's formula, which may be of use in other areas of application, are also discussed in detail.

1. Introduction

This paper deals with the numerical computation of moments of high order of the busy period for several fundamental queueing models. We shall specifically consider the following queueing processes.

The Models

For brevity we refer to the two models as (a) the continuous model and (b) the discrete model respectively.

(a) The continuous model.

This is the classical $M^{(X)}|G|1$ queue in which customers arrive in groups of random size in epochs which form a homogeneous Poisson process of rate λ. They are served, one at the time, and their successive service times are independent and identically distributed random variables with the common distribution $H(\cdot)$. We further assume that the successive group sizes are independent, identically distributed random variables. The probability that an arrival consists of a group of exactly k customers is denoted by p_k, where $p_0=0$ and $\sum_{k=1}^{\infty} p_k=1$.

The n-th moments of the distribution $H(\cdot)$ and of the discrete density $\{p_k\}$ are denoted by α_n and η_n respectively. We assume throughout this paper that all moments considered are finite.

The busy period of this queue is defined as the length of time until a server, starting with a single customer, becomes idle for the first time. For every $x \geq 0$, $G(x)$ denotes the probability that the duration of the busy period does not exceed x. The function $G(\cdot)$ is then a (possibly defective) probability distribution. The following theorems are well-known, but are stated here for completeness.

Theorem 1

For every $x \geq 0$, the function $G(\cdot)$ satisfies the nonlinear integral equation

$$(1) \qquad G(x)=\sum_{\nu=0}^{\infty}\int_0^x e^{-\lambda y}\frac{(\lambda y)^\nu}{\nu!}\sum_{r=\nu}^{\infty} p_r^{(\nu)}G^{(r)}(x-y)dH(y),$$

in which $G^{(r)}(\cdot)$ is the r-fold convolution of $G(\cdot)$ with itself and $\{p_r^{(\nu)}\}$ is the ν-fold convolution of the discrete density $\{p_r\}$ with itself.

Proof

By an application of the law of total probability. Let there be exactly ν arrivals during the service of the first customer and let there be exactly $r \geq \nu$ customers in the ν arriving groups. Each of these r customers can be considered as the initial one of r independent busy periods. If the first service has a duration y, then the r busy periods so generated can last, at most for a length of time x-y. This argument is due to L. Takàcs and may be found in more detail in [7], p. 47.

Introducing the moment generating functions $\gamma(s)$, $h(s)$ and $\theta(s)$ of $G(\cdot)$, $H(\cdot)$ and $\{p_k\}$, i.e.

$$(2) \qquad \gamma(s)=\int_0^\infty e^{sx}dG(x), \qquad h(s)=\int_0^\infty e^{sx}dH(x),$$

$$\theta(s)=\sum_{k=1}^{\infty} p_k e^{ks},$$

for Re $s \leq 0$, the integral Equation (1) may be equivalently written as

$$(3) \qquad \gamma(s)=h[s-\lambda+\lambda\theta[\log \gamma(s)]],$$

where log denotes the principal branch of the logarithmic function.

Theorem 2

For every s with Re $s < 0$, the functional Equation (3) has a unique solution in the unit disk. This solution $\gamma(s)$ is analytic in the half-plane Re $s < 0$ and continuous on the boundary. Moreover $\gamma(\cdot)$ is the moment generating function of a (possibly defective) probability distribution $G(\cdot)$.

Proof

This result, which is usually proved by applying Rouché's theorem, is a classical theorem in the theory of queues.

Theorem 3

The probability distribution $G(\cdot)$ is proper, i.e. $G(+\infty)=1$, if and only if

$$(4) \qquad \lambda\eta_1\alpha_1 \leq 1.$$

If equality holds, the first moment of the busy period is infinite.

Proof

See e.g. Takàcs [6].

Remark

Throughout this paper we consider only stable queues, i.e. we assume henceforth that

$$(5) \qquad \lambda\eta_1\alpha_1 < 1.$$

(b) The discrete model

As discussed by S. Dafermos and M. F. Neuts [1], there is a considerable advantage, particularly from a computational viewpoint, in considering the discrete-time analogue of the model (a). The time variable is now discrete. The numbers of customers arriving during successive unit time intervals are independent, identically distributed integer-valued random variables with discrete density $\{p_k'\}$. p_0' is now (usually) positive and is the probability that no customer arrives during a unit of time.

Customers are served singly and the probability that a customer requires ν units of service time is denoted by r_ν for $\nu \geq 1$. The usual independence assumptions are made.

The busy period is again the length of time until a server, starting with one customer, becomes idle for the first time. We shall denote by β_n, $n \geq 1$, the probability that the busy period lasts for exactly n units of time. $\{\beta_n\}$ is a (possibly defective) discrete probability density.

Theorem 4

The density $\{\beta_n\}$ satisfies the nonlinear recurrence relation

$$(6) \qquad \beta_n = \sum_{\nu=1}^{n} r_\nu \sum_{k=0}^{\infty} p_k'^{(\nu)} \beta_{n-\nu}^{(k)},$$

for $n \geq 1$. In Equation (6), $\{p_k'^{(\nu)}\}$ and $\{\beta_k^{(\nu)}\}$ are the ν-fold convolutions of the densities $\{p_k'\}$ and $\{\beta_k\}$.

Proof

By an application of the law of total probability as in the proof of theorem 1, (6) follows.

We now introduce the moment generating functions

$$(7) \qquad B(s) = \sum_{n=1}^{\infty} \beta_n e^{ns}, \; R(s) = \sum_{n=1}^{\infty} r_n e^{ns}, \; P(s) = \sum_{n=0}^{\infty} p'_n e^{ns},$$

for Re $s \leq 0$ and obtain by using Equation (6) that

$$(8) \qquad B(s) = \sum_{n=1}^{\infty} \beta_n e^{ns} = \sum_{\nu=1}^{\infty} \sum_{n=\nu}^{\infty} r_\nu e^{ns} \sum_{k=0}^{\infty} p_k'^{(\nu)} \beta_{n-\nu}^{(k)}$$

$$= \sum_{\nu=1}^{\infty} r_\nu e^{\nu s} \sum_{\tau=0}^{\infty} e^{\tau s} \sum_{k=0}^{\infty} p_k'^{(\nu)} \beta_\tau^{(k)}$$

$$= \sum_{\nu=1}^{\infty} r_\nu e^{\nu s} \sum_{k=0}^{\infty} p_k'^{(\nu)} B^k(s) = \sum_{\nu=1}^{\infty} r_\nu e^{\nu s} P^\nu[\log B(s)]$$

$$= R[s + \log P[\log B(s)]],$$

for Re $s < 0$. In both cases log is the principal branch of the logarithmic function.

Theorem 5

The functional Equation (8) has a unique solution inside the unit disk for every s with Re s < 0. The solution B(s) is analytic inside the region Re s < 0 and continuous on the boundary and the function B(.) is the moment generating function of a discrete density $\{\beta_n\}$. The density $\{\beta_n\}$ is proper, i.e. $\sum_{n=1}^{\infty} \beta_n=1$, if and only if

$$\eta_1 \alpha_1 \leq 1 \tag{9}$$

where $\alpha_1=R'(0-)$ is the mean service time and $\eta_1=P'(0-)$ is the mean number of arrivals per unit of time. When equality holds in (9), the mean of the busy period is infinite.

Proof

This theorem is proved in its generating function version in S. Dafermos and M. F. Neuts, [1].

Remark

In the remainder of this paper we shall again restrict our attention to the case of stable queues and we assume therefore that $\eta_1\alpha_1 < 1$.

2. Statement and Significance of the problem.

The moments of the busy period may be obtained by successive differention with respect to s in the Equation (3) for the continuous case and in the Equation (8) for the discrete case. By taking the limit as $s \to 0-$ and by appealing to Abel's theorem, we obtain a relation expressing the n-th moment in terms of the moments of orders one up to n-1. However in view of the multiple functional compositions which occur on the right hand sides of the Equations (3) and (8), this recurrence relation soon becomes very unwieldy and its value for numerical calculations is far from obvious.

The purpose of our discussion is to show that, on the contrary, the recurrence relations generated by successive differentiations are practical for the evaluation of moments up to order fifty approximately. We substantiate this claim by exhibiting the results of a number of actual computations.

Knowledge of the higher moments of the busy period is useful, in particular for the following reason. The equation (1) and (6) are both of the general form

$$G = \sum_{n=0}^{\infty} A_n * G^{(n)}. \tag{10}$$

The former is now a nonlinear integral equation and the latter a nonlinear difference equation. Both can be shown to be well-suited for numerical solution by successive substitution methods. We shall report in a subsequent paper on procedures for selecting a good starting solution and on the computer implementation of the method of iterative solution.

However, a very important step in this algorithm is the selection of a practical upper bound for the support of the distribution $G(\cdot)$ or of the density $\{\beta_n\}$. Explicitly, one needs to find a quantity A such that for a given error term ε, (say 10^{-4}), either

$$1-G(A) \leq \varepsilon, \quad \text{or} \quad \sum_{\nu=A+1}^{\infty} \beta_\nu \leq \varepsilon. \tag{11}$$

One such procedure for finding A is by application of Markov's inequality which is easily implemented but rather crude.

3. The Recurrence Relation.

The functional Equations (3) and (8) are very similar in nature. We shall give a detailed discussion of Equation (3) and indicate the appropriate changes for application of our results to the Equation (8). The n-th moment of the busy period will be denoted by g_n. Clearly

$$(12) \qquad g_n = \gamma^{(n)}(0-), \quad \text{for} \quad n \geq 1.$$

This leads for n=1 and n=2 to the well-known formulas

$$(13) \qquad g_1 = \alpha_1(1-\lambda\eta_1\alpha_1)^{-1},$$

$$g_2 = [\alpha_2 + \lambda(\eta_2-\eta_1)\alpha_1^3](1-\lambda\eta_1\alpha_1)^{-3}.$$

In order to express the n-th derivative we appeal to the classical formula of Fáa di Bruno [21], which expresses the n-th derivative of a composite function $f[\varphi(x)]$ as follows.

Fáa di Bruno's Formula

Assuming the existence of all the derivatives involved, we have that

$$(14) \qquad \left\{\frac{d^n}{dx^n} f[\varphi(x)]\right\}_{x=0} =$$

$$\sum_{r=1}^{n} \left\{\frac{d^r}{dy^r} f(y)\right\}_{y=\varphi(0)} \sum_{\substack{j_1+\ldots+j_n = r \\ j_1+2j_2+\ldots+nj_n = n \\ j_1 \geq 0,\ldots,j_n \geq 0}} \frac{n!}{j_1!j_2!\ldots j_n!} \times$$

$$\left(\frac{\varphi^{(1)}(0)}{1!}\right)^{j_1}\left(\frac{\varphi^{(2)}(0)}{2!}\right)^{j_2}\cdots\left(\frac{\varphi^{(n)}(0)}{n!}\right)^{j_n}.$$

In general, assuming the existence of all the derivatives involved, the n-th moment satisfies

$$g_n = \left\{\frac{d^n}{ds^n} h[s-\lambda+\lambda\theta[\log \gamma(s)]]\right\}_{s=0-} = \sum_{r=1}^{n} \alpha_r Y_{nr}^{(1)} \tag{15}$$

where

$$Y_{nr}^{(1)} = \sum_{\substack{j_1+\dots+j_n = r \\ j_1+2j_2+\dots+nj_n = n \\ j_1 \geq 0,\dots,j_n \geq 0}} \frac{n!}{j_1!\dots j_n!} A_1^{j_1} A_2^{j_2} \dots A_n^{j_n}. \tag{16}$$

The quantities $A_1,\dots,A_n$ are given by

$$A_1 = 1+\lambda\theta'(0)\gamma'(0) = 1 + \lambda\eta_1 g_1 = (1-\lambda\eta_1\alpha_1)^{-1}, \tag{17}$$

and for $2 \leq \nu \leq n$

$$A_\nu = \frac{\lambda}{\nu!}\left\{\frac{d^\nu}{dx^\nu}\theta[\log \gamma(x)]\right\}_{x=0}.$$

The latter derivatives are given by

$$\left\{\frac{d^\nu}{dx^\nu}[\log \gamma(x)]\right\}_{x=0} = \sum_{m=1}^{\nu} \eta_m Y_{\nu m}^{(2)}, \tag{18}$$

for $2 \le \nu \le n$, where

(19) $$Y^{(2)}_{\nu m} = \sum_{\substack{i_1+i_2+\dots+i_\nu=m \\ i_1+2i_2+\dots+\nu i_\nu=\nu \\ i_1 \ge 0,\dots,i_\nu \ge 0}} \frac{\nu!}{i_1!\,i_2!\dots i_\nu!} B_1^{i_1} B_2^{i_2} \dots B_\nu^{i_\nu}.$$

The quantities B_ρ are given by:

(20) $$B_1 = \gamma'(0) = g_1 = \alpha_1(1-\lambda\eta_1\alpha_1)^{-1},$$

and for $2 \le \rho \le \nu \le n$

$$B_\rho = \frac{1}{\rho!} \left\{ \frac{d^\rho}{dx^\rho} \log \gamma(x) \right\}_{x=0}.$$

The latter derivatives, in turn, are given by:

(21) $$\left\{ \frac{d^\rho}{dx^\rho} \log \gamma(x) \right\}_{x=0} = \sum_{h=1}^{\rho} (-1)^{h+1} (h-1)!\, Y^{(3)}_{\rho h}$$

where

(22) $$Y^{(3)}_{\rho h} = \sum_{\substack{\tau_1+\dots+\tau_\rho=h \\ \tau_1+2\tau_2+\dots+\rho\tau_\rho=\rho \\ \tau_1 \ge 0,\dots,\tau_\rho \ge 0}} \frac{\rho!}{\tau_1!\,\tau_2!\dots\tau_\rho!} \left(\frac{g_1}{1!}\right)^{\tau_1} \left(\frac{g_2}{2!}\right)^{\tau_2} \dots \left(\frac{g_\rho}{\rho!}\right)^{\tau_\rho}.$$

The formulas (15)-(22) may be combined to express g_n as a (complicated) polynomial in $g_1,\ldots,g_{n-1},g_n$. A notable simplication occurs however if we observe that g_n occurs only once among the terms on the right. By examining the successive conditions on the indices in the three applications of Fáa di Bruno's formula, we find that the only term containing g_n which appears on the right hand side is

$$(\lambda\eta_1\alpha_1)g_n.$$

It follows that

$$(23) \qquad (1-\lambda\eta_1\alpha_1)g_n = \sum_{r=2}^{n} Y_{nr}^{(1)}\alpha_r + \lambda\alpha_1\sum_{m=2}^{n} Y_{nm}^{(2)}\eta_m + \lambda\alpha_1\eta_1\sum_{h=2}^{n}(-1)^{h+1}(h-1)!\,Y_{nh}^{(3)}.$$

The expression on the right is a polynomial of degree n in $g_1,g_2,\ldots g_{n-1}$. By application of the formulas (16)-(23) we compute the higher moments of the busy period recursively. It is clear that the practical limitations on this method depend mainly on the growth of the number of terms appearing on the right hand side in Fáa di Bruno's formula. This matter is discussed below, but first we indicate the modifications necessary for the discrete case and for some particular cases.

The case of single arrivals

When customers arrive singly, we have $\theta(s)=e^s$, so that the Equation (3) reduces to

(24) $\gamma(s)=h[s-\lambda+\lambda\gamma(s)].$

This case requires only <u>two</u> applications of Fáa di Bruno's formula.

<u>The discrete case</u>

In general, the discrete case requires four applications of Fáa di Bruno's formula. The basic formulas in this case are

(25) $$g_1=\alpha_1(1-\lambda_1\alpha_1)^{-1},$$

$$g_2=[\alpha_2+\alpha_1(\eta_2-\eta_1^2-\eta_1)](1-\eta_1\alpha_1)^{-3},$$

and

(26) $$g_n=\sum_{r=1}^{n}\alpha_r Z_{nr}^{(1)}, \qquad \text{for } n \geq 3,$$

where

(27) $$Z_{nr}^{(1)}=\sum_{\substack{j_1+\ldots+j_n=r \\ j_1+2j_2+\ldots+nj_n=n \\ j_1\geq 0,\ldots,j_n\geq 0}} \frac{n!}{j_1!\ldots j_n!} A_1^{j_1}\ldots A_n^{j_n},$$

for $1 \leq r \leq n$.

(28) $$A_1 = (1-\eta_1\alpha_1)^{-1}$$

(29) $$A_\nu = \frac{1}{\nu!} \left\{ \frac{d^\nu}{dx^\nu} \log P[\log B(x)] \right\}_{x=0} ,$$

for $2 \leq \nu \leq n$.

