LA-5980

UC-20
Reporting Date: May 1975
Issued: August 1975

XPECT—A Monte Carlo Program to Predict the Expected-Time-to-Next-Failure in Controlled Thermonuclear Research Systems

by

G. P. Boicourt

los alamos
scientific laboratory
of the University of California
LOS ALAMOS, NEW MEXICO 87545

UNITED STATES
ENERGY RESEARCH AND DEVELOPMENT ADMINISTRATION
CONTRACT W-7405-ENG. 36

Printed in the United States of America. Available from
National Technical Information Service
U S Department of Commerce
5285 Port Royal Road
Springfield, VA 22151
Price: Printed Copy $4.00 Microfiche $2.25

XPECT—A MONTE CARLO PROGRAM TO PREDICT THE EXPECTED-TIME-TO-NEXT-FAILURE IN CONTROLLED THERMONUCLEAR RESEARCH SYSTEMS

by

G. P. Boicourt

ABSTRACT

The ability to predict failure rates is of increasing importance in controlled thermonuclear research (CTR) engineering as the systems increase in size. If a large CTR system is assembled without an examination of failure rates, its usefulness may be limited by insufficient time between failures. The usual mean-time-between-failure calculation does not apply here. Instead, an analogous quantity, the expected-time-to-next-failure, is defined and a Monte Carlo program (XPECT) is given for its computation. The computation takes advantage of the fact that failures in present CTR systems occur predominantly in developmental components being used in large quantities.

I. INTRODUCTION

The ability to predict failure rates is of increasing importance in controlled thermonuclear research (CTR) engineering as the systems increase in size. A large theta-pinch system may contain thousands of identical components, many of which are hardly beyond the development stage and whose failure rates may be fairly high. If such a system is assembled without due regard for these failure rates, it quite possibly will not operate satisfactorily.

The usual mean-time-between-failure calculation does not apply here because it assumes each component to be on the flat part of its failure-rate curve. This means that early-failure components have been eliminated before the system is assembled. Unfortunately, because of time and expense, some critical components in a large CTR system may not have been tested sufficiently to reach the flat part of the failure-rate curve. Usually only a few types of components determine the failure rate in CTR systems, and this makes possible the Monte Carlo calculation of an analogous quantity, the expected-time-to-next-failure, provided the failure distributions of the critical component types are known.

II. THEORETICAL PRELIMINARIES

We consider a probabilistic series system, that is, one in which the failure of any component causes the system to fail. The failure rate r(t) of any system is given by

$$r(t) = -\frac{1}{R_s}\frac{dR_s}{dt} \quad , \tag{1}$$

where R_s is the system reliability function. For series systems,

$$R_s = \prod_{i=1}^{N} R_i \quad , \tag{2}$$

where N is the total number of components and R_i is the reliability function of the i^{th} component. R_i is defined to be

$$R_i = \int_t^{\infty} f_i(t')dt' \quad . \tag{3}$$

Here $f_i(t)$ is the failure probability density of the i^{th} component.

From (3),

$$\frac{dR_i}{dt} = -f_i(t) \quad . \tag{4}$$

Also,

$$\frac{dR_s}{dt} = \sum_{i=1}^{N}\left(\prod_{\substack{j\neq i\\ j=1}}^{N} R_j\right)\frac{dR_i}{dt}$$

$$= -\sum_{i=1}^{N}\left(\prod_{\substack{j\neq i\\ j=1}}^{N} R_j\right)f_i(t) \quad .$$

Hence,

$$-\frac{1}{R_s}\frac{dR_s}{dt} = \frac{1}{\prod_{j=1}^{N} R_j} \cdot \left(\sum_{i=1}^{N}\left(\prod_{\substack{j\neq i\\ j=1}}^{N} R_j\right) f_i(t)\right)$$

$$= \sum_{i=1}^{N} \frac{f_i(t)}{R_i} \quad ,$$

so, from (1),

$$r(t) = \sum_{i=1}^{N} \frac{f_i(t)}{R_i} \quad . \tag{5}$$

In analogy with the mean-time-to-next failure, defined to be the reciprocal of the constant failure rate of an exponential distribution, we define the expected-time-to-next-failure by

$$\mathrm{ETNF}(t) = \frac{1}{r(t)} \tag{6}$$

Using the equivalent notation $R_j(t) = 1 - F_j(t)$,

$$\mathrm{ETNF}(t) = 1 \Bigg/ \sum_{j=1}^{N} \frac{f_j(t)}{1 - F_j(t)} \quad . \tag{7}$$