Furthermore

(30) $$\left\{ \frac{d^\nu}{dx^\nu} \log P[\log B(x)] \right\}_{x=0} = \sum_{m=1}^{\nu} (-1)^{m+1} (m-1)!\, Z_{\nu m}^{(2)},$$

where

(31) $$Z_{\nu m}^{(2)} = \sum_{\substack{j_1+\ldots+j_\nu = m \\ j_1+2j_2+\ldots+\nu j_\nu = \nu \\ j_1 \geq 0,\ldots,j_\nu \geq 0}} \frac{\nu!}{j_1!\ldots j_\nu!} C_1^{j_1}\ldots C_\nu^{j_\nu},$$

where

(32) $$C_1 = \eta_1 g_1 = \alpha_1 \eta_1 (1-\alpha_1\eta_1)^{-1},$$

and

(33) $$C_\rho = \frac{1}{\rho!} \left\{ \frac{d^\rho}{dx^\rho} P[\log B(x)] \right\}_{x=0} , \quad \text{for } 2 \leq \rho \leq \nu \leq n.$$

The latter derivative is given by

(34) $$\left\{ \frac{d^\rho}{dx^\rho} P[\log B(x)] \right\}_{x=0} = \sum_{\tau=1}^{\rho} \eta_\tau Z_{\rho\tau}^{(3)}$$

where

$$(35) \qquad Z^{(3)}_{\rho\tau} = \sum_{\substack{j_1 + \cdots + j_\rho = \tau \\ j_1 + 2j_2 + \cdots + \rho j_\rho = \rho \\ j_1 \geq 0, \ldots, j_\rho \geq 0}} \frac{\rho!}{j_1! \cdots j_\rho!} E_1^{j_1} \cdots E_\rho^{j_\rho}.$$

The quantities E_q, for $1 \leq q \leq \tau \leq \nu \leq n$ are given by

$$(36) \qquad E_q = \frac{1}{q!} \left\{ \frac{d^q}{dx^q} \log B(x) \right\}_{x=0} ,$$

where

$$(37) \qquad \left\{ \frac{d^q}{dx^q} \log B(x) \right\}_{x=0} = \sum_{u=1}^{q} (-1)^{u+1} (u-1)! \, Z^{(4)}_{qu} ,$$

and

$$(38) \qquad Z^{(4)}_{qu} = \sum_{\substack{j_1 + \cdots + j_q = u \\ j_1 + 2j_2 + \cdots + qj_q = q \\ j_1 \geq 0, \ldots, j_q \geq 0}} \frac{u!}{j_1! \cdots j_q!} \left(\frac{g_1}{1!} \right)^{j_1} \cdots \left(\frac{g_q}{q!} \right)^{j_q} ,$$

for $u=1,\ldots,q$.

A similar observation as in the continuous case can be made here. By examination of the conditions on the indices of the Fáa di Bruno formulas, we note that the only term involving g_n which appears on the right hand side is $\alpha_1 \eta_1 g_n$. It follows that the quantity $(1-\alpha_1 \eta_1)g_n$ may be expressed as a polynomial in $g_1,\ldots,g_{n-1}$.

Remark

It is often convenient to choose the mean service time α_1 as a new unit of time. The normalized n-th moment $\tilde{g}_n$ corresponding to $\alpha_1 = 1$ is related to the one corresponding to a general value of α_1 by the formula

(39) $$g_n = \tilde{g}_n \alpha_1^n.$$

4. The Moments for the Busy Period of the M|M|1 Queue.

Explicit expressions for the higher moments of the busy period are available in rare cases only. As is often the case, the M|M|1 queue leads to tractable expressions. The moments of the M|M|1 queue were used to verify the accuracy and the correctness of our general computer routines.

In this section, we study the moments $\tilde{g}_n$ of the busy period for the M|M|1 queue, first with single arrivals and next for the case of geometrically distributed bunch sizes.

a. Single Arrivals

The moment generating function for the busy period of the queue with Poisson arrivals of rate ρ and with mean service time $\mu = 1$ is given by

(40) $$\gamma(s) = \frac{1+\rho-s}{2\rho} - \frac{1-\rho}{2\rho}\left[1-s(1-\sqrt{\rho})^{-2}\right]^{\frac{1}{2}}\left[1-s(1+\sqrt{\rho})^{-2}\right]^{\frac{1}{2}}.$$

After a routine series expansion, using the binomial series, one obtains

$$(41) \qquad \gamma(s) = 1+s(1-\rho)^{-1} + \frac{1-\rho}{2\rho} \sum_{n=2}^{\infty} (-1)^{n+1} s^n \sum_{\nu=0}^{n} \binom{\frac{1}{2}}{\nu}\binom{\frac{1}{2}}{n-\nu} (1-\sqrt{\rho})^{-2\nu}(1+\sqrt{\rho})^{2\nu-}$$

The moments $\tilde{g}_n$ of the busy period are therefore given by

$$(42) \qquad \tilde{g}_1 = (1-\rho)^{-1},$$

and

$$(43) \qquad \tilde{g}_n = n! \, \frac{(1-\rho)}{2\rho} (-1)^{n+1} \sum_{\nu=0}^{n} \binom{\frac{1}{2}}{\nu}\binom{\frac{1}{2}}{n-\nu} (1-\sqrt{\rho})^{-2\nu}(1+\sqrt{\rho})^{2\nu-2n},$$

for $n \geq 2$.

Simplified expressions for $2 \leq n \leq 5$ are

$$(44) \qquad \begin{aligned} \tilde{g}_2 &= 2(1-\rho)^{-3}, \\ \tilde{g}_3 &= 3! \ (1-\rho)^{-5}(1+\rho), \\ \tilde{g}_4 &= 4! \ (1-\rho)^{-7}(1+3\rho+\rho^2), \\ \tilde{g}_5 &= 5! \ (1-\rho)^{-9}(1+\rho)(1+5\rho+\rho^2). \end{aligned}$$

In order to evaluate $\tilde{g}_n$ numerically we wrote $\tilde{g}_n$ in the form

$$(45) \qquad \tilde{g}_n = A'_{n0} - \sum_{\nu=1}^{n-1} A'_{n\nu} + A'_{nn},$$

where

$$(46) \qquad A'_{n0} = n!\,\frac{1-\rho}{2\rho}\,(-1)^{n+1}\binom{\frac{1}{2}}{n}\,(1+\sqrt{\rho})^{-2n} ,$$

$$A'_{nn} = n!\,\frac{1-\rho}{2\rho}\,(-1)^{n+1}\binom{\frac{1}{2}}{n}\,(1-\sqrt{\rho})^{-2n} ,$$

and

$$A'_{n\nu} = n!\,\frac{1-\rho}{2\rho}\,(-1)^{n}\binom{\frac{1}{2}}{\nu}\binom{\frac{1}{2}}{n-\nu}\,(1-\sqrt{\rho})^{-2}\,(1+\sqrt{\rho})^{2\nu-2n} ,$$

for $\nu = 1,\ldots,n-1$.

We further noted the following recurrence relations

$$(47) \qquad A'_{n0} = A'_{n-1,0}(n-3/2)(1+\sqrt{\rho})^{-2} ,$$

$$A'_{n,n} = A'_{n-1,n-1}(n-3/2)(1-\sqrt{\rho})^{-2} ,$$

$$A'_{n,\nu} = A'_{n-1,\nu}n[1-\frac{3}{2}\frac{1}{n-\nu}](1+\sqrt{\rho})^{-2} , \qquad \text{for } \nu = 1,\ldots,n-2$$

$$A'_{n,n-1} = A'_{n-1,n-1}\,\frac{n}{2}\,(1+\sqrt{\rho})^{-2} .$$

For each n, the quantities $A'_{n,\nu}$, $0 \leq \nu \leq n$ are readily computed in terms of the corresponding quantities for n-1. The accuracy of the single precision calculation was checked by comparison with a double precision routine and, for the values computed, agreement was found for the first eleven significant digits. Moments up to order eighty were computed in this manner.

b. Group Arrivals

It was further desirable to have explicit expressions for a queue with group arrivals.

Explicit expressions for the moments $\tilde{g}_n^*$ for the queue with exponential service times and geometric bunch size distribution were obtained. For this queue, the moment generating function $\gamma(s)$ satisfies the quadratic equation

$$(\lambda+q-qs)\gamma^2(s) - (\lambda+q+1-s)\gamma(s)+1 = 0, \tag{48}$$

where p is the parameter of the geometric bunch size distribution, $q = 1-p$ and λ is the arrival rate of the groups of customers. The solution in the unit disk of this equation is given by

$$\gamma(s) = \frac{(\lambda+q+1) - s}{2(\lambda+q)} \left(1- \frac{qs}{\lambda+q}\right)^{-1} + \frac{\lambda-p}{2(\lambda+q)} \left(1- \frac{qs}{\lambda+q}\right)^{-1} \left[1 - \frac{s}{(\sqrt{\lambda} + \sqrt{p})^2}\right]^{\frac{1}{2}} \left[1- \frac{s}{(\sqrt{\lambda} - \sqrt{p})^2}\right]^{\frac{1}{2}} . \tag{49}$$

Using the negative binomial and the geometric series and performing several multiplications of series, we found the expansion

$$\gamma(s) = \frac{\lambda+q+1}{2(\lambda+q)} \left\{1 + \frac{q^2-\lambda p}{q(\lambda+q+1)} \sum_{r=1}^{\infty} \left(\frac{q}{\lambda+q}\right)^r s^r\right\} + \sum_{r=0}^{\infty} s^r \sum_{k=0}^{r} \left(\frac{q}{\lambda+r}\right)^{r-k} (-1)^k \sum_{\nu=0}^{k} \binom{\frac{1}{2}}{\nu}\binom{\frac{1}{2}}{k-\nu} (\sqrt{\lambda}+\sqrt{p})^{-2\nu} (\sqrt{\lambda}-\sqrt{p})^{2\nu-2k} . \tag{50}$$

It is known that this queue is stable if and only if $\lambda \leq p$. If we denote $\frac{\lambda}{p}$ by θ, then the mean of the busy period is given by

(51) $$\tilde{g}_1^* = (1-\theta)^{-1} .$$

The higher moments $\tilde{g}_n^*$ of the busy period are simply related to the moments $\tilde{g}_k(\theta)$, $k \geq 1$, of the M|M|1 queue with single arrivals and traffic intensity θ. Upon examination of the coefficient of s^n in the series expansion (50) and by using formula (43), we obtain

(52) $$\tilde{g}_n^* = n! \left(\frac{q}{\lambda+q}\right)^n + \frac{\lambda}{\lambda+q} n! \sum_{k=1}^{n} \left(\frac{q}{\lambda+q}\right)^{n-k} \frac{1}{p^k k!} \tilde{g}_k(\theta)$$

for $n \geq 2$. Formula (52) indicates clearly how the quantities $\tilde{g}_n^*$ can be computed by a routine modification of the algorithm used for the evaluation of the moments $\tilde{g}_n(\theta)$.

5. Numerical Aspects of Faà di Bruno's Formula

a. The number of terms

In order to study the number of terms appearing on the right hand side in Faà di Bruno's formula, let $\Psi(n,r,s)$ be the number of n-tuples $(j_1,\ldots,j_n)$ of nonnegative integers satisfying

(53) $$j_1+j_2+\ldots+j_n = r, \quad j_1+2j_2+\ldots+nj_n = s, \quad \text{for } s \geq r.$$

The quantities $\Psi(n,r,s)$ satisfy the recurrence relations

$$\Psi(n,r,s) = \Psi(n-1,r,s) + \sum_{j=1}^{j^*} \Psi(n-1,r-j,s-nj), \tag{54}$$

where:

$$j^* = \min\left(r, \frac{s-r}{n-1}\right), \quad \text{for} \quad n \geq 2. \tag{55}$$

If $j^* = 0$, then the latter summation is vacuous. The recurrence is initialized by $\Psi(1,1,1) = 1$.

The number of terms in Fáa di Bruno's formula is then given by

$$\sum_{r=1}^{n} \Psi(n,r,n) = \Psi^*(n). \tag{56}$$

Using (54), we computed $\Psi^*(n)$ for n up to 100. The growth of $\Psi^*(n)$ is quite slow up to n = 30, is moderate between n = 31 and n = 40 and is fast from n = 40 on. The terms slow, moderate and fast are, of course, strictly qualitative. We note the following values of $\Psi^*(n)$.

(57)
$$\Psi^*(10) = 42,$$
$$\Psi^*(20) = 627,$$
$$\Psi^*(30) = 5{,}604,$$
$$\Psi^*(40) = 37{,}338,$$
$$\Psi^*(50) = 204{,}226.$$

Again in qualitative terms, and in relation to the speed of present-day computers, we observe that Fáa di Bruno's formula is readily applicable for $n \leq 30$ whereas n = 50 is very much a practical upper bound for its computational use.

b. The number of nonzero indices

Computationally it is advantageous to use only the nonzero indices in Fáa di Bruno's formula. We therefore investigated the maximum number $\chi(n)$ of positive integers j_ν occurring in an n-tuple $(j_1,\ldots,j_n)$ of nonnegative integers, satisfying

(58) $$j_1+2j_2+\ldots+nj_n = n.$$

The practical value of Fáa di Bruno's formula is greatly enhanced by the slow growth of $\chi(n)$. A numerical investigation of $\chi(n)$ for $n \leq 50$ showed that

(59)
$$\begin{aligned}
\chi(n) &= 2, &&\text{for} && 3 \leq n \leq 5,\\
&\ 3, &&\text{for} && 6 \leq n \leq 9,\\
&\ 4, &&\text{for} && 10 \leq n \leq 14,\\
&\ 5, &&\text{for} && 15 \leq n \leq 20,\\
&\ 6, &&\text{for} && 21 \leq n \leq 27,\\
&\ 7, &&\text{for} && 28 \leq n \leq 35,\\
&\ 8, &&\text{for} && 36 \leq n \leq 44,\\
&\ 9, &&\text{for} && 45 \leq n \leq 50.
\end{aligned}$$

Knowledge of the values of $\chi(n)$ is very important for designing computer programs for calculating the busy period moments.

6. Computer Program Considerations and Results.

The authors have designed a system of computer routines to implement the recurrence relations for $\tilde{g}_n$, derived in Section 3. The main problem lies in the large number of terms occuring in the summations used to evalute the coefficients Y_{nr} in Fáa di Bruno's formula. The considerations of Section 5 show that the volume of computation required for the higher moments, dictates efficient program organization.

Since Fáa di Bruno's formula is applied several times, it is advantageous to have a list of the indices $(j_1, j_2, \ldots j_n)$ stored in memory rather than to generate them repeatedly. Therefore the computer routines were divided into two sets.

The first set generates a list of the nonzero indices $(j_1, j_2, \ldots, j_n)$ of summation in Fáa di Bruno's formula. This list may be stored once and for all on a magnetic tape or in a disk storage device. The indices are subsequently read from this storage medium. The second set of programs reads the indices from their storage medium and performs the computations appropriate for the main recurrence relation.

Not all indices are read in from the storage medium at once. In order to conserve memory space, a first set of indices is read and is used to compute all the quantities Y_{nr} in which they occur. A new set of indices, corresponding to higher values of n is then read in and is stored in the same locations as those of the previous set.

Independently, a third set of programs was written to calculate the moments for the special queues, discussed in Section 4. The moments were calculated using the methods given there. The results were used for testing the general programs for their correctness and numerical stability.

a. Generation of the Indices

The n-tuples $(j_1,j_2,\ldots,j_n)$ of indices are generated by a subroutine called INDEX, one at a time according to the algorithm discussed in [4]. A subroutine called PRNT then packs several sets of them into a single computer word for transfer to the external storage medium (tape or disk). Thus, the programs take into account the number of bits in each computer word and are therefore machine dependent. This machine dependence was necessary to make fully efficient use of the processing time and the memory storage.

An array of size 10,000 was reserved for the indices thus generated. The indices are formatted into records of as many complete sets of indices as possible. A complete set of indices for n is the set of all nonnegative n-tuples $(j_1,j\ ,\ldots j_n)$ satisfying the conditions $\sum_{i=1}^{n} ij_i = n$. Thus the first tape record contains indices for n=3,...,24, the second for n=25,26, and 27, and similarly for the further records. A magnetic tape was written containing the complete sets of indices for n from three to forty nine. This was the maximum number of complete sets of indices which would fit on a single 2400 foot reel of 7 channel tape, recorded at 800 lines per inch.

b. Computation of moments

The core of the moment computation routines is a subroutine called FORMY which evaluates the coefficients Y_{nr} in Fáa di Bruno's formula. The computations are organized so as to utilize the packed format of the indices without unpacking them. Furthermore, the subroutine is self contained i.e., does not call upon the system routines for functions, such as raising numbers to powers etc. At each stage necessary indices are read into memory from the tape and the coefficients $Y_{nr}^{(1)}$, $Y_{nr}^{(2)}$, $Y_{nr}^{(3)}$, given by the Formulas (16), (19) and (22) for the model (a), are computed. After these computations, the memory space is released. This approach considerably shortened the running time and permitted efficient use of memory space, which becomes important after n=20. Unfortunately, this approach necessitated writing the subroutine in assembly language.

The required input data for each moment calculation are N, the number of moments desired and λ the bunch arrival rate. In addition, pairs of subroutines, called ALPHA and ETA, which depend on the service time and bunch size distributions, are required to calculate the moments of these distributions.

The output first gives a summary of the input data, as well as the traffic intensity. If necessary the service time distribution is normalized to obtain $\alpha_1 = 1$. If such a normalization occurs, an appropriate message is given in the output. In addition, the moments of the service time and the bunch size distributions are given, along with the busy period moments.

c. Computational accuracy.

The authors have written programs for calculating busy period moments for the two types of queues, discussed in Section 4. The moment calculations used the recurrence relationships derived in Section 4, rather than Faa di Bruno's formula. Selected results are presented in tables I and II. The same moments were calculated using the general algorithm and the results of this calculation are given in tables III and IV. There is excellent agreement for all forty moments calculated. This is truly remarkable in view of the rather large number of summands in Faa di Bruno's formula.

Examples.

The first twenty four moments for queues with various service time and bunch size distributions have been calculated and are presented in tables V through XIV.

For these examples we needed recurrence relations for the moments of the various service time and bunch size distributions.

Specifically, for the geometric distribution with the moment generating function $\phi(s) = pe^s(1-qe^s)^{-1}$, we derived the relationship

(60)
$$\mu_1 = 1/p,$$
$$\mu_{r+1} = \frac{\mu_r}{p} + \frac{q}{p}\sum_{j=1}^{r}\binom{r}{j}\mu_{r-j+1}, \quad r = 1,2,\ldots$$

using the methods, described in [3] p. 122. Similar methods were used for the negative binomial distribution with the moment generating function $\phi(s) = p^v(1-qe^s)^{-v}$ to yield

(61)
$$\mu_0 = 1$$
$$\mu_1 = vq/p$$
$$\mu_{r+1} = \frac{q}{p}\left[v\,\mu_r + \sum_{j=1}^{r}\binom{r}{j}[v\,\mu_{r-j} + \mu_{r-j+1}]\right], \quad r = 1,2,\ldots$$

The Poisson distribution with mean λ has moments satisfying

(62) $$\mu_0 = 1$$

$$\mu_1 = \lambda$$

$$\mu_{r+1} = \lambda \sum_{j=0}^{r} \binom{r}{j} \mu_j \qquad r = 1,2,\ldots.$$

The gamma distribution with the density function $x^{\alpha-1}\beta^{\alpha}e^{-x/\beta}/\,\Gamma(\alpha)$, $x > o$, has moments satisfying

(63) $$\mu_1 = \alpha/\beta$$

$$\mu_{r+1} = (\alpha+r)\mu_r/\beta \qquad r = 1,2,\ldots.$$

The moments for the busy period, found in these examples exhibit considerable variation even for queues having the same traffic intensity, but with different service and bunch size distributions.

d. Running Times.

The running times required to generate the indices, as well as the moment calculation times are listed below. The time required by the system to load and set up the program was subtracted from the running time to give a better idea of the actual computation times.

Number of Indices	Index Generation (Seconds)	Moment Computation (Seconds)
10	.066	.189
20	.568	1.263
30	5.888	14.661
40	46.802	141.233
50	≐ 300 sec	?

These are the central processing times for the CDC 6500 computer at Purdue University.

7. Conclusions.

Our results adequately substantiate the claim that differentiation of the moment generating function using Fáa di Bruno's formula is entirely practical for obtaining the busy period moments of a single server queue with bunch arrivals. Up to thirty moments may be obtained using a very small amount of computer time; even up to forty moments may be calculated with a moderate amount of computing time. Moments, beyond order forty require considerable amounts of additional computing time.

The authors furthermore report that no problems of numerical instability or round off errors arose.

The varied behavior of the moments for various bunch size and service time distributions, corresponding to the same traffic intensity shows that these considerations are worthwhile, especially in view of the dearth of explicit and usable analytic results.

If a large amount of computer time is available, one could readily calculate up to fifty moments.

For further information on the computer programs described here, one may contact either of the authors at the Department of Statistics, Purdue University, W. Lafayette, Indiana, 47906.

BIBLIOGRAPHY

(1) Dafermos, D. Stella C. and Neuts, Marcel F. (1971):
A Single Server Queue in Discrete time.
Cahiers du Centre de Recherche Opérationnelle, 13, 23-40.

(2) Fáa di Bruno, C. (1876):
Théorie des Formes Binaires.
Librairie Brero, Succ[r] de P. Marietti, Turin, Italy.

(3) Kendal, M. G. and Stuart, A. (1969)
The Advanced Theory of Statistics Vol 1. 3rd ed.
Charles Griffin & Co., London.

(4) Klimko, E. M. (1972)
An Algorithm for Calculating Indices in Fáa di Bruno's formula.
Purdue Mimeo Series, No. 277, Department of Statistics, W. Lafayette, IN.

(5) Neuts, Marcel F. (1972)
The Single Server Queue in Discrete time-Numerical Analysis I.
Purdue Mimeo Series, No. 270, Department of Statistics, W. Lafayette, IN.

(6) Takàcs, L. (1961)
The Transient Behavior of a Single Server Process with Poisson Input.
Fourth Berkeley Symposium, Univ. of California Press, 535-567.

(7) Takàcs, L. (1962)
Introduction to the Theory of Queues.
Oxford University Press, New York.

APPENDIX I.

Appendix I.

Selected Examples of Output

Table I.

The first fifty moments of the busy period for the M|M|1 queue with single arrivals and arrival rate $\lambda = .9$ are listed. These moments were computed by the special methods of Section 4 for comparison with those presented in table III. Note that the traffic intensity is .9.

Table II.

The first fifty moments of the busy period for the M|M|1 queue with group arrivals are given. The group size distribution is geometric with parameter p equal to .2 and the group arrival rate λ is .1 corresponding to a traffic intensity of .5. These moments were computed by the methods of section 4 for comparison with the results obtained by the general method presented in table IV.

Table III.

This table lists the first forty moments of the same queue whose moments are listed in table I. These moments were computed by the general methods developed in this paper.

Table IV.

The first forty moments of the queue, whose moments are given in table II, are listed for comparison. These moments were computed by the general methods using Faà di Bruno's formula.

Tables V through XIV list the first twenty four busy period moments for single server queues with group arrivals and various service time and bunch size distributions. In addition, four different traffic intensities are given for each type of queue. The traffic intensities in each case are .1, .5, .9 and .99. In each case, the mean service time is one.

Table V.

The service time distribution is exponential with mean one and the arrivals are single.

Table VI.

The service time distribution is exponential with mean one and the group arrival distribution is Poisson with mean five.

Table VII.

The service time distribution is exponential with mean one and the group arrival distribution is negative binomial with

$$\text{(A.1)} \quad P(X=k) = \left(\frac{2}{3}\right)^2 \left(\frac{1}{3}\right)^k \binom{k+1}{k}.$$

Table VIII.

The service time distribution is exponential with mean one and the group arrival distribution is geometric with parameter $p=.2$.

Table IX.

The service time distribution is exponential with mean one and the group size distribution is geometric with parameter p=.5.

Table X.

The service time distribution is gamma with density

$$(A.2) \qquad f(x) = \frac{2^2 x e^{-2x}}{\Gamma(2)} .$$

The group size distribution is Poisson with mean five.

Table XI.

The service time distribution is gamma with density (A.2) and the group size distribution is negative binomial with distribution (A.1)

Table XII.

The service time distribution is gamma with density (A.2) and the group size distribution is geometric with parameter p=.2.

Table XIII.

The service time distribution is gamma with density (A.2) and the arrivals are single.

Table XIV.

The service time distribution is a mixture of gamma distributions with density

$$\text{(A.3)} \qquad f(x) = \frac{1}{2}\left(\frac{2^2 x e^{-2x}}{\Gamma\ (2)}\right) + \left(\frac{1}{2}\ \ \frac{3^3 x^2 e^{-3x}}{\Gamma\ (3)}\right)$$

and the service time distribution is geometric with p=.2.

THE TRAFFIC INTENSITY RHO ECUALS .90000

1	1.0000000E+01	2	2.0C00000E+03
3	1.1400C00E+06	4	1.0824000E+09
5	1.4386800E+12	6	2.4585192E+15
7	5.1348523E+18	8	1.2674480E+22
9	3.6097622E+25	10	1.1651547E+29
11	4.2033206E+32	12	1.6759714E+36
13	7.3189511E+39	14	3.4741052E+43
15	1.7809864E+47	16	9.8064603E+50
17	5.7720157E+54	18	3.6165532E+58
19	2.4033455E+62	2C	1.6883838E+66
21	1.2502272E+70	22	9.7325330E+73
23	7.9459785E+77	24	6.7891108E+81
25	6.0584804E+85	26	5.6365416E+89
27	5.4580283E+93	28	5.4924298E+97
29	5.7356157+101	30	6.2073712+105
31	6.9536452+109	32	8.0536944+113
33	9.6335967+117	34	1.1889253+122
35	1.5124538+126	36	1.9814537+130
37	2.6711298+134	38	3.7022910+138
39	5.2721106+142	40	7.7077554+146
41	1.1561327+151	42	1.7780558+155
43	2.8020517+159	44	4.5221788+163
45	7.4699833+167	46	1.2622991+172
47	2.1810033+176	48	3.8511630+180
49	6.9465336+184	50	1.2793592+189

Table 1

LAMBDA = .10000 P = .20C00 RHO = .50000

1	2.0000C00E+00	2	4.8000000E+01
3	4.128CC00E+03	4	6.0134400E+05
5	1.2267264E+08	6	3.2174254E+10
7	1.0313795E+13	8	3.9073265E+15
9	1.7080061E+18	1C	8.4616941E+20
11	4.6852234E+23	12	2.8672747E+26
13	1.9218394E+29	14	1.4C01587E+32
15	1.1016949E+35	16	9.3106456E+37
17	8.4112793E+40	18	8.0890337E+43
19	8.2505978E+46	20	8.8962696E+49
21	1.011C983E+53	22	1.2080868E+56
23	1.5138664E+59	24	1.9852767E+62
25	2.7191922E+65	26	3.8829073E+68
27	5.7709627E+71	28	8.9134374E+74
29	1.4286604E+78	30	2.3731489E+81
31	4.0803569E+84	32	7.2535255E+87
33	1.3317137E+91	34	2.5225826E+94
35	4.9253987E+97	36	9.9040241+100
37	2.0492326+104	38	4.3594864+107
39	9.5283526+110	40	2.1381087+114
41	4.9224138+117	42	1.1619418+121
43	2.8105009+124	44	6.9618381+127
45	1.76508C2+131	46	4.5779993+134
47	1.2140551+138	48	3.2903539+141
49	9.1093524+144	50	2.5750195+148

Table 2

THE MOMENTS OF THE BUSY PERIOD OF A SINGLE SERVER QUEUE
WITH GROUP ARRIVALS AND A POISSON ARRIVAL PROCESS

THE SERVICE TIME DISTRIBUTION IS EXPONENTIAL WITH MEAN 1
THE GROUP SIZE IS 1 WITH PROBABILITY 1
GROUP ARRIVAL RATE LAMBDA = .90000000
THE TRAFFIC INTENSITY RHO = .90000000

N	SERVICE TIME MOMENTS	BUNCH SIZE MOMENTS	BUSY PERIOD MOMENTS
1	1.0000000E+00	1.0000000E+00	1.0000000E+01
2	2.0000000E+00	1.0000000E+00	2.0000000E+03
3	6.0000000E+00	1.0000000E+00	1.1400000E+06
4	2.4000000E+01	1.0000000E+00	1.0824000E+09
5	1.2000000E+02	1.0000000E+00	1.4386800E+12
6	7.2000000E+02	1.0000000E+00	2.4585192E+15
7	5.0400000E+03	1.0000000E+00	5.1348523E+18
8	4.0320000E+04	1.0000000E+00	1.2674480E+22
9	3.6288000E+05	1.0000000E+00	3.6097622E+25
10	3.6288000E+06	1.0000000E+00	1.1651547E+29
11	3.9916800E+07	1.0000000E+00	4.2033206E+32
12	4.7900160E+08	1.0000000E+00	1.6759714E+36
13	6.2270208E+09	1.0000000E+00	7.3189511E+39
14	8.7178291E+10	1.0000000E+00	3.4741052E+43
15	1.3076744E+12	1.0000000E+00	1.7809864E+47
16	2.0922790E+13	1.0000000E+00	9.8064603E+50
17	3.5568743E+14	1.0000000E+00	5.7720157E+54
18	6.4023737E+15	1.0000000E+00	3.6165532E+58
19	1.2164510E+17	1.0000000E+00	2.4033455E+62
20	2.4329020E+18	1.0000000E+00	1.6883838E+66
21	5.1090942E+19	1.0000000E+00	1.2502272E+70
22	1.1240007E+21	1.0000000E+00	9.7325330E+73
23	2.5852017E+22	1.0000000E+00	7.9459785E+77
24	6.2044840E+23	1.0000000E+00	6.7891108E+81
25	1.5511210E+25	1.0000000E+00	6.0584804E+85
26	4.0329146E+26	1.0000000E+00	5.6365416E+89
27	1.0888869E+28	1.0000000E+00	5.4580283E+93
28	3.0488834E+29	1.0000000E+00	5.4924298E+97
29	8.8417620E+30	1.0000000E+00	5.7356157+101
30	2.6525286E+32	1.0000000E+00	6.2073712+105
31	8.2228387E+33	1.0000000E+00	6.9536452+109
32	2.6313084E+35	1.0000000E+00	8.0536944+113
33	8.6833176E+36	1.0000000E+00	9.6335967+117
34	2.9523280E+38	1.0000000E+00	1.1889253+122
35	1.0333148E+40	1.0000000E+00	1.5124538+126
36	3.7199333E+41	1.0000000E+00	1.9814537+130
37	1.3763753E+43	1.0000000E+00	2.6711298+134
38	5.2302262E+44	1.0000000E+00	3.7022910+138
39	2.0397882E+46	1.0000000E+00	5.2721106+142
40	8.1591528E+47	1.0000000E+00	7.7077554+146

Table 3

THE MOMENTS OF THE BUSY PERIOD OF A SINGLE SERVER QUEUE
WITH GROUP ARRIVALS AND A POISSON ARRIVAL PROCESS

THE SERVICE TIME DISTRIBUTION IS EXPONENTIAL WITH MEAN 1
THE GROUP SIZE DISTRIBUTION IS GEOMETRIC WITH MEAN 5
GROUP ARRIVAL RATE LAMBDA = .10000000
THE TRAFFIC INTENSITY RHO = .50000000

N	SERVICE TIME MOMENTS	BUNCH SIZE MOMENTS	BUSY PERIOD MOMENTS
1	1.0000000E+00	5.0000000E+00	2.0000000E+00
2	2.0000000E+00	4.5000000E+01	4.8000000E+01
3	6.0000000E+00	6.0500000E+02	4.1280000E+03
4	2.4000000E+01	1.0845000E+04	8.0134400E+05
5	1.2000000E+02	2.4300500E+05	1.2267264E+08
6	7.2000000E+02	6.5340450E+06	3.2174254E+10
7	5.0400000E+03	2.0497260E+08	1.0313795E+13
8	4.0320000E+04	7.3485468E+09	3.9073265E+15
9	3.6288000E+05	2.9639733E+11	1.7080061E+18
10	3.6288000E+06	1.3282361E+13	8.4616941E+20
11	3.9916800E+07	6.5476226E+14	4.6852234E+23
12	4.7900160E+08	3.5211177E+16	2.8672747E+26
13	6.2270208E+09	2.0513490E+18	1.9218394E+29
14	8.7178291E+10	1.2870139E+20	1.4001587E+32
15	1.3076744E+12	8.6514753E+21	1.1016949E+35
16	2.0922790E+13	6.2033433E+23	9.3106456E+37
17	3.5568743E+14	4.7259638E+25	8.4112793E+40
18	6.4023737E+15	3.8122253E+27	8.0890337E+43
19	1.2164510E+17	3.2459948E+29	8.2505978E+46
20	2.4329020E+18	2.9093333E+31	8.8962696E+49
21	5.1090942E+19	2.7379684E+33	1.0110983E+53
22	1.1240007E+21	2.6993970E+35	1.2080868E+56
23	2.5852017E+22	2.7823404E+37	1.5138664E+59
24	6.2044840E+23	2.9925207E+39	1.9852767E+62
25	1.5511210E+25	3.3526856E+41	2.7191922E+65
26	4.0329146E+26	3.9064461E+43	3.8829073E+68
27	1.0888869E+28	4.7267351E+45	5.7709627E+71
28	3.0488834E+29	5.9310960E+47	8.9134374E+74
29	8.8417620E+30	7.7081226E+49	1.4286604E+78
30	2.6525286E+32	1.0363001E+52	2.3731489E+81
31	8.2228387E+33	1.4396697E+54	4.0803569E+84
32	2.6313084E+35	2.0645648E+56	7.2535255E+87
33	8.6833176E+36	3.0532201E+58	1.3317137E+91
34	2.9523280E+38	4.6521391E+60	2.5225826E+94
35	1.0333148E+40	7.2968664E+62	4.9253987E+97
36	3.7199333E+41	1.1772117E+65	9.9040241+100
37	1.3763753E+43	1.9519546E+67	2.0492326+104
38	5.2302262E+44	3.3240779E+69	4.3594864+107
39	2.0397882E+46	5.8096700E+71	9.5283526+110
40	8.1591528E+47	1.0414229E+74	2.1381087+114