$F_j(t)$ is the unreliability of the j^{th} component defined by

$$F_j(t) = \int_0^t f_j(t')dt' \quad ;$$

it is the probability that the j^{th} component has failed at some time equal to or less than t.

Thus, the problem of calculating the expected-time-to-next-failure involves merely the mechanics of evaluating the series in Eq. (7) at each time point desired. If the system consists of thousands of dissimilar components, this evaluation would be very time-consuming or even impossible. However, only a few types of critical components are found in CTR experiments, and evaluation of the sum in Eq. (7) is considerably easier because one evaluation of f_j and F_j at each time step suffices to evaluate the contribution of all type-j components that have survived from the initial time point. The required computations are detailed after the following notation.

Let

J = number of component types

$N_k(t)$ = number of original units of the k^{th} type at time t

$M_k(t)$ = number of replacement units of the k^{th} type at time t

f_{ok} = probability density associated with all remaining original units of the k^{th} type

f_{ik} = probability density associated with the i^{th} individual replacement unit of the k^{th} type

F_{ok} = unreliability of any remaining original unit of the k^{th} type

F_{ik} = unreliability of the i^{th} individual replacement unit of the k^{th} type

p_{ok} = a posteriori failure probability of any original individual unit of the k^{th} type

p_{ik} = a posteriori failure probability of the i^{th} replacement individual unit of the k^{th} type

t_{ik} = time at which the i^{th} individual unit of the k^{th} type began operation

t = time of operation of the system.

A constant total number of operating units is assumed and is given by

$$N_o = \sum_{k=1}^{J} \left(N_k(t) + M_k(t) \right) \quad .$$

This implies that each sum. $N_k + M_k$. is a constant: thus. when an original unit fails, N_k is reduced by one and the number of replacements M_k is increased by one.

In the notation just defined. Eq. (7) can be written

$$ETNF(t) = 1 \Bigg/ \left(\sum_{k=1}^{J} \frac{N_k(t) f_{ok}(t)}{1 - F_{ok}(t)} + \sum_{k=1}^{J} \sum_{i=1}^{M_k(t)} \frac{f_{ik}(t - t_{ik})}{1 - F_{ik}(t - t_{ik})} \right) \tag{8}$$

Although the sum in Eq. (8) looks more complicated than that in Eq. (7), its computation is actually much simpler. Instead of computing $f_{ok}(t)$ and $F_{ok}(t)$ N_k times at point t, we need only compute these values once at time t. Moreover, if we use a constant time step Δt,

$$t_{ik} = t - n\Delta t \tag{9}$$

for some n. At any time n will be known, so if the values of

$$f_{ok}(n\Delta t) \Big/ \left(1 - F_{ok}(n\Delta t)\right)$$

are saved, much computation can be avoided. Computation of this ratio is quite time-consuming for certain types of statistics, so this storage strategy can save large amounts of computer time. The required computations of f_{ok} and F_{ok} will be treated later under the individual type of statistics.

The next concern is the computation of $N_k(t)$ and $M_k(t)$. A short time step Δt is chosen so that no more than one component is likely to fail during the interval $(t, t + \Delta t)$. For pulsed CTR systems, this interval could be a single shot. Then we calculate the probability that a failure will occur in the interval $(t, t + \Delta t)$, assuming all components to be working at time t. This probability is found as follows. The probability that a given unit of type k did not fail is $q_{ok} = 1 - p_{ok}$ if the unit is an original unit, or $q_{ik} = 1 - p_{ik}$ if the unit is a replacement. $p_{ok}(t)$ is the a posteriori failure probability for an original type-k unit in the time interval $(t, t + \Delta t)$, and is given by

$$p_{ok}(t) = \frac{\int_t^{t+\Delta t} f_{ok}(t')dt'}{1 - F_{ok}(t)} \quad .$$

The $p_{ik}(t)$ represent the a posteriori failure probabilities for the replacement units and can be obtained by use of Eq. (9) from the stored $p_{ok}(t)$ for earlier times.