Table 4

THE MOMENTS OF THE BUSY PERIOD OF A SINGLE SERVER QUEUE
WITH GROUP ARRIVALS AND A POISSON ARRIVAL PROCESS

THE SERVICE TIME DISTRIBUTION IS EXPONENTIAL WITH MEAN 1
THE GROUP SIZE IS 1 WITH PROBABILITY 1

THE GROUP ARRIVAL RATE LAMBDA =			.30000000	.40000000	.50000000	.60000000
THE TRAFFIC INTENSITY RHO =			.30000000	.40000000	.50000000	.60000000

N	SERVICE TIME MOMENTS	BUNCH SIZE MOMENTS	BUSY PERIOD MOMENTS	BUSY PERIOD MOMENTS	BUSY PERIOD MOMENTS	BUSY PERIOD MOMENTS
1	1.0000000E+00	1.0000000E+00	1.4285714E+00	1.6666667E+00	2.0000000E+00	2.5000000E+00
2	2.0000000E+00	1.0000000E+00	5.5309038E+00	9.2592593E+00	1.6000000E+01	3.1250000E+01
3	6.0000000E+00	1.0000000E+00	4.6409234E+01	1.0802469E+02	2.8800000E+02	9.3750000E+02
4	2.4000000E+01	1.0000000E+00	5.7593329E+02	2.0233196E+03	8.4480000E+03	4.6289062E+04
5	1.2000000E+02	1.0000000E+00	1.0012488E+04	5.2678708E+04	3.4560000E+05	3.1933594E+06
6	7.2000000E+02	1.0000000E+00	2.2132063E+05	1.7594498E+06	1.8155520E+07	2.8306274E+08
7	5.0400000E+03	1.0000000E+00	5.9685414E+06	7.1753438E+07	1.1650867E+09	3.0657898E+10
8	4.0320000E+04	1.0000000E+00	1.9004516E+08	3.4564773E+09	8.8334991E+10	3.9236068E+12
9	3.6288000E+05	1.0000000E+00	6.9783482E+09	1.9206072E+11	7.7264526E+12	5.7934364E+14
10	3.6288000E+06	1.0000000E+00	2.9030337E+11	1.2092464E+13	7.6583774E+14	9.6943499E+16
11	3.9916800E+07	1.0000000E+00	1.3494337E+13	8.5081856E+14	8.4833038E+16	1.8129593E+19
12	4.7900160E+08	1.0000000E+00	6.9317424E+14	6.6150088E+16	1.0385691E+19	3.7472308E+21
13	6.2270208E+09	1.0000000E+00	3.8993039E+16	5.6338673E+18	1.3925055E+21	8.4826588E+23
14	8.7178291E+10	1.0000000E+00	2.3839793E+18	5.2145763E+20	2.0293521E+23	2.0871738E+26
15	1.3076744E+12	1.0000000E+00	1.5740172E+20	5.2123913E+22	3.1939740E+25	5.5463014E+28
16	2.0922790E+13	1.0000000E+00	1.1161569E+22	5.5959629E+24	5.3992254E+27	1.5829900E+31
17	3.5568743E+14	1.0000000E+00	8.4602705E+23	6.4219176E+26	9.7563791E+29	4.8296207E+33
18	6.4023737E+15	1.0000000E+00	6.8261960E+25	7.8450802E+28	1.8766910E+32	1.5685459E+36
19	1.2164510E+17	1.0000000E+00	5.8413538E+27	1.0164272E+31	3.8286575E+34	5.4029776E+38
20	2.4329020E+18	1.0000000E+00	5.2840920E+29	1.3921381E+33	8.2571513E+36	1.9674367E+41
21	5.1090942E+19	1.0000000E+00	5.0382569E+31	2.0097667E+35	1.8770407E+39	7.5514361E+43
22	1.1240007E+21	1.0000000E+00	5.0501136E+33	3.0501660E+37	4.4857361E+41	3.0470258E+46
23	2.5852017E+22	1.0000000E+00	5.3088371E+35	4.8549172E+39	1.1242840E+44	1.2894547E+49
24	6.2044840E+23	1.0000000E+00	5.8403046E+37	8.0868745E+41	2.9489023E+46	5.7105639E+51

Table 5

THE MOMENTS OF THE BUSY PERICC CF A SINGLE SERVER QUEUE
WITH GRCUP ARRIVALS ANC A POISSCN ARRIVAL PROCESS

THE SERVICE TIPE CISTRIBLTION IS EXPCNENTIAL WITH MEAN 1
THE GROUP SIZE IS POISSON WITH MEAN 5

THE GROUP ARRIVAL RATE LAMEDA =	.0200000C	.1C00CCC0	.180C0000	.19800000
THE TRAFFIC INTENSITY RHC =	.10000C0C	.5000CCC0	.900C0000	.99000000

N	SERVICE TIME MOMENTS	EUNCH SIZE MOMENTS	BUSY PERIOD MOMENTS	BLSY PERICD MCMENTS	BUSY PERICC PCMENTS	BUSY PERIOD MOMENTS
1	1.0000000E+00	5.0000000E+00	1.1111111E+0C	2.CCOCCOOE+CO	1.000000CE+01	1.0000000E+02
2	2.0000000E+00	3.0000000E+01	3.4293553E+0C	3.6COCCOOE+C1	6.50CC00CE+03	6.9500000E+06
3	6.0000000E+00	2.0500000E+02	2.6418737E+01	2.048C000E+03	1.2840000E+07	1.4509500E+12
4	2.4000C00E+01	1.5550000E+03	3.8718629E+02	1.5644800E+C5	4.227990CE+1C	5.0485817E+17
5	1.2000CC0E+02	1.2880000E+04	8.2607144E+03	2.639E800E+C7	1.9491213E+14	2.4593166E+23
6	7.2000CC0E+02	1.1515500E+05	2.2700494E+05	4.561C867E+C9	1.1552897E+18	1.5402953E+29
7	5.0400C00E+03	1.1017050E+06	7.6090507E+06	9.6331906E+11	8.3694003E+21	1.1790812E+35
8	4.0320000E+04	1.1202680E+07	3.0115589E+0E	2.4C45950E+14	7.1655427E+25	1.0666799E+41
9	3.6288C00E+05	1.2041575E+08	1.3749942E+1C	6.9258548E+16	7.0786777E+29	1.1134544E+47
10	3.6288C00E+06	1.3620572E+09	7.1145940E+11	2.2608417E+19	7.9252523E+33	1.3172506E+53
11	3.9916800E+07	1.6151604E+10	4.1143617E+13	8.2485366E+21	9.916968CE+37	1.7416828E+59
12	4.7900160E+08	2.0014402E+11	2.6297929E+15	3.3262366E+24	1.3715465E+42	2.5452792E+65
13	6.22702C8E+09	2.5844290E+12	1.8409873E+17	1.4690640E+27	2.0775468E+46	4.0738997E+71
14	8.7178291E+10	3.4691479E+13	1.4008578E+19	7.0524920E+29	3.42C6081E+50	7.0875707E+77
15	1.3076744E+12	4.8304031E+14	1.1512322E+21	3.6565366E+32	6.0824652E+54	1.3317056E+84
16	2.0922790E+13	6.9633138E+15	1.0161733E+23	2.C362580E+35	1.1616899E+59	2.6875298E+90
17	3.5568743E+14	1.0374736E+17	9.5882192E+24	1.2121614E+38	2.3717262E+63	5.7977843E+96
18	6.4023737E+15	1.5951321E+18	9.6307882E+26	7.6814145E+4C	5.1545546E+67	1.3314445+103
19	1.2164510E+17	2.5273636E+19	1.0259852E+29	5.1626989E+43	1.1881516E+72	3.2429344+109
20	2.4329020E+18	4.1213308E+20	1.1554608E+31	3.6681466E+46	2.8952515E+76	8.3500092+115
21	5.1090942E+19	6.9087619E+21	1.3716183E+33	2.7471348E+49	7.4364150E+8C	2.2662020+122
22	1.12400C7E+21	1.1893003E+23	1.711715CE+35	2.1628826E+52	2.0079838E+85	6.4659084+128
23	2.5852017E+22	2.1003093E+24	2.2403459E+37	1.7859566E+55	5.6864524E+89	1.9348401+135
24	6.2044840E+23	3.8017129E+25	3.0686225E+39	1.5433103E+58	1.6852591E+94	6.0590516+141

Table 6

THE MOMENTS OF THE BUSY PERICC CF A SINGLE SERVER QUEUE
WITH GRCUP ARRIVALS ANC A POISSCN ARRIVAL PROCESS

THE SERVICE TIME DISTRIBUTION IS EXPCNENTIAL WITH MEAN 1
THE GROUP SIZE CISTRIBUTION IS NEGATIVE BINOMIAL WITH P= 2/3 AND V=2

THE GROUP ARRIVAL RATE LAMBDA =		.10CCCCCC	.5C000CCO	.900C0000	.99C00000
THE TRAFFIC INTENSITY RHC =		.100C000C	.50000CCO	.90000000	.99000000

N	SERVICE TIME MOMENTS	BUNCH SIZE MOMENTS	BUSY PERIOD MOMENTS	BUSY PERIOD MOMENTS	BUSY PERIOD MOMENTS	BUSY PERIOD MOMENTS
1	1.CCCCCC00E+00	1.0000000E+00	1.1111111E+0C	2.CCOCC00E+C0	1.0CCC00CE+01	1.C000000E+02
2	2.0000000E+00	2.5000000E+00	2.9492455E+0C	2.2C0CC00E+C1	3.35C000CE+03	3.4850000E+06
3	6.0000CC0E+00	8.5C00000E+00	1.4797033E+01	6.5400000E+02	3.333750CE+06	3.6405375E+11
4	2.4000000E+01	3.7000000E+01	1.2299500E+02	3.2898C00E+04	5.5318312E+09	6.3383882E+16
5	1.2000000E+02	1.9600000E+02	1.4907308E+03	2.3253300E+C6	1.2851755E+13	1.5449715E+22
6	7.2000000E+02	1.2212500E+03	2.3834476E+04	2.1155607E+C8	3.8389301E+16	4.8417979E+27
7	5.0400000E+03	8.7422500E+03	4.7117411E+05	2.3535503E+10	1.4015630E+20	1.8545693E+33
8	4.032C000E+04	7.C657000E+04	1.1061081E+07	3.C551464E+12	6.0473918E+23	8.3951827E+38
9	3.6288CC0E+05	6.3615100E+05	3.0025223E+08	4.6573353E+14	3.0107378E+27	4.3849552E+44
10	3.6288CC0E+06	6.3122275E+06	9.247C993E+09	8.0801838E+16	1.6987769E+31	2.5957200E+50
11	3.9916800E+07	6.8425913E+07	3.1850213E+11	1.5535455E+19	1.0712852E+35	1.7173358E+56
12	4.7900160E+08	8.0446745E+08	1.2130388E+13	3.3C15C98E+21	7.4668937E+38	1.2557936E+62
13	6.2270208E+09	1.0194550E+10	5.0615355E+14	7.6846599E+23	5.7C01165E+42	1.0057496E+68
14	8.7178291E+10	1.3851699E+11	2.2961734E+16	1.9442862E+26	4.7297689E+46	8.7553533E+73
15	1.3076744E+12	2.0087332E+12	1.1252009E+18	5.3128587E+28	4.2385769E+5C	8.2315423E+79
16	2.0922790E+13	3.0965793E+13	5.9231422E+19	1.5593282E+31	4.0797598E+54	8.3123342E+85
17	3.5568743E+14	5.0564208E+14	3.3334052E+21	4.8923340E+33	4.1977151E+58	8.9728099E+91
18	6.4023737E+15	8.7183746E+15	1.9971797E+23	1.6339979E+36	4.5977321E+62	1.C310653E+98
19	1.2164510E+17	1.5828235E+17	1.2692115E+25	5.7882166E+38	5.3410736E+66	1.2566028+104
20	2.4329020E+18	3.0180689E+18	8.5273225E+26	2.1675789E+41	6.5591447E+70	1.6189876+110
21	5.1090942E+19	6.0301050E+19	6.0391803E+28	8.5559972E+43	8.4904128E+74	2.1986288+116
22	1.1240007E+21	1.2598222E+21	4.4965845E+3C	3.5504867E+46	1.1553925E+79	3.1389150+122
23	2.5852017E+22	2.7469472E+22	3.5114741E+32	1.5452249E+49	1.6489794E+83	4.6999344+128
24	6.2044840E+23	6.2400030E+23	2.8698238E+34	7.0378710E+51	2.4628892E+87	7.3645819+134

Table 7

THE MOMENTS OF THE BUSY PERIOD OF A SINGLE SERVER QUEUE
WITH GROUP ARRIVALS AND A POISSON ARRIVAL PROCESS

THE SERVICE TIME DISTRIBUTION IS EXPONENTIAL WITH MEAN 1
THE GROUP SIZE DISTRIBUTION IS GEOMETRIC WITH MEAN 5

THE GROUP ARRIVAL RATE LAMBDA =			.02000000	.10000000	.18000000	.19800000
THE TRAFFIC INTENSITY RHO =			.10000000	.50000000	.90000000	.99000000

N	SERVICE TIME MOMENTS	BUNCH SIZE MOMENTS	BUSY PERIOD MOMENTS	BUSY PERIOD MOMENTS	BUSY PERIOD MOMENTS	BUSY PERIOD MOMENTS
1	1.0000000E+00	5.0000000E+00	1.1111111E+00	2.0000000E+00	1.0000000E+01	1.0000000E+02
2	2.0000000E+00	4.5000000E+01	3.8408779E+00	4.8000000E+01	9.2000000E+03	9.9200000E+06
3	6.0000000E+00	6.0500000E+02	4.5318295E+01	4.1280000E+03	2.6196000E+07	2.9610960E+12
4	2.4000000E+01	1.0845000E+04	1.1788828E+03	6.0134400E+05	1.2434064E+11	1.4731354E+18
5	1.2000000E+02	2.4300500E+05	4.4964044E+04	1.2267264E+08	8.2628047E+14	1.0260322E+24
6	7.2000000E+02	6.5340450E+06	2.1956752E+06	3.2174254E+10	7.0597463E+18	9.1880597E+29
7	5.0400000E+03	2.0497260E+08	1.3028678E+08	1.0313795E+13	7.3722852E+22	1.0056267E+36
8	4.0320000E+04	7.3485468E+09	9.1063297E+09	3.9073265E+15	9.0984309E+26	1.3007699E+42
9	3.6288000E+05	2.9638733E+11	7.3308137E+11	1.7080061E+18	1.2956246E+31	1.9413868E+48
10	3.6288000E+06	1.3282361E+13	6.6813913E+13	8.4616941E+20	2.0909796E+35	3.2838350E+54
11	3.9916800E+07	6.5476226E+14	6.8014987E+15	4.6852234E+23	3.7715968E+39	6.2080507E+60
12	4.7900160E+08	3.5211177E+16	7.6492759E+17	2.8672747E+26	7.5191118E+43	1.2971640E+67
13	6.2270208E+09	2.0513490E+18	9.4191734E+19	1.9218394E+29	1.6417857E+48	2.9685410E+73
14	8.7178291E+10	1.2870139E+20	1.2604368E+22	1.4001587E+32	3.8965354E+52	7.3841991E+79
15	1.3076744E+12	8.6514753E+21	1.8212953E+24	1.1016949E+35	9.9876797E+56	1.9837525E+86
16	2.0922790E+13	6.2033433E+23	2.8262956E+26	9.3106456E+37	2.7496971E+61	5.7240818E+92
17	3.5568743E+14	4.7259638E+25	4.6878525E+28	8.4112793E+40	8.0922418E+65	1.7655819E+99
18	6.4023737E+15	3.8122253E+27	8.2764993E+30	8.0890337E+43	2.5351585E+70	5.7972514+105
19	1.2164510E+17	3.2459948E+29	1.5496879E+33	8.2505978E+46	8.4235567E+74	2.0188801+112
20	2.4329020E+18	2.9093333E+31	3.0672643E+35	8.8962696E+49	2.9588271E+79	7.4324600+118
21	5.1090942E+19	2.7379684E+33	6.3988325E+37	1.0110983E+53	1.0954854E+84	2.8841479+125
22	1.1240007E+21	2.6993970E+35	1.4033059E+40	1.2080868E+56	4.2639572E+88	1.1765807+132
23	2.5852017E+22	2.7823404E+37	3.2275567E+42	1.5138664E+59	1.7406190E+93	5.0339687+138
24	6.2044840E+23	2.9925207E+39	7.7683154E+44	1.9852767E+62	7.4359874E+97	2.2539453+145

Table 8

THE MOMENTS OF THE BUSY PERIOD OF A SINGLE SERVER QUEUE
WITH GROUP ARRIVALS AND A POISSON ARRIVAL PROCESS

THE SERVICE TIME DISTRIBUTION IS EXPONENTIAL WITH MEAN 1
THE GROUP SIZE DISTRIBUTION IS GEOMETRIC

THE GROUP ARRIVAL RATE LAMBDA =			.05000000	.10000000	.45000000	.49500000
THE TRAFFIC INTENSITY RHO =			.10000000	.20000000	.90000000	.99000000
N	SERVICE TIME MOMENTS	BUNCH SIZE MOMENTS	BUSY PERIOD MOMENTS	BUSY PERIOD MOMENTS	BUSY PERIOD MOMENTS	BUSY PERIOD MOMENTS
1	1.0000000E+00	2.0000000E+00	1.1111111E+00	1.2500000E+00	1.0000000E+01	1.0000000E+02
2	2.0000000E+00	6.0000000E+00	3.0178326E+00	4.6875000E+00	3.8000000E+03	3.9800000E+06
3	6.0000000E+00	2.6000000E+01	1.6359295E+01	4.1015625E+01	4.3260000E+06	4.7520600E+11
4	2.4000000E+01	1.5000000E+02	1.5510032E+02	6.3720703E+02	8.2125600E+09	9.4565278E+16
5	1.2000000E+02	1.0820000E+03	2.2016698E+03	1.4328003E+04	2.1828972E+13	2.6345706E+22
6	7.2000000E+02	9.3660000E+03	4.1511967E+04	4.1868210E+05	7.4600884E+16	9.4369694E+27
7	5.0400000E+03	9.4586000E+04	9.6822759E+05	1.5006852E+07	3.1160905E+20	4.1314783E+33
8	4.0320000E+04	1.0916700E+06	2.6801617E+07	6.3665155E+08	1.5382596E+24	2.1376131E+39
9	3.6288000E+05	1.4174522E+07	8.5744278E+08	3.1189605E+10	8.7619097E+27	1.2761468E+45
10	3.6288000E+06	2.0449513E+08	3.1112766E+10	1.7325631E+12	5.6562250E+31	8.6343544E+50
11	3.9916800E+07	3.2452651E+09	1.2623322E+12	1.0760076E+14	4.0809381E+35	6.5292572E+56
12	4.7900160E+08	5.6183135E+10	5.6624958E+13	7.3876716E+15	3.2543148E+39	5.4571185E+62
13	6.2270208E+09	1.0537167E+12	2.7825834E+15	5.5562492E+17	2.8422878E+43	4.9954145E+68
14	8.7178291E+10	2.1282686E+13	1.4865339E+17	4.5427974E+19	2.6982925E+47	4.9704059E+74
15	1.3076744E+12	4.6056638E+14	8.5779406E+18	4.0117287E+21	2.7665210E+51	5.3411643E+80
16	2.0922790E+13	1.0631309E+16	5.3170581E+20	3.8054651E+23	3.0465853E+55	6.1647328E+86
17	3.5568743E+14	2.6074153E+17	3.5233889E+22	3.8589997E+25	3.5863821E+59	7.6059991E+92
18	6.4023737E+15	6.7710693E+18	2.4856062E+24	4.1659697E+27	4.4941974E+63	9.9896560E+98
19	1.2164510E+17	1.8560317E+20	1.8598732E+26	4.7701260E+29	5.9731295E+67	1.3915503+105
20	2.4329020E+18	5.3553756E+21	1.4712588E+28	5.7742095E+31	8.3923866E+71	2.0491840+111
21	5.1090942E+19	1.6224965E+23	1.2268050E+30	7.3676842E+33	1.2428880E+76	3.1807232+117
22	1.1240007E+21	5.1496888E+24	1.0754654E+32	9.8832481E+35	1.9350750E+80	5.1902713+123
23	2.5852017E+22	1.7087690E+26	9.8881493E+33	1.3904726E+38	3.1597177E+84	8.8825742+129
24	6.2044840E+23	5.9165582E+27	9.5145661E+35	2.0472807E+40	5.3993670E+88	1.5908590+136

Table 9

THE MOMENTS OF THE BUSY PERIOD OF A SINGLE SERVER QUEUE
WITH GROUP ARRIVALS AND A POISSON ARRIVAL PROCESS

THE SERVICE TIME DISTRIBUTION IS GAMMA WITH PARAMETERS ALPHA = 2.000000 AND BETA = 2.000000
THE GROUP SIZE IS POISSON WITH MEAN 5

THE GROUP ARRIVAL RATE LAMBDA =	.02000000	.10000000	.18000000	.19800000
THE TRAFFIC INTENSITY RHO =	.10000000	.50000000	.90000000	.99000000

N	SERVICE TIME MOMENTS	BUNCH SIZE MOMENTS	BUSY PERIOD MOMENTS	BUSY PERIOD MOMENTS	BUSY PERIOD MOMENTS	BUSY PERIOD MOMENTS
1	1.0000000E+00	5.0000000E+00	1.1111111E+00	2.0000000E+00	1.0000000E+01	1.0000000E+02
2	1.5000000E+00	3.0000000E+01	2.7434842E+00	3.2000000E+01	6.0000000E+03	6.4500000E+06
3	3.0000000E+00	2.0500000E+02	1.8416908E+01	1.6760000E+03	1.0987500E+07	1.2501750E+12
4	7.5000000E+00	1.5550000E+03	2.4724444E+02	1.4770000E+05	3.3537562E+10	4.0386006E+17
5	2.2500000E+01	1.2880000E+04	4.7631701E+03	1.8220780E+07	1.4331622E+14	1.8264974E+23
6	7.8750000E+01	1.1515500E+05	1.1710928E+05	2.8899667E+09	7.8741697E+17	1.0620665E+29
7	3.1500000E+02	1.1017050E+06	3.5050683E+06	5.6023765E+11	5.2876763E+21	7.5480458E+34
8	1.4175000E+03	1.1202680E+07	1.2384466E+08	1.2835274E+14	4.1963904E+25	6.3396899E+40
9	7.0875000E+03	1.2041575E+08	5.0475626E+09	3.3930309E+16	3.8426868E+29	6.1439747E+46
10	3.8981250E+04	1.3620572E+09	2.3312438E+11	1.0165517E+19	3.9879697E+33	6.7482083E+52
11	2.3388750E+05	1.6151604E+10	1.2032620E+13	3.4038999E+21	4.6256574E+37	8.2838500E+58
12	1.5202687E+06	2.0014402E+11	6.8639024E+14	1.2597678E+24	5.9300837E+41	1.1239362E+65
13	1.0641881E+07	2.5844290E+12	4.2881613E+16	5.1063818E+26	8.3263902E+45	1.6701660E+71
14	7.9814109E+07	3.4691479E+13	2.9118516E+18	2.2498246E+29	1.2707640E+50	2.6976761E+77
15	6.3851287E+08	4.8304031E+14	2.1354070E+20	1.0705529E+32	2.0945793E+54	4.7059110E+83
16	5.4273594E+09	6.9633138E+15	1.6819702E+22	5.4714460E+34	3.7081986E+58	8.8172235E+89
17	4.8846235E+10	1.0374736E+17	1.4161635E+24	2.9892346E+37	7.0176687E+62	1.7659715E+96
18	4.6403923E+11	1.5951321E+18	1.2692751E+26	1.7384825E+40	1.4137578E+67	3.7651976+102
19	4.6403923E+12	2.5273636E+19	1.2065580E+28	1.0723459E+43	3.0207254E+71	8.5142415+108
20	4.8724119E+13	4.1213308E+20	1.2124700E+30	6.9925127E+45	6.8230914E+75	2.0353440+115
21	5.3596531E+14	6.9087619E+21	1.2842630E+32	4.8061202E+48	1.6244787E+80	5.1285263+121
22	6.1636011E+15	1.1893003E+23	1.4300587E+34	3.4727645E+51	4.0659881E+84	1.3585217+128
23	7.3963213E+16	2.1003093E+24	1.6700768E+36	2.6317263E+54	1.0673400E+89	3.7742024+134
24	9.2454016E+17	3.8017129E+25	2.0410864E+38	2.0871335E+57	2.9321321E+93	1.0973062+141

Table 10

THE MOMENTS OF THE BUSY PERIOD OF A SINGLE SERVER QUEUE
WITH GROUP ARRIVALS AND A POISSON ARRIVAL PROCESS

THE SERVICE TIME DISTRIBUTION IS GAMMA WITH PARAMETERS ALPHA = 2.000000 AND BETA = 2.000000
THE GROUP SIZE DISTRIBUTION IS NEGATIVE BINOMIAL WITH P= 2/3 AND V=2

THE GROUP ARRIVAL RATE LAMBDA =		.10000000	.50000000	.90000000	.99000000
THE TRAFFIC INTENSITY RHO =		.10000000	.50000000	.90000000	.99000000

N	SERVICE TIME MOMENTS	BUNCH SIZE MOMENTS	BUSY PERIOD MOMENTS	BUSY PERIOD MOMENTS	BUSY PERIOD MOMENTS	BUSY PERIOD MOMENTS
1	1.0000000E+00	1.0000000E+00	1.1111111E+00	2.0000000E+00	1.0000000E+01	1.0000000E+02
2	1.5000000E+00	2.5000000E+00	2.2633745E+00	1.8000000E+01	2.8500000E+03	2.9850000E+06
3	3.0000000E+00	8.5000000E+00	8.5733882E+00	4.5000000E+02	2.4262500E+06	2.6722875E+11
4	7.5000000E+00	3.7000000E+01	5.7261766E+01	1.5170000E+04	3.4444312E+09	3.9872455E+16
5	2.2500000E+01	1.9600000E+02	5.7769426E+02	1.1475500E+06	6.8463931E+12	8.3289666E+21
6	7.8750000E+01	1.2212500E+03	7.7247500E+03	8.8481610E+07	1.7496935E+16	2.2369375E+27
7	3.1500000E+02	8.7422500E+03	1.2729827E+05	8.3395198E+09	5.4653438E+19	7.3428860E+32
8	1.4175000E+03	7.0657000E+04	2.4840908E+06	9.2915851E+11	2.0175599E+23	2.8495931E+38
9	7.0875000E+03	6.3615100E+05	5.5964809E+07	1.1946812E+14	8.5937942E+26	1.2750933E+44
10	3.8981250E+04	6.3122275E+06	1.4294139E+09	1.7410615E+16	4.1486062E+30	6.4686135E+49
11	2.3388750E+05	6.8425913E+07	4.0814159E+10	2.8360225E+18	2.2383340E+34	3.6676267E+55
12	1.5202687E+06	8.0446745E+08	1.2882912E+12	5.1061251E+20	1.3347926E+38	2.2983954E+61
13	1.0641881E+07	1.0194550E+10	4.4544410E+13	1.0069245E+23	8.7178948E+41	1.5775128E+67
14	7.9814109E+07	1.3851699E+11	1.6743188E+15	2.1583707E+25	6.1890206E+45	1.1768828E+73
15	6.3851287E+08	2.0087332E+12	6.7975064E+16	4.9967525E+27	4.7452185E+49	9.4823793E+78
16	5.4273594E+09	3.0965793E+13	2.9643525E+18	1.2424844E+30	3.9077333E+53	8.2060782E+84
17	4.8846235E+10	5.0564208E+14	1.3819805E+20	3.3026570E+32	3.4399922E+57	7.5913266E+90
18	4.6403923E+11	8.7183746E+15	6.8588214E+21	9.3452925E+34	3.2236091E+61	7.4757015E+96
19	4.6403923E+12	1.5828235E+17	3.6105267E+23	2.8044604E+37	3.2039187E+65	7.8080094+102
20	4.8724119E+13	3.0180689E+18	2.0092854E+25	8.8582497E+39	3.3663121E+69	8.6210941+108
21	5.3596531E+14	6.0301050E+19	1.1786625E+27	2.9757367E+42	3.7281228E+73	1.0033382+115
22	6.1636011E+15	1.2598222E+21	7.2688958E+28	1.0461790E+45	4.3405559E+77	1.2275849+121
23	7.3963213E+16	2.7469472E+22	4.7015633E+30	3.8574832E+47	5.3001132E+81	1.5752167+127
24	9.2454016E+17	6.2400030E+23	3.1825052E+32	1.4884572E+50	6.7728124E+85	2.1153058+133

Table 11

THE MOMENTS OF THE BUSY PERIOD OF A SINGLE SERVER QUEUE
WITH GROUP ARRIVALS AND A POISSON ARRIVAL PROCESS

THE SERVICE TIME DISTRIBUTION IS GAMMA WITH PARAMETERS ALPHA = 2.000000 AND BETA = 2.000000
THE GROUP SIZE DISTRIBUTION IS GEOMETRIC WITH MEAN 5

THE GROUP ARRIVAL RATE LAMBDA =	.02000000	.10000000	.18000000	.19800000
THE TRAFFIC INTENSITY RHO =	.10000000	.50000000	.90000000	.99000000

N	SERVICE TIME MOMENTS	BUNCH SIZE MOMENTS	BUSY PERIOD MOMENTS	BUSY PERIOD MOMENTS	BUSY PERIOD MOMENTS	BUSY PERIOD MOMENTS
1	1.0000000E+00	5.0000000E+00	1.1111111E+00	2.0000000E+00	1.0000000E+01	1.0000000E+02
2	1.5000000E+00	4.5000000E+01	3.1550069E+00	4.4000000E+01	8.7000000E+03	9.4200000E+06
3	3.0000000E+00	6.0500000E+02	3.5792308E+01	3.6120000E+03	2.3533500E+07	2.6712210E+12
4	7.5000000E+00	1.0845000E+04	9.1710243E+02	5.0019600E+05	1.0610555E+11	1.2624602E+18
5	2.2500000E+01	2.4300500E+05	3.3543757E+04	9.6908580E+07	6.6976002E+14	8.3532143E+23
6	7.8750000E+01	6.5340450E+06	1.5581802E+06	2.4133065E+10	5.4355726E+18	7.1061445E+29
7	3.1500000E+02	2.0497260E+08	8.7798284E+07	7.3445625E+12	5.3916513E+22	7.3886420E+35
8	1.4175000E+03	7.3485468E+09	5.8241815E+09	2.6414730E+15	6.3204658E+26	9.0791645E+41
9	7.0875000E+03	2.9638733E+11	4.4487177E+11	1.0961259E+18	8.5491779E+30	1.2872860E+48
10	3.8981250E+04	1.3282361E+13	3.8465096E+13	5.1549394E+20	1.3105613E+35	2.0685303E+54
11	2.3388750E+05	6.5476226E+14	3.7142414E+15	2.7094766E+23	2.2454068E+39	3.7149536E+60
12	1.5202687E+06	3.5211177E+16	3.9619956E+17	1.5740146E+26	4.2520508E+43	7.3741268E+66
13	1.0641881E+07	2.0513490E+18	4.6270766E+19	1.0014690E+29	8.8188188E+47	1.6031581E+73
14	7.9814109E+07	1.2870139E+20	5.8721065E+21	6.9258989E+31	1.9880833E+52	3.7883864E+79
15	6.3851287E+08	8.6514753E+21	8.0466589E+23	5.1729314E+34	4.8404140E+56	9.6684297E+85
16	5.4273594E+09	6.2033433E+23	1.1841372E+26	4.1498280E+37	1.2657988E+61	2.6502806E+92
17	4.8846235E+10	4.7259638E+25	1.8625030E+28	3.5586562E+40	3.5384295E+65	7.7658940E+98
18	4.6403923E+11	3.8122253E+27	3.1181692E+30	3.2485778E+43	1.0529522E+70	2.4223858+105
19	4.6403923E+12	3.2459948E+29	5.5363049E+32	3.1452403E+46	3.3232338E+74	8.0139987+111
20	4.8724119E+13	2.9093333E+31	1.0390674E+35	3.2191561E+49	1.1087836E+79	2.8027793+116
21	5.3596531E+14	2.7379684E+33	2.0554407E+37	3.4725887E+52	3.8993804E+83	1.0332167+125
22	6.1636011E+15	2.6993970E+35	4.2742891E+39	3.9389285E+55	1.4416627E+88	4.0041763+131
23	7.3963213E+16	2.7823404E+37	9.3215698E+41	4.6852958E+58	5.5900580E+92	1.6274946+138
24	9.2454016E+17	2.9925207E+39	2.1273670E+44	5.8322973E+61	2.2683655E+97	6.9226119+144