The probability that the entire system worked is

$$Q_s = \prod_{k=1}^{J} \left(\prod_{j=1}^{N_k} (1 - p_{ok}) \cdot \prod_{i=1}^{M_k} (1 - p_{ik}) \right) \quad ,$$

so the probability that a failure occurred is

$$P(t) = \min \left\{ \left[1 - \prod_{k=1}^{J} \left(\prod_{j=1}^{N_k} (1 - p_{ok}) \cdot \prod_{i=1}^{M_k} (1 - p_{ik}) \right) \right], 1 \right\} \quad . \tag{10}$$

Given the probability of failure during the time step Δt, one can use Monte Carlo methods to decide if a failure occurred. A random number between 0 and 1 is selected and compared to P(t); if it is greater than P(t), no failure occurred and the calculation proceeds to compute ETNF and print, if desired. If P(t) is greater than or equal to the random number, the program must branch to a computation to find the failed unit and to replace it. Of course, if P(t) equals one, then the system cannot operate and the computation should be terminated with a print of the failure probabilities.

Determination of the failed component should be made in a way that takes into account the contribution of each component to the total failure probability. If the product in Eq. (10) is expanded it can be written

$$P(t) = \sum_{i=1}^{N} p_i' \tag{11}$$

where the p_i' are of the form

$$\begin{aligned} p_i' &= p_i - 1/2\, p_i \sum_{j \neq i} p_j + 1/3\, p_i \sum_{\substack{j \neq i \\ k \neq i}} p_j p_k + \cdots \\ &= p_i \cdot A_i \quad . \end{aligned}$$

Thus the p_i' are proportional to the individual failure probabilities of the components. The factor A_i is independent of other contributions of the i^{th} component and represents the most natural way of assigning to an individual component the effects of multiple failures. In general, the A_i' are not equal, but if the assumption of equality is made, then the determination of the failed component can be made according to the normalized probabilities obtained by dividing each probability by the sum of the probabilities. Thus

$$p'_{ok} = p_{ok} \Big/ \sum_{k=1}^{J} \left(N_k p_{ok} + \sum_{i=1}^{M_k} p_{ik} \right) ,$$

$$p'_{ik} = p_{ik} \Big/ \sum_{k=1}^{J} \left(N_k p_{ok} + \sum_{i=1}^{M_k} p_{ik} \right) ,$$

and

$$1 = \sum_{k=1}^{J} \left(N_k p'_{ok} + \sum_{i=1}^{M_k} p'_{ik} \right) . \tag{12}$$

Use of Eq. (12) can also be justified by assuming that Δt is short enough that the probabilities of multiple failures are small compared to single failure probabilities. This amounts to taking A_i equal to 1. In CTR systems where Δt equals one shot. this is probably a good approximation. Usually when a single component fails in such systems the rest of the shot is aborted. The remaining components then either do not receive the full stress of the shot or get an overstress during the abort—it is impossible to foretell which will happen on a given shot, but over a long period the average effect should be equivalent to the assignment of a shot to the remaining components.

To find the type that failed. a random number is picked and the sum in Eq. (12) is built up until it equals or exceeds the number. The k value for which this occurs gives the type. Using the same random number, the procedure is then used on the term

$$N_k p'_{ok} + \sum_{i=1}^{M_k} p'_{ik}$$

to decide if an original unit or a replacement unit of type k failed. After the failure is found it is replaced by making the necessary changes in N_k, M_k, and t_{ik}.

Control is then returned to the point of origin and the computation is continued. A flow diagram for the computation is given in Fig. 1.