Table 12

THE MOMENTS OF THE BUSY PERIOD OF A SINGLE SERVER QUEUE
WITH GROUP ARRIVALS AND A POISSON ARRIVAL PROCESS

THE SERVICE TIME DISTRIBUTION IS GAMMA WITH PARAMETERS ALPHA = 2.000000 AND BETA = 2.000000
THE GROUP SIZE IS 1 WITH PROBABILITY 1

THE GROUP ARRIVAL RATE LAMBDA =			.10000000	.50000000	.90000000	.99000000
THE TRAFFIC INTENSITY RHO =			.10000000	.50000000	.90000000	.99000000

N	SERVICE TIME MOMENTS	BUNCH SIZE MOMENTS	BUSY PERIOD MOMENTS	BUSY PERIOD MOMENTS	BUSY PERIOD MOMENTS	BUSY PERIOD MOMENTS
1	1.0000000E+00	1.0000000E+00	1.1111111E+00	2.0000000E+00	1.0000000E+01	1.0000000E+02
2	1.5000000E+00	1.0000000E+00	2.0576132E+00	1.2000000E+01	1.5000000E+03	1.5000000E+06
3	3.0000000E+00	1.0000000E+00	5.7155921E+00	1.5600000E+02	6.3750000E+05	6.7125000E+10
4	7.5000000E+00	1.0000000E+00	2.2227303E+01	3.3000000E+03	4.5131250E+08	5.0063813E+15
5	2.2500000E+01	1.0000000E+00	1.1427264E+02	9.7380000E+04	4.4726906E+11	5.2274666E+20
6	7.8750000E+01	1.0000000E+00	7.4014784E+02	3.6905400E+06	5.6989337E+14	7.0178322E+25
7	3.1500000E+02	1.0000000E+00	5.8130275E+03	1.7086230E+08	8.8748993E+17	1.1515023E+31
8	1.4175000E+03	1.0000000E+00	5.3761983E+04	9.3462806E+09	1.6333572E+21	2.2329413E+36
9	7.0875000E+03	1.0000000E+00	5.7261155E+05	5.8980831E+11	3.4685279E+24	4.9961755E+41
10	3.8981250E+04	1.0000000E+00	6.9038678E+06	4.2179019E+13	8.3476801E+27	1.2669390E+47
11	2.3388750E+05	1.0000000E+00	9.2958741E+07	3.3709832E+15	2.2453826E+31	3.5906941E+52
12	1.5202687E+06	1.0000000E+00	1.3826591E+09	2.9775625E+17	6.6754429E+34	1.1247780E+58
13	1.0641881E+07	1.0000000E+00	2.2515109E+10	2.8804334E+19	2.1735917E+38	3.8589018E+63
14	7.9814109E+07	1.0000000E+00	3.9839481E+11	3.0286906E+21	7.6928584E+41	1.4390399E+69
15	6.3851287E+08	1.0000000E+00	7.6115913E+12	3.4392706E+23	2.9404990E+45	5.7956971E+74
16	5.4273594E+09	1.0000000E+00	1.5616776E+14	4.1547357E+25	1.2072254E+49	2.5071064E+80
17	4.8846235E+10	1.0000000E+00	3.4245753E+15	5.4689072E+27	5.2980894E+52	1.1593206E+86
18	4.6403923E+11	1.0000000E+00	7.9932231E+16	7.5900363E+29	2.4751545E+56	5.7067200E+91
19	4.6403923E+12	1.0000000E+00	1.9785545E+18	1.1172161E+32	1.2264203E+60	2.9793649E+97
20	4.8724119E+13	1.0000000E+00	5.1769226E+19	1.7384445E+34	6.4240656E+63	1.6443509+103
21	5.3596531E+14	1.0000000E+00	1.4276697E+21	2.8513102E+36	3.5468548E+67	9.5659517+108
22	6.1636011E+15	1.0000000E+00	4.1388256E+22	4.9163739E+38	2.0587141E+71	5.8503402+114
23	7.3963213E+16	1.0000000E+00	1.2583134E+24	8.6505335E+40	1.2532370E+75	3.7524823+120
24	9.2454016E+17	1.0000000E+00	4.0033663E+25	1.6824919E+43	7.9838876E+78	2.5188381+126

Table 13

THE MOMENTS OF THE BUSY PERICC CF A SINGLE SERVER QUEUE
WITH GRCUP ARRIVALS AND A POISSCN ARRIVAL PROCESS

THE SERVICE TIME DISTRIBUTION IS A MIXTURE OF GAMMA DISTRIBUTIONS WITH PARAMETERS 2,2 AND 3,3 AND ALPHA .5
THE GROUP SIZE DISTRIBUTICN IS GEOMETRIC WITH MEAN 5

THE GROUP ARRIVAL RATE LAMEDA =	.020CCCOC	.1COOCCCC	.180C0000	.19800000
THE TRAFFIC INTENSITY RHC =	.100C000C	.5COOCOCO	.90000000	.99000000

N	SERVICE TIME MOMENTS	BUNCH SIZE MOMENTS	BUSY PERIOC MOMENTS	BUSY PERICO MCMENTS	BUSY PERIOD MOMENTS	BUSY PERIOD MOMENTS
1	1.0000000E+00	5.0000000E+00	1.1111111E+0C	2.CCOCCOOE+CC	1.0CCC00CE+01	1.C000000E+02
2	1.4166667E+00	4.5000000E+01	3.0406950E+0C	4.3333333E+C1	8.6166667E+03	9.3366667E+06
3	2.6111111E+00	6.C500000E+02	3.4398691E+01	3.5301111E+C3	2.31C3986E+07	2.6243634E+12
4	5.9722222E+00	1.0845000E+04	8.7959830E+02	4.8466665E+C5	1.0325457E+11	1.2294332E+18
5	1.6435185E+01	2.4300500E+05	3.1934108E+04	9.3C79175E+C7	6.4604067E+14	8.0633215E+23
6	5.3202160E+01	6.5340450E+06	1.4708096E+06	2.2975807E+10	5.1970267E+18	6.7993513E+29
7	1.9898148E+02	2.0497260E+08	8.2155050E+07	6.9308C69E+12	5.1097571E+22	7.0076299E+35
8	8.4702160E+02	7.3485468E+09	5.4021086E+09	2.4706949E+15	5.937399CE+26	8.5354320E+41
9	4.0507459E+03	2.9638733E+11	4.0900276E+11	1.0162154E+18	7.9604962E+30	1.1995760E+48
10	2.1518609E+04	1.3282361E+13	3.5C51733E+13	4.7369585E+2C	1.2C55998E+35	1.9106788E+54
11	1.2573168E+05	6.5476226E+14	3.3547164E+15	2.4678C54E+23	2.0542248E+39	3.4013575E+60
12	8.0114471E+05	3.5211177E+16	3.5467993E+17	1.4209634E+26	3.8558481E+43	6.6924106E+66
13	5.5259923E+06	2.0513490E+18	4.1054593E+19	8.961C669E+28	7.9268460E+47	1.4421865E+73
14	4.1000664E+07	1.2870139E+20	5.1639015E+21	6.1425105E+31	1.7713039E+52	3.3780994E+79
15	3.2545355E+08	8.6514753E+21	7.0133561E+23	4.5473C07E+34	4.2747382E+56	8.5456910E+85
16	2.7508624E+09	6.2033433E+23	1.0229077E+26	3.6157131E+37	1.1080521E+61	2.3219677E+92
17	2.4658608E+10	4.7259638E+25	1.5946075E+28	3.C732406E+4C	3.07C2557E+65	6.7441757E+98
18	2.3358955E+11	3.8122253E+27	2.6459246E+3C	2.7806753E+43	9.0560988E+69	2.0852293+105
19	2.3311857E+12	3.2459948E+29	4.6560493E+32	2.6684383E+46	2.8330997E+74	6.8380592+111
20	2.4442650E+13	2.9093333E+31	8.6608418E+34	2.7C7C543E+49	9.3694966E+78	2.3705307+118
21	2.6860051E+14	2.7379684E+33	1.6980063E+37	2.8946695E+52	3.2661305E+83	8.6620600+124
22	3.0867434E+15	2.6993970E+35	3.4995766E+39	3.254C160E+55	1.1969336E+88	3.3274845+131
23	3.7022797E+16	2.7823404E+37	7.5641008E+41	3.8364056E+58	4.6003533E+92	1.3405885+138
24	4.6262707E+17	2.9925207E+39	1.7109085E+44	4.7233975E+61	1.85C3606E+97	5.6522194+144

Table 14

APPENDIX II.

Appendix II.

Program Listings.

The Fortran and Compass (assembler) listings of the programs used to generate the output in appendix I are given. The program MTAPE generates the indices for Fáa di Bruno's formula and uses the subroutines FLUSH, INDEX, PRNT1 and PRNT. The program BUSY performs the moment calculations using Fáa di Bruno's formula and calls the subroutines GETINDX, SUM and FORMY. A typical example of the subroutines ALPHA and ETA are included together with the subroutines which generate the moments of the geometric (GEO), negative binomial (NEGBIN), Poisson (POIS) and gamma distributions (GAM).

The programs called MOMENTS generated tables I and II. The second MOMENTS program was adapted from the first to generate table II.

```
              PROGRAM MTAPE(TAPE1,TAPE2,INPUT,OUTPUT,TAPE6=OUTPUT)
000002        COMMON/Y/Y(50)
000002        COMMON/B1/FA(50)
000002        COMMON/I1/K,IC1,INDX(10000)
000002        COMMON/I2/NN,IR,NC,IC(10),JC(10)
000002        COMMON/I3/INA1,INA2,INA(50)
000002        DIMENSION G(51)
000002        K=10000
000003        M=50
000004        PRINT 2
000010   2    FORMAT(*1*)
000010        DO 1 I=3,M
000012        CALL PRNT1(I)
000013        CALL INDEX(I)
000015   1    CONTINUE
000020        CALL FLUSH
000021        END

PROGRAM LENGTH INCLUDING I/O BUFFERS
004243

UNUSED COMPILER SPACE
007100
```

```
              SUBROUTINE PRNT1(N)
000002        COMMON/Y/Y(50)
000002        COMMON/B1/FA(50)
000002        COMMON/I1/K,IC1,INDX(10000)
000002        COMMON/I2/NN,IR,NC,IC(10),JC(10)
000002        COMMON/I3/INA1,INA2,INA(50)
000002        IF(INA1.EQ.0)GOTO1
000003   5    INA2=INA2+1
000005        K1=INA(N-1).A.3R
000007   2    INA(N)=ISHFTL(K1,18)+K1
000014        RETURN
000015   1    K1=1
000016        INA1=N
000017        INA2=N
000020        GOTO2
000021        END

SUBPROGRAM LENGTH
000040

UNUSED COMPILER SPACE
007100
```

```
                 SUBROUTINE FLUSH
000001           COMMON/Y/Y(50)
000001           COMMON/B1/FA(50)
000001           COMMON/I1/K,IC,INDX(10000)
000001           COMMON/ID/IDAT(50,3)
000001           COMMON/I3/INA1,INA2,INA(50)
000001           IF(INA1.EQ.0)RETURN
000003           KL=10
000004           IRN=IRN+1
000006           IF(IC.LT.0)GOTO1
000007           ISW=0
000010    4      CONTINUE
000011           NW=INA(INA2).A.3R
000013           NW=NW-1
000014           WRITE(1)INA1,INA2,(INA(I),I=INA1,INA2),IC,NW,(INDX(I),I=1,NW)
000036           NLW=NLW+KL+INA2-INA1
000041           IF(NLW.LT.50)GOTO10
000043           NLW=KL+INA2-INA1
000044           WRITE(6,27)
000050   10      CONTINUE
000050           WRITE(6,21)IRN,NW
000060           IF(INA1.EQ.INA2)GOTO20
000062           WRITE(6,22) INA1,INA2
000072   25      CONTINUE
000072           WRITE(6,24)
000076           WRITE(6,23)((I,(IDAT(I,J),J=1,3)),I=INA1,INA2)
000120           DO 31 I=INA1,INA2
000125   31      INA(I)=0
000127           IF(ISW.NE.0)GOTO3
000130           IC=INA1=INA2=0
000133           RETURN
000133   20      CONTINUE
000133           WRITE(6,26) INA1
000141           GOTO25
000142    1      IF(INA1.EQ.INA2)GOTO2
000144           IC=0
000144           IT=INA2
000145           ISW=1
000146           INA2=INA2-1
000150           GOTO4
000150    3      INA1=INA2=IT
000153           IF(ISW.EQ.2)GOTO5
000155           K1=ISHFTRA(INA(IT),18)
000161           K2=INA(IT).A.3R
000163           K2=K2-K1
000163           INA(IT)=1000000B+K2+1
000165           K1=K1-1
000166           DO 15 I=1,K2
000175           INDX(I)=INDX(I+K1)
000176   15      CONTINUE
000177           IC=0
000200           RETURN
000200    2      ISW=2
000201           IT=INA2
000203           GOTO4
000203    5      CONTINUE
000203           INA(IT)=10000C1[illegible]
000205           [illegible]
```

```
000206          RETURN
000207    21    FORMAT(///*    RECORD NUMBER *,I3,5X,*INCEX LENGTH IS *,I6/)       C0059.C00
000207    22    FORMAT(*    N VALUES RANGE FROM *,I3,*    TO *,I3/)                 C0060.C00
000207    23    FORMAT(2X,I2,3X,I8,X,I8,3X,I2)                                       00061.000
000207    24    FORMAT(5X,*STATISTICS*/*    N     NO SETS  NO INDICES MAX*)          C0062.C00
000207    26    FORMAT(*    N VALUE IS *,I3/)                                         C0063.000
000207    27    FORMAT(*1*)                                                          C0064.000
000207          END                                                                  00065.000
                                                                                     C0066.C00
SUBPROGRAM LENGTH
000304

UNUSED COMPILER SPACE
006200



                SUBRCUTINE INCEX(N)                                                  C0001.000
000002          COMMON/I2/NN,IR,NO,IC(1C),JC(10)                                     C0002.000
000002          NN=N                                                                 C0003.000
000003          IC(1)=N                                                              C0004.000
000003          JC(1)=1                                                              C0005.000
000004          IR=1                                                                 00006.000
000005          NO=1                                                                 C0007.C00
000006          K=0                                                                  C0008.000
000007    10    CCNTINUE                                                             00009.000
000007          CALL PRNT                                                            00010.000
000010          IF(IC(1).EQ.1)RETURN                                                 C0011.000
000014    16    K=K+IC(NO)                                                           00012.000
000016          IR=IR-1                                                              00013.000
000017          JC(NO)=JC(NC)-1                                                      CC014.C00
000020          M=IC(NO)-1                                                           C0015.000
000022          IF(M.EQ.0)GCTC11                                                     00016.000
000023          IF(JC(NO).EC.C)NC=NO-1                                               00017.C00
000026    14    J=K/M                                                                C0018.000
000031          IF(J.EQ.0)GCTC15                                                     C0019.000
000033          NC=NC+1                                                              00020.000
000034          JC(NO)=J                                                             CC021.C00
000036          IC(NO)=M                                                             C0022.000
000037          K=K-M*J                                                              00023.000
000040          IR=IR+J                                                              CC024.C00
000041          IF(K.EQ.0)GCTC1C                                                     CC025.000
000042    15    CCNTINUE                                                             CC026.000
000042          NC=NC+1                                                              00027.000
000044          JC(NO)=1                                                             CC028.C00
000046          IC(NO)=K                                                             C0029.000
000047          IR=IR+1                                                              00030.000
000050          K=0                                                                  CC031.000
000051          GCTO10                                                               C0032.000
000051    11    K=K+JC(NO)                                                           CC033.000
000053          IR=IR-JC(NO)                                                         00034.000
000055          NO=NC-1                                                              C0035.C00
000056          GOTO16                                                               C0036.C00
000056          END                                                                  C0037.000

SUBPROGRAM LENGTH
000071

UNUSED COMPILER SPACE
007000
```

STORAGE ALLOCATION.