III. FAILURE DISTRIBUTIONS

Seven distributions are included in the program. They may not seem as familiar as some used in probability and statistics, but they are those most commonly obeyed by

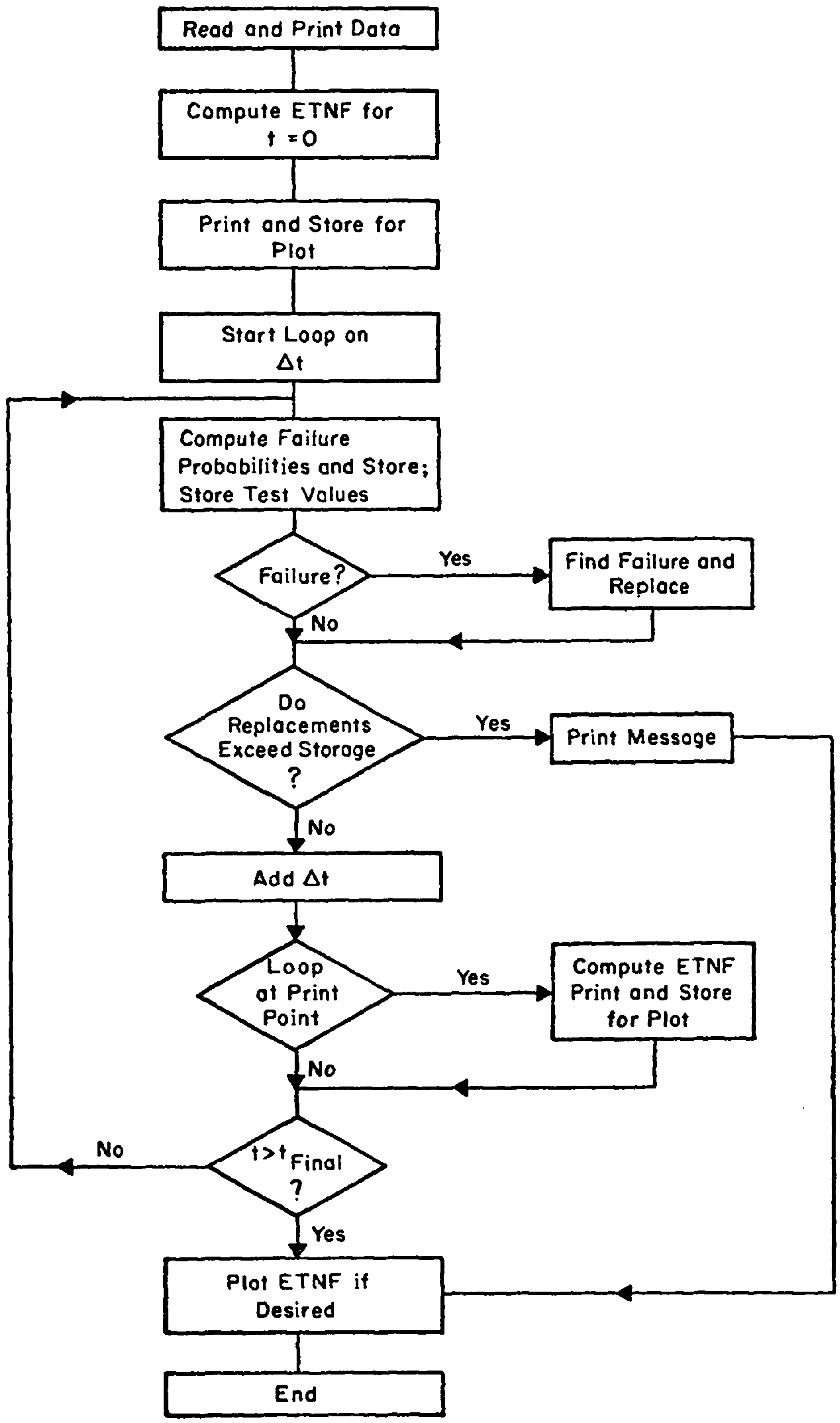

Fig. 1.

Flow diagram for computation of expected-time-to-next-failure.

components and systems. Additional distributions can be added to the program if desired. A subroutine must be written for the distribution and it can be modeled after the distribution subroutines already included. To include the calls to the subroutine, an additional GOTO branch is then required in the computed GOTO statements in subroutines DIDITFL and CETNF.

A. Exponential Distribution

The exponential distribution is followed by many components and component assemblies, provided sufficient bench testing has been done before installation.[1,2,3] Components that follow different exponential distributions can be combined easily into a single composite type, also of the exponential family, provided that the components are connected statistically in series.

The exponential density function is

$$f(t) = \begin{cases} 0 & t < \beta \\ \alpha e^{-\alpha(t-\beta)} & t \geq \beta \end{cases}$$

The parameter β is the "guarantee" time. The failure rate is a constant

$$r = \begin{cases} 0 & t < \beta \\ \alpha & t \geq \beta \end{cases}$$

and the a posteriori failure probability in the interval Δt is

$$p(t) = \begin{cases} 0 & t < \beta \\ 1 - e^{-\alpha \Delta t} & t \geq \beta \end{cases}$$

B. Weibull Distribution

The Weibull density function is[1,3,4]

$$f(t) = \begin{cases} 0 & t < \gamma \\ \dfrac{\beta(t-\gamma)^{\beta-1}}{\alpha} \exp\left[-(t-\gamma)^{\beta}/\alpha\right] & t > \gamma \end{cases}$$

Here γ represents the guarantee time. For $t = \gamma$, the value of f depends on β. We have

$$f(\gamma) = \begin{cases} 0 & \beta > 1 \\ 1/\alpha & \beta = 1 \\ \infty & \beta < 1 \end{cases}$$

Notice that this produces a singularity in the failure rate when $\beta < 1$. The failure rate is given by

$$r(t) = \begin{cases} 0 & t < \gamma \\ 0 & t = \gamma,\ \beta > 1 \\ 1/\alpha & t = \gamma,\ \beta = 1 \\ \infty & t = \gamma,\ \beta < 1 \\ (\beta/\alpha)(t-\gamma)^{\beta-1} & t > \gamma \end{cases}$$

Because the rate is well behaved for $t > \gamma$ when $\beta < 1$, we arbitrarily set $t = \gamma + 0.01\Delta t$ if $\beta < 1$. This is merely a device to obtain a finite failure rate for computing purposes. If $\beta = 1$, the Weibull distribution reduces to an exponential distribution and such instances are probably better handled as exponential. For replacement units we ignore the singularity, and set the contribution for the unit to zero when $t = \gamma$. The a posteriori failure probability for the Weibull distribution is

$$p(t) = \begin{cases} 0 & t < \gamma \\ 1 - e^{-(1/\alpha)\left[(t+\Delta t-\gamma)^{\beta} - (t-\gamma)^{\beta}\right]} & t \geq \gamma \end{cases}$$

C. Normal Distribution (Truncated Normal)

Two forms of the normal distribution are commonly used in reliability computations:[1,2,3] the standard normal and the truncated normal. The density function of each is of the form

$$f(t) = (C/\beta\sqrt{2\pi})\ \exp\left[-(t-\alpha)^2/(2\beta^2)\right].$$

C is a normalizing constant determined from the condition that the integral of f(t) equals one. In the case of the standard normal distribution, the integral ranges over all t values from $-\infty$ to $+\infty$. In the truncated distribution, t ranges only from 0 to $+\infty$, on the assumption that no failures occur until t is greater than zero. For the purpose of the expected-time-to-next-failure calculation, it makes no difference which distribution we consider because the constant C disappears and we obtain identical values for the failure rate and a posteriori probability of failure. These values are given by

$$r(t) = \frac{(\sqrt{2/\pi}\ \exp\left[-(t-\alpha)^2/(2\beta^2)\right]}{\beta\ \mathrm{erfc}\ \left[(t-\alpha)/(\beta\sqrt{2})\right]}$$

and

$$p(t) = \frac{\sqrt{2/\pi}\int_t^{t+\Delta t} \exp\left[-(x-\alpha)^2/(2\beta^2)\right]\, dx}{\beta\ \mathrm{erfc}\ \left[(t-\alpha)/(\beta\sqrt{2})\right]}$$

The integral appearing in the expression for p(t) could be converted to the difference of two error function values, but this would lead to considerable round-off error for small Δt. In the program the integral is computed numerically, using a 41-point Simpson's rule.