ADDRESS	LENGTH
0	73
73	

BINARY CONTROL CARDS.

```
IDENT PRNT
END
```

BLOCKS	TYPE	ADDRESS	LENGTH
PROGRAM*	LOCAL	0	73
I1	COMMON	0	23422
I2	COMMON	0	27
I3	COMMON	0	64
ID	COMMON	0	226

ENTRY POINTS.

PRNT - 1

EXTERNAL SYMBOLS.

FLUSH

```
                                       IDENT PRNT
                                       ENTRY PRNT
                               *
                               *       THIS ROLTINE PACKS  INDICES INTO BUFFER
                               *
                               *
                               *       THIS MACRO  PACKS ONE SET OF INDICES
                               *
                               PCK     MACRO A
                                       IFC   NE,*A*N*,2
                                       SA4   A4+B5
                                       SA5   A5+B5
                                       LX4   6
                                       BX4   X4+X5           PACK I AND J
                                       LX3   12
                                       BX3   X3+X4           MERGE TC PREVICLS PACKING
                                       SB1   B1-12
                                       IX1   X1-X7
                                       ZR    X1,PACKD
                                       ENDM
                               *
                               *
                               *
                               *
                               *
 0  20010313000000000000               VFD   42/CLPACK,18/0
 1                             PRNT    BSSZ  1
 2  5110000000 C               PACK1   SA1   N               GET N INTO B2
           63210                       SB2   X1
 3  5112000001 C                       SA1   B2+INA-1        FIND PCSITICN FCR
           10011                       BXC   X1              PLACING NEXT PACKEC SET OF INDICES
                63310                  SB3   X1              START IS IN B3
 4  5120000000 C                       SA2   K
           63420                       SB4   X2              LIMIT IS IN B4
                               *
                               *
                               *
 5  5110000002 C                       SA1   NC              GET NC CF SETS CF INDICES IN X1
           712C0CC004                  SX2   4
 6  37321                              IX3   X2-X1
      43601                            MX6   1
           11336                       BX3   X3*X6
                               *                             0 IF 4 CR LESS INCICES
                               *                             1 IF 5 CR MCRE SETS
 7  7170CCC001                         SX7   1
           5142777776 C                SA4   B2+IDA1-1
10  5152000061 C                       SA5   B2+IDA2-1
           36647                       IX6   X4+X7
                36751                  IX7   X5+X1
11  54640                              SA6   A4
      54750                            SA7   A5
           5142000143 C                SA4   B2+IDA3-1
12  37441                              IX4   X4-X1
      0324000014 +                     PL    X4,PACK11
                10611                  BX6   X1
13  54640                              SA6   A4
                               *
```

```
 14  5120000001 C        PACK11   SA2   IR
            20260                 LX2   48
                 12332            BX3   X3+X2
 15  20314                        LX3   12
                          *
                          *       INITIALIZE INDEX PACKING
                          *
       6110000060                 SB1   48
 16  7170000001                   SX7   1
            63570                 SB5   X7
                          *       GET FIRST SET OF INDICES
 17  5140000003 C                 SA4   I
            5150000015 C          SA5   J
                          *
 20  20406                        PCK   N              PACK FIRST SET OF INDICES
            54445                 PCK                  PACK ONE SET OF INDICES
            54445                 PCK                  PACK ONE SET OF INDICES
            54445                 PCK                  PACK ONE SET OF INDICES
                          *
                          *       STORE FIRST WORD
                          *
            10633                 BX6   X3
 34  0323000056 +                 PL    X3,PACK5       EXIT IF NO MORE WORDS TO PACK
            5163000001 C          SA6   B3+INDX-1
 35  36007                        IX0   X0+X7
       66335                      SB3   B3+B5
            0743000062 +          LT    B4,B3,PACK3    INDEX BUFFER OVERFLOW
                          *
                          *       START SECOND WORD
                          *
 36  13333                        BX3   X2-X3
       6110000074                 SB1   60
                          *
                          *
                 54445            PCK                  PACK ONE SET OF INDICES
            54445                 PCK                  PACK ONE SET OF INDICES
            54445                 PCK                  PACK ONE SET OF INDICES
            54445                 PCK                  PACK ONE SET OF INDICES
            54445                 PCK                  PACK ONE SET OF INDICES
                          *
                          *       STORE SECOND WORD
            10633                 BX6   X3
 56  5163000001 C        PACK5    SA6   B3+INDX-1
            36607                 IX6   X0+X7
 57  5162000001 C                 SA6   B2+INA-1       UPDATE INA TABLE
            66335                 SB3   B3+B5
 60  0743000065 +                 LT    B4,B3,PACK4    IF INDEX BUFFER IS FULL GO
            0400000001 +          EQ    PRNT           TO FLUSH, OTHERWISE RETURN
                          *
                          *       COME HERE IF WORD NOT FULL
                          *
 61  22613               PACK0    LX6   B1,X3          LEFT JUSTIFY
       0400000056 +               EQ    PACK5          FINISH PROCESSING AND EXIT
                          *
                          *
                          *
 62  43673               PACK3    MX6   59             SET CONTINUE FLAG
```

```
          5160000001 C              SA6   IC                                            00115.000
   63  01C0000000 X          +      RJ    =XFLUSH                                       00116.000
              0700000000 +   -      LT    BC,B0,PRNT-1                                  00117.000
   64  0400000002 +                 EQ    PACK1          START CVER                     00118.000
                             *                                                          00119.000
   65  5110000015 C          PACK4  SA1   J                                             00120.000
              76220                 SX2   B2                                            00121.000
                   37112            IX1   X1-X2                                         00122.000
   66  0301000072 +                 ZR    X1,PACK41      NC MORE INCICES FCR THIS N     00123.000
              43673                 MX6   59             SET CCNTINLE FLAG              00124.000
   67  5160CC0001 C          PACK42 SA6   IC                                            00125.000
   70  0100000000 X          +      RJ    =XFLUSH                                       00126.000
              0700000000 +   -      LT    B0,B0,PRNT-1                                  00127.000
   71  04000C0001 +                 EQ    PRNT                                          00128.000
   72  13666                 PACK41 BX6   X6-X6                                         00129.000
          0400000067 +              EQ    PACK42                                        00130.000
                                    USE   /I1/                                          00131.000
    0                        K      BSS   1                                             00132.000
    1                        IC     BSS   1                                             00133.000
    2                        INDX   BSS   100C0                                         00134.000
                                    USE   /I2/                                          00135.000
    0                        N      BSS   1                                             00136.000
    1                        IR     BSS   1                                             00137.000
    2                        ND     BSS   1                                             00138.000
    3                        I      BSS   10                                            00139.000
   15                        J      BSS   1C                                            00140.000
                                    USE   /I3/                                          00141.000
    0                               BSS   2                                             00142.000
    2                        INA    BSS   50                                            00143.000
                                    USE   /ID/                                          00144.000
    0                        IDA1   BSS   5C                                            00145.000
   62                        IDA2   BSS   50                                            00146.000
  144                        IDA3   BSS   5C                                            00147.000
                                    USE   *                                             00148.000
   73                               ENC                                                 00149.000
                   43204     STORAGE USEC             238 STATEMENTS      22 SYMBOLS
                             6000 SERIES ASSEMBLY   1.392 SECONDS         61 REFERENCES
```

SYMBOLIC REFERENCE TABLE.

FLUSH	0	EXTERNAL*	4/02	4/12				
I	3	I2	3/12	4/25 L				
IC	1	I1	4/01 S	4/11 S	4/19 L			
IDA1	0	IC	2/46	4/31 L				
IDA2	62	IC	2/47	4/32 L				
IDA3	144	IC	2/52	4/33 L				
INA	2	I3	2/30	3/45 S	4/29 L			
INDX	2	I1	3/24 S	3/43 S	4/2C L			
IR	1	I2	3/01	4/23 L				
J	15	I2	3/13	4/06	4/26 L			
K	0	I1	2/33	4/18 L				
N	0	I2	2/28	4/22 L				
NO	2	I2	2/38	4/24 L				
PACKD	61	PROGRAM*	3/16	3/18	3/36	3/38	3/4C	
			3/17	3/19	3/37	3/39	3/52 L	
PACK1	2	PROGRAM*	2/28 L	4/04				
PACK11	14	PROGRAM*	2/54	3/01 L				
PACK3	62	PROGRAM*	3/27	3/57 L				
PACK4	65	PROGRAM*	3/47	4/06 L				
PACK41	72	PROGRAM*	4/09	4/15 L				
PACK42	67	PROGRAM*	4/11 L	4/16				
PACK5	56	PROGRAM*	3/23	3/43 L	3/53			
PRNT	1	PROGRAM*	2/02 E	2/27 L	3/48	4/03	4/13	4/14

```
                PROGRAM BUSY(INPUT,OUTPUT,TAPE5=INPUT,TAPE1,TAPE2)
000002          COMMON/Y/Y(50)
000002          COMMON/B1/FA(50)
000002          COMMON/I1/K,IC1,INDX(10000)
000002          COMMON/I2/NN,IR,NO,IC(10),JC(10)
000002          COMMON/I3/INA1,INA2,INA(50)
000002          DIMENSION G1(50,4)
000002          DIMENSION G(50,4)
000002          DIMENSION A(50,4),B(50,4)
000002          DIMENSION AL(50),ET(50)
000002          DIMENSION YL(4),RHC(4)
000002          DIMENSION FAN(50)
          C
000002          M1=0
          C
000003    21    CONTINUE
000003          REWIND 1
000005          READ 20,M,JO,YL
000017    20     FORMAT(I2,X,I1,X4(X,F10.0))
000017          IF(EOF,5)22,23
000022    22    STOP
000024    23    CONTINUE
000024          PRINT 60
          C
          C
000030          IF(M.LE.M1)GOTO35
000033          M1=M
          C
000033           CALL ALPHA(AL,M)
000035           CALL ETA   (ET,M)
          C
000037          FA(1)=1.
000040          FAN(1)=1.
000042          DO 1 I=2,M
000050          FA(I  )=FA(I-1)*I
000052          FAN(I)=FAN(I-1)*(1.-I)
000054    1     CONTINUE
          C
          C
000055          AN1=AL(1)
000056          IF(AL(1).EQ.1.)GOTO30
000057          AN=AN1=AL(1)
000061          DO 31 I=1,M
000067          AL(I)=AL(I)/AN
000070          AN=AN*AL(1)
000071    31    CONTINUE
000072          PRINT 67
000076    30    CONTINUE
          C
          C   INITIALIZE INDEX
          C
000076          MM1=M
000077          IF(M.GT.20)MM1=20
000103          CALL GETINDX(3,MM1)
          C
          C
          C
000105    35     DO 36 J=1,JO
```

```
000114          YL(J)=YL(J)*AN1
000115          RHC(J)=YL(J)*ET(1)
000116    36    CCNTINUE
000117          PRINT 66,(YL(J),J=1,JO)
000125          PRINT 68,(RHO(J),J=1,JO)
          C
          C    INITIALIZATICN
          C
000134          DC 40 J=1,JO
000136          A(1,J)=1./(1.-YL(J)*ET(1)*AL(1))
000144          G(1,J)=G1(1,J)=B(1,J)=AL(1)*A(1,J)
000153    40    CONTINUE
          C
          C   N = 2 SETUP
          C
000155          DC 41 J=1,JO
000171          Y3=G(1,J)**2
000172          Y2=B(1,J)**2
000173          Y1=A(1,J)**2
000174           G(2,J)=A(1,J)*(AL(2)*Y1+AL(1)*YL(J)*(ET(2)*Y2-ET(1)*Y3))
000204          B(2,J)=(G(2,J)-Y3)/2
000206          A(2,J)=YL(J)*(ET(2)*Y2+ET(1)*2.*B(2,J))/2.
000212          G1(2,J)=G(2,J)/2.
000213    41    CCNTINUE
000216          PRINT 64
000221          DO 46 I=1,2
000223          PRINT 65,I,AL(I),ET(I),(G(I,J),J=1,JC)
000245    46    CCNTINUE
          C
          C
000247          N=2
000250    100   CONTINUE
000250          N=N+1
000252          DC 45 J=1,JC
          C
000253          CALL FORMY(N,G1(1,J))
000257          S3=SUM(FAN,N)
          C
          C
000262          CALL FORMY(N,B(1,J))
000266          S2=SUM(ET ,N)
          C
          C
000271          CALL FORMY(N,A(1,J))
000275          S1=SUM(AL ,N)
          C
          C
          C
000305          G(N,J)=G(1,J)*(S1+YL(J)*AL(1)*(S2+ET(1)*S3))/AL(1)
000314          B(N,J)=(S3+G(N,J))/FA(N)
000316          A(N,J)=YL(J)*(S2/FA(N)+ET(1)*B(N,J))
000322          G1(N,J)=G(N,J)/FA(N)
000324    45    CCNTINUE
          C
          C
000327          PRINT 65,N,AL(N),ET(N),(G(N,J),J=1,JO)
000351          IF(N.LT.M)GCTC1CC
          C
```

```
              C
000354   60    FORMAT(*1  THE MCMENTS CF THE BUSY PERICD OF*
              1* A SINGLE SERVER QUEUE*/
              1*   WITH GRCUF ARRIVALS AND A POISSON ARRIVAL PROCESS*///)
000354   62    FORMAT(8(X,E15.6,X))
000354   64    FCRMAT(///12X,*SERVICE TIME*,8X,*BUNCH SIZE*,
              14(7X,*BUSY PERICC*)/
              2 5X,*N*,6(9X,*MCMENTS  *))
000354   65    FORMAT(3X,I3,7E18.7)
000354   66    FORMAT(*   THE GROUP ARRIVAL RATE LAMBDA =*8X,4F18.8)
000354   68    FORMAT(*   THE TRAFFIC INTENSITY RHO =*12X,4F18.8)
000354   67    FORMAT(*    THE SERVICE TIMES HAVE BEEN NCRMALIZEC*)
              C
              C
              C
              C
000354         GCTO21
000354         END

PROGRAM LENGTH INCLUDING I/O BUFFERS
006561

UNUSED COMPILER SPACE
005300
```

```
                 SUBROUTINE GETINDX(M2,M)
000004           COMMON/I1/K,IC1,INDX(10000)
000004           COMMON/I3/INA1,INA2,INA(50)
000004           M1=M2
000004           IE=0
000005           IF(INA1.EQ.0)GOTO4
000007           IF(M1.LT.INA1)GOTO11
000010           IF(M1.GT.INA2)GOTO12
000013           RETURN
000013    11     REWIND 1
000015    12     CONTINUE
000015           DO 3 I=INA1,INA2
000022    3      INA(I)=0
000024    4      CONTINUE
000024           READ(1)INA1,INA2,(INA(I),I=INA1,INA2),IC1,NW,(INDX(I),I=1,NW)
000046           IF(EOF,1)14,13
000052    13     IF(M1.GT.INA2)GOTO4
000056           IF(M1.LT.INA1)GOTO15
000057           IF(IC1.EQ.0)RETURN
000061           REWIND 2
000063           WRITE(2)INA1,INA2,(INA(I),I=INA1,INA2),IC1,NW,(INDX(I),I=1,NW)
000105           RETURN
000106           ENTRY GETINDB
000115           REWIND 2
000117    10     CONTINUE
000117           READ(2)INA1,INA2,(INA(I),I=INA1,INA2),IC1,NW,(INDX(I),I=1,NW)
000141           IF(IC1.LT.0)RETURN
000145           IC1=1
000146           RETURN
000147           ENTRY GETINDC
000156           IF(IE.NE.0)GOTO10
000157           READ(1)INA1,INA2,(INA(I),I=INA1,INA2),IC1,NW,(INDX(I),I=1,NW)
000201           WRITE(2)INA1,INA2,(INA(I),I=INA1,INA2),IC1,NW,(INDX(I),I=1,NW)
000225           IF(IC1.LT.0)RETURN
000231           IE=1
000232           IC1=1
000233           RETURN
000233    14     PRINT 16
000241    15     PRINT 17,M1
000247           STOP
000252    16     FORMAT(*  END OF TAPE ENCOUNTERED*)
000252    17     FORMAT(*   INDEX VALUES FOR *,I3,*  WERE NOT FOUND*)
000252           END

SUBPROGRAM LENGTH
000312

UNUSED COMPILER SPACE
006300
```

```
                 FUNCTION SUM(A,N)                                   CC135.C00
000004           COMMON/Y/Y(30)                                      C0136.C00
000004           DIMENSION A(N)                                      00137.000
000004           SUM=0                                               00138.000
000005           DO 1 I=2,N                                          CC139.000
000012           SUM=SUM+A(I)*Y(I)                                   C0140.000
000014    1      CONTINUE                                            00141.000
000015           RETURN                                              CC142.C00
000016            END                                                CC143.C00

SUBPROGRAM LENGTH
000030

UNUSED COMPILER SPACE
007200
```

```
                 SUBROUTINE  ALPHA(A,M)                              00001.000
000004           DIMENSION A(M)                                      CC002.C00
000004           A(1)=1.                                             CC003.C00
000005           DO 2 I=2,M                                          C0004.000
000011           A(I)=A(I-1)*I                                       CC005.C00
000013    2      CONTINUE                                            C0006.000
000014           RETURN                                              00007.000
000015           END                                                 00008.000

SUBPROGRAM LENGTH
000027

UNUSED COMPILER SPACE
007200
```

```
                 SUBROUTINE ETA(A,M)                                 CC009.C00
000004           DIMENSION A(M)                                      CC010.C00
000004           DO 1 I=1,M                                          C0011.C00
000011    1       A(I)=1.                                            CC012.C00
000012           RETURN                                              CC013.C00
000013           END                                                 C0014.000

SUBPROGRAM LENGTH
000026

UNUSED COMPILER SPACE
007200
```

```
                   SUBRCUTINE GAMMA(A,N,AL,B)                                C0001.000
000006             DIMENSICN A(N)                                            CC002.C00
000006             A(1)=AL/B                                                 CC003.000
00C007             DC 1 I=2,N                                                C0004.000
000013             A(I)=A(I-1)*(AL+I-1)/B                                    00005.000
000020      1      CCNTINUE                                                  CC006.C00
00C021             RETURN                                                    C0007.000
0C0021             END                                                       C0008.000

SUBPROGRAM LENGTH
000034

UNUSED COMPILER SPACE
007100
```

```
                   SUBRCUTINE NEGBIN(A,N,P,V)                                C0001.000
00C006             DIMENSICN A(N)                                            CC002.000
00C007             P1=(1-P)/P                                                C0003.C00
00C011             A(1)=V*P1                                                 00004.000
000012             A(2)=P1*((V+1)*A(1)+V)                                    C0005.C00
00C016             DC 1 I=3,N                                                C0006.000
000017             J2=I-2                                                    00007.000
000020             S=0                                                       00008.C00
000021             C1=1.                                                     C0009.C00
0C0023             DC 2 J=1,J2                                               C0010.000
000027             C1=(C1*(I-J))/J                                           00011.000
00C033             S=S+C1*(V*A(I-J-1)+A(I-J))                                CC012.C00
000042      2      CCNTINUE                                                  C0013.000
0C0043             A(I)=P1*(V*A(I-1)+S+V*A(1))                               00014.000
000054      1      CCNTINUE                                                  CC015.C00
000056             RETURN                                                    CC016.C00
000057             END                                                       C0017.000

SUBPROGRAM LENGTH
000113

UNUSED COMPILER SPACE
0C7000
```

```
                 SUBRCUTINE POIS(A,N,YL)
000005           DIMENSICN A(N)
000005           A(1)=YL
000006           DC 1 I=2,N
000010           A(I)=1.
000011           C=1.
000011           I1=I-1
000013           DC 2 J=1,I1
000021           C=(C*(I-J))/J
000025           A(I)=A(I)+C*A(J)
000027    2      CONTINUE
000030           A(I)=A(I)*YL
000034    1      CONTINUE
000036           RETURN
000036           END

SUBPROGRAM LENGTH
000067

UNUSED COMPILER SPACE
007100
```

```
                 SUBROUTINE GEC(A,N,P)
000005           DIMENSICN A(N)
000005           A(1)=1./P
000006           Q=1.-P
000007           DO 1 I=2,N
000011           J1=I-1
000012           C1=1.
000013           S=0
000015           DO 2 J=1,J1
000021           C1=(C1*(I-J))/J
000025           S=S+C1*A(I-J)
000030    2      CCNTINUE
000031           A(I)=(A(I)+C*S)/P
000037    1      CCNTINUE
000041           RETURN
000041           END

SUBPROGRAM LENGTH
000070

UNUSED COMPILER SPACE
007000
```

```
STORAGE ALLOCATION.                      BINARY CONTROL CARDS.

      ADDRESS   LENGTH                   IDENT FORMY
          0       157                   END
        157

                         BLOCKS      TYPE      ADDRESS    LENGTH

                         PROGRAM*   LOCAL          0        156
                         LITERALS*  LOCAL        156          1
                         I1         COMMON         0      23422
                         I3         COMMON         0         64
                         Y          COMMON         0         62
                         B1         COMMON         0         62

                         ENTRY POINTS.

                         FORMY    -      1

                         EXTERNAL SYMBOLS.

                         GETINDX   GETINDC   GETINDB
```

```
                                                ICENT FCRMY
                                                ENTRY FCRMY
                                         *
                                         *      THIS MACRO PROCESSES ONE SET CF INDICES
                                         *
                                         PROC   MACRO
                                                LOCAL  F51,F52,F53
                                                LX1    6
                                                BX2    -X0*X1          GET I
                                                ZR     X2,F4        DCNE IF I = 0
                                                SA4    B7+X2
                                                LX1    6            GET J(I)
                                         *
                                         *      BEGIN PRODUCT FORMATION
                                                BX2    -X0*X1
                                         *
                                                SB5    X2
                                         *
                                         *      BEGIN POWER ALGORITHM
                                         *
                                                SA5    =1.C
                                         F52    LX2    59
                                                PL     X2,F51
                                                FX5    X5*X4
                                                BX2    -X0*X2
                                                ZR     X2,F53
                                         F51    BX6    X4
                                                FX4    X6*X4
                                                EQ     F52
                                         *
                                         *
                                         F53    SA4    B5+FA-1         POWER FORMED IN X5
                                                FX5    X5/X4
                                                FX7    X7+X5
                                                ENCM
                                         *
0  0617221531000C000002                         VFC    42/CLFORMY,18/2
1                                        FORMY  BSS    1
2  56110                                        SA1    B1
        63110                                   SB1    X1
              6172777776                        SB7    B2-1
3  54210                                        SA2    A1
                                         *
                                         *      (B7) = G-ARRAY ACCRESS
                                         *      (B1) = N
                                         *
                                         *
                                         *      CLEAR Y
                                         *
        6120000062                              SB2    50D
4  71600C0000                                   SX6    0
5  5162777776 C                          F3     SA6    B2+Y-1
              6122777776                        SB2    B2-1
6  0520000005 +                                 NE     B2,B0,F3
                                         *
                                         *
                                         *
```