D. Logarithmic Normal Distribution

If the logarithm of a random variable has a normal distribution, the variable itself follows a logarithmic normal distribution. There are at least three log normal distributions, ranging from two parameters to four parameters.[1,2,3,5] We use a three-parameter distribution which includes a guaranteed life. The density function is

$$
f(t) = \begin{cases} 0 & t \le \gamma \\ \dfrac{1}{(t-\gamma)\,\beta\,\sqrt{2\pi}} \exp\left\{-\left(\ell n(t-\gamma) - \alpha\right)^2/(2\beta^2)\right\} & t > \gamma \end{cases}
$$

which reduces to the standard two-parameter distribution when $\gamma = 0$.

The failure rate is given by

$$
r(t) = \begin{cases} 0 & t \le \gamma \\ \dfrac{\sqrt{2/\pi}\; e^{-\left\{\ell n(t-\gamma) - \alpha\right\}^2/2\beta^2}}{(t-\gamma)\,\beta\,\mathrm{erfc}\left\{\dfrac{\ell n\,(t-\gamma) - \alpha}{\beta\sqrt{2}}\right\}} & t > \gamma \end{cases}
$$

and the a posteriori failure probability by

$$
p(t) = \begin{cases} 0 & t < \gamma \\ \dfrac{\sqrt{2/\pi}}{\beta\,\mathrm{erfc}\left\{\dfrac{\ell n(t-\gamma) - \alpha}{\beta\sqrt{2}}\right\}} \displaystyle\int_t^{t+\Delta t} \frac{1}{(t'-\gamma)}\, e^{-\left\{\ell n(t'-\gamma) - \alpha\right\}^2 / 2\beta^2}\, dt' & t > \gamma \\ (1/2\beta)\,\sqrt{2/\pi} \displaystyle\int_t^{t+\Delta t} \frac{1}{(t'-\gamma)}\, e^{-\left\{\ell n(t'-\gamma) - \alpha\right\}^2/ 2\beta^2}\, dt' & t = \gamma \end{cases}
$$

A 41-point Simpson's rule is also used to find this integral. In this case we also assume that γ is an integral multiple of Δt.

E. Gamma Distribution

The gamma distribution in its three-parameter form has the density function.[1,3]

$$
f(t) = \begin{cases} 0 & t - \gamma < 0 \\ \dfrac{\alpha\left\{\alpha(t-\gamma)\right\}^{\beta-1}\, e^{-\alpha(t-\gamma)}}{\Gamma(\beta)} & t - \gamma \ge 0 \end{cases}
$$

The exponential and Erlang distributions are special cases of this distribution. The failure rate is given by

$$r(t) = \begin{cases} 0 & t - \gamma \le 0 \\ \dfrac{\alpha\left\{\alpha(t-\gamma)\right\}^{\beta-1} e^{-\alpha(t-\gamma)}}{\Gamma\left(\beta, \alpha(t-\gamma)\right)} & t - \gamma > 0 \end{cases}$$

and the a posteriori failure probability by

$$p(t) = \begin{cases} 0 & t - \gamma \le 0 \\ \dfrac{\alpha \displaystyle\int_t^{t+\Delta t} \left\{\alpha(t' - \gamma)\right\}^{\beta-1} e^{-\alpha(t'-\gamma)}\, dt'}{\Gamma\left(\beta, \alpha(t-\gamma)\right)} & t - \gamma > 0 \end{cases}$$

In those formulas $\Gamma(\beta,u)$ is one of the incomplete gamma functions, and is defined by

$$\Gamma(\beta,u) = \int_u^{\infty} x^{\beta-1} e^{-x}\, dx \quad .$$

The integral in the expression for p(t) could be expressed as the difference between incomplete gamma functions, but would result in considerable round-off error when Δt is small. A 41-point Simpson's rule is used instead and, as in the log normal case, γ is assumed to be an integral multiple of Δt.