FORMY

```
  7  5111000001 C          F2    SA1   B1+INA-1      CHECK FOR PRESENCE OF INDEX
             0301000131 +        ZR    X1,F1         GO GET INDEX
 10  5130000001 C                SA3   IC
             0303000015 +        ZR    X3,F21
 11  0333000015 +                NG    X3,F21        CONTINUATION OF PREVIOUS BLOCK
 12  01300000000070000136 +      SAVREG REG
 13  0100000000 X           +    RJ    =XGETINDB
             07020C0000 +   -    LT    BC,B2,FORMY-1
 14  013000000001C0000136 +      RESREG REG
 15  5111000001 C          F21   SA1   B1+INA-1      GET NEW LIMITS
          63310                  SB3   X1             (B2) = START OF INDICES
               21122             AX1   18            (B3) = END OF INDICES
 16  63210                       SB2   X1            IF  (B3).GT.K  CONTINUE
                            .
                            .    START FORMING Y(N,R)
                            .
 17  5111777776 C          F5    SA1   B1+FA-1
          10711                  BX7   X1
 20  5112C00001 C                SA1   B2+INDX-1
          612200C001             SB2   B2+1
 21  43066                       MX0   54
       10311                     BX3   X1
            20114                LX1   12
                 15210           BX2   -X0*X1
 22  63420                       SB4   X2            SAVE R IN B4
       20106                     PROD
 31  20106                       PROD
 40  20106                       PROD
 47  20106                       PROD
 56  0323000122 +                PL    X3,F4         FINISHED IF POSITIVE
          51120C0001 C           SA1   B2+INDX-1     GET SECOND PART
 57  6122000001                  SB2   B2+1
          20106                  PROD
 66  20106                       PROD
 75  20106                       PROD
104  20106                       PROD
113  20106                       PROD
                            .
                            .
                            .
122  5114777776 C          F4    SA1   B4+Y-1        ADD PART FORMED TO Y(N,R)
          30717                  FX7   X1+X7
               54710             SA7   A1
                            .
                            .
                            .
123  07230C0017 +                LT    B2,B3,F5
          511C0CC001 C           SA1   IC
124  0321000001 +                PL    X1,FORMY      POSITIVE MEANS ALL DONE .
125  01300000000070C0C136 +      SAVREG  REG
126  01C0000000 X           +    RJ    =XGETINDC
          07020C0000 +      -    LT    BO,B2,FORMY-1
127  01300000C001CC000136 +      RESREG REG
130  0400C00015 +                EQ    F21
                            .
                            .
                            .
```

```
 131  0130000000070000136 +  F1        SAVREG REG                                       00115.000
 132  64120                            SB1    A2                                        00116.000
          66210                        SB2    B1                                        00117.000
 133  01C0000000 X            +        RJ     =XGETINCX                                 00118.000
                070200C000 +  -        LT     B0,B2,FCRMY-1                             00119.000
 134  0130000000100000136 +            RESREG REG                                       00120.000
 135  040C000007 +                     EC     F2                                        00121.000
 136                          REG      BSS    16D                                       00122.000
                                       USE    /I1/                                      00123.000
   0                          K        BSS    1                                         00124.0C0
   1                          IC       BSSZ   1                                         00125.000
   2                          INDX     BSSZ   1C0C0                                     00126.000
                                       USE    /I3/                                      00127.0C0
   0                                   BSS    2                                         00128.000
   2                          INA      BSS    50                                        00129.0C0
                                       USE    /Y/                                       00130.000
   0                          Y        BSS    5CD                                       00131.0C0
                                       USE    /B1/                                      00132.000
   0                          FA       BSS    5CD                                       00133.000
 157                                   END                                              00134.0C0
                    43204     STORAGE USED            329 STATEMENTS    44 SYMBOLS    000027 INVENTED SYMBOLS
                              6000 SERIES ASSEMBLY  1.881 SECONDS       63 REFERENCES
```


SYMBOLIC REFERENCE TABLE.

```
FA          0    B1          3/17     3/28     3/3C     3/35     3/37     4/19 L
                             3/27     3/29     3/34     3/36     3/38
FORMY       1    PROGRAM*    2/02 E   2/38 L   3/08     3/49     3/52     4/05
F1        131    PROGRAM*    3/02     4/01 L
F2          7    PROGRAM*    3/01 L   4/07
F21        15    PROGRAM*    3/04     3/05     3/1C L   3/54
F3          5    PROGRAM*    2/52 L   2/54
F4        122    PROGRAM*    3/27     3/29     3/3C     3/35     3/37     3/41 L
                             3/28     3/3C     3/34     3/36     3/38
F5         17    PROGRAM*    3/17 L   3/47
GETINDB     0    EXTERNAL*   3/07
GETINDC     0    EXTERNAL*   3/51
GETINDX     0    EXTERNAL*   4/04
IC          1    I1          3/03     3/48     4/11 L
INA         2    I3          3/01     3/10     4/15 L
INDX        2    I1          3/19     3/31     4/12 L
K           0    I1          4/10 L
REG       136    PROGRAM*    3/06     3/09     3/5C     3/53     4/C1     4/06     4/08 L
Y           0    Y           2/52 S   3/41     4/17 L
```

```
            PRCGRAM MCMENTS(INPUT,CUTPUT)
000002      DIMENSICN G(100),B(101)
000002      R=C.0
000003      NNN=50
000004      DO 10 K=1,19
000006      R=R+0.05
000010      G(1)=1./(1.-R)
000011      X1=(1.-R)/R
000013      R1=SCRT(R)
000016      X2=1./(1.+R1)**2
000020      X3=1./(1.-R1)**2
000022      B(1)=1./(1.+R1)**4
000024      B(2)=1./(1.-R)**2
000027      B(3)=1./(1.-R1)**4
000032      B(1)=(X1*B(1))/8.
000034      B(2)=(X1*B(2))/4.
000036      B(3)=(X1*B(3))/8.
000040      G(2)=B(1)-B(2)+B(3)
000042      N=2
000043    2 N=N+1
000045      B(1)=B(1)*X2*(N-1.5)
000051      B(N+1)=B(N)*X3*(N-1.5)
000055      N1=N-1
000056      DO 3 NU=2,N1
000067      B(NU)=B(NU)*X2*N*(1.-1.5/(N-NU+1.))
000074    3 CONTINUE
000076      B(N)=B(N)*X2*N*.5
000101      Y1=B(1)+B(N+1)
000103      DO 4 NU=2,N
000111      Y1=Y1-B(NU)
000112    4 CCNTINUE
000113      G(N)=Y1
000115      IF(N.LT.NNN)GCTC 2
000117      PRINT 1000
000123      PRINT 1001,R
000131      PRINT 1003,(J,G(J),J=1,NNN)
000145   10 CONTINUE
000147 1000 FCRMAT('1')
000147 1003 FCRMAT(2X,I4,E18.7,2X,I4,E18.7)
000147 1001 FORMAT(//2X'THE TRAFFIC INTENSITY RHC ECUALS ',F8.5///)
000147      END

PROGRAM LENGTH INCLUDING I/O BUFFERS
002642

UNUSED COMPILER SPACF
006400
```

```
              PROGRAM MOMENTS(INPUT,OUTPUT)
000002        DIMENSION B(61),C(60,60),GG(60)
000002        DIMENSION A(51),C(51,51)
000002        NNN=50
000003        DO 10 KK1=2,9
000005        P=KK1*.1
000006        KK2=KK1-1
000011        DO 11 KK3=1,KK2
000012        XLAM=KK3*.1
000014        Q=1.-P
000015        U1=Q/(XLAM+Q)
000020        U2=XLAM+Q
000021        UL=Q/U2
000022        U2=XLAM/U2
000023        R=XLAM/P
000023        X1=(1.-R)/R
000025        R1=SQRT(R)
000030        X2=1./(1.+R1)**2
000032        G(1)=U2/(P-XLAM)
000035        X3=1./(1.-R1)**2
000040        B(1)=1./(1.+R1)**4
000042        B(2)=1./(1.-R)**2
000044        B(3)=1./(1.-R1)**4
000050        B(1)=(X1*B(1))/8.
000052        B(2)=(X1*B(2))/4.
000054        B(3)=(X1*B(3))/8.
000056        G(2)=B(1)-B(2)+B(3)
000060        B(1)=B(1)/P**2
000062        B(2)=B(2)/P**2
000063        B(3)=B(3)/P**2
000064        G(2)=G(2)/P**2
000065        G(2)=G(2)*U2
000066        N=2
000067      2 N=N+1
000071        B(1)=B(1)*X2*(N-1.5)
000075        B(1)=B(1)/P
000076        B(N+1)=B(N)*X3*(N-1.5)
000102        B(N+1)=B(N+1)/P
000104        N1=N-1
000106        DO 3 NU=2,N1
000121        B(NU)=B(NU)*X2*N*(1.-1.5/(N-NU+1.))
000127      3 B(NU)=B(NU)/P
000132        B(N)=B(N)*X2*N*.5
000135        B(N)=B(N)/P
000136        Y1=B(1)+B(N+1)
000140        DO 4 NU=2,N
000145        Y1=Y1-B(NU)
000147      4 CONTINUE
000150        G(N)=Y1*U2
000153        IF(N.LT.NNN)GOTO 2
000155        A(1)=U1
000156        DO 12 K=2,NNN
000164        A(K)=A(K-1)*U1*K
000166     12 CONTINUE
000167        DO 13 J=1,NNN
000177        C(J,J)=1.
000177     13 CONTINUE
000201        DO 14 J=2,NNN
```

```
000202          J1=J-1
000203          DC 14 K=1,J1
000207          C(J,J-K)=C(J,J-K+1)*U1*(J-K+1)
000221       14 CCNTINUE
000224          DC 15 J=2,NNN
000225          GG(J)=A(J)
000227          DC 151 K=1,J
000237          GG(J)=GG(J)+G(K)*C(J,K)
000241      151 CCNTINUE
000242       15 CCNTINUE
000245          GG(1)=1./(1.-R)
000247          PRINT 1000
000253          PRINT 1001,XLAM,F,R
000265          PRINT 1003,(J,GG(J),J=1,NNN)
000301       11 CCNTINUE
000304       10 CCNTINUE
000306     1000 FORMAT('1')
000306     1003 FORMAT(2X,I4,E18.7,2X,I4,E18.7)
000306     1001 FCRMAT(/'  LAMBDA = ',F7.5,'    P =',F8.5,
              *'    RHO =',F8.5//)
000306          END

PROGRAM LENGTH INCLUDING I/O BUFFERS
010137

UNUSED COMPILER SPACE
005700
```