F. Uniform Distribution

The uniform distribution has the density function[1]

$$f(t) = \begin{cases} 0 & t < \alpha \text{ and } t \ge \beta \\ \dfrac{1}{\beta - \alpha} & \alpha \le t < \beta \end{cases}$$

The failure rate is

$$r(t) = \begin{cases} 0 & t < \alpha \text{ and } t \geq \beta \\ \dfrac{1}{\beta - t} & \alpha \leq t < \beta \end{cases}$$

and the a posteriori failure probability is

$$p(t) = \begin{cases} 0 & t < \alpha \\ \dfrac{\Delta t}{\beta - t} & \alpha \leq t < \beta \\ 1 & \beta \leq t \end{cases}$$

G. Rayleigh Distribution

The Rayleigh distribution has the density[6]

$$f(t) = \begin{cases} 0 & -\infty \leq t < t_o \\ \dfrac{(t - t_o)}{\sigma^2} e^{-\frac{(t-t_o)^2}{2\sigma^2}} & t_o \leq t < \infty \end{cases}$$

This distribution is a special case of the Weibull distribution, as is easily shown by making the following substitutions in the Weibull density function:

$$\alpha = 2\sigma^2 \quad ,$$

$$\beta = 2 \quad ,$$

$$\gamma = t_o \quad .$$

To input a Rayleigh component type to the program, the first parameter is σ and the second parameter is t_o. The program makes the above substitutions and thereafter the component is treated as if it were following a Weibull distribution.

IV. Description of the Program

The calculation has been described. The subroutines and their functions are described below, a complete listing is given in Appendix A, and an example is given in Appendix B. The program is written for the CDC 7600 using the CROS operating system.

Subroutine	Function
EXPECT	DRIVER FOR PROGRAM The program calls SETUP and initializes certain variables. A loop on the time step Δt is started and continued until the required final time is reached or until one of three other conditions requires that the calculation be terminated. Diagnostic prints are made in the latter event. The loop calls the subroutine DIDITFL to determine if a failure occurred; subroutine FAILURE is called if one occurred. One time step is then added to each component of the system being considered, and subroutine CETNF is called. Data for a plot is stored if a plot is desired, and a print is made if an output time has been reached. On exit from the loop, the program makes a plot if it has been requested.
SETUP	Reads and prints the input data, initializes the replacement array, and determines the index of the last time step required. A Rayleigh distribution component is changed to a Weibull component.
CETNF	Calculates the ETNF. It calls PEXPON, PWEIB, PNORM, PLNORM, PGAMMA, and PUNIFM.
DIDITFL	Determines by Monte Carlo methods whether a failure occurred by the end of the current time step. It calls PPEXPON, PPWEIB, PPNORM, PPLNORM, PPGAMMA. and PPUNIFM. It signals the main program if the system failure probability is too great.
FAILURE	This routine is called when DIDITFL decides that a failure has occurred. It determines which component failed and replaces the component.

The following six subroutines compute the failure rates and a posteriori failure probabilities for the various distributions. The probabilities are stored for future use. In each case, the failure rate is calculated by a call to the subroutine, whereas failure probabilities are calculated by a call to the entry name.

Subroutine	Entry	Function
PEXPON	PPEXPON	Used for components following exponential distributions.
PWEIB	PPWEIB	Used for components following Weibull distributions.
PNORM	PPNORM	Used for components following normal distributions.
PLNORM	PPLNORM	Used for components following log normal distributions.
PGAMMA	PPGAMMA	Used for components following gamma distribution.
PUNIF	PPUNIF	Used for components following uniform distributions.

The following subroutines are used to compute integrals.

ERK	LOGERK	ERK is called by PPNORM to compute the a posteriori failure probability of a single component of normal type. A 41-point Simpson's rule is used for the required integration. LOGERK performs a similar computation for single log normal components.
GAMPROB		This routine is called by PPGAMMA to compute the a posteriori failure probability of a single component following a gamma distribution.

V. INPUT REQUIREMENTS

TITLE CARD	Format (8A10)
Cols	
1-80	Title
CONTROL CARD	Format (4I6, 2E12.6)
Cols	
1-6	Number of component groups. The program will accept up to 10 groups and can be modified to accept more. These groups may obey the same or different types of distribution.
7-12	MSP, an integer giving the spacing in numbers of steps of Δt desired between output points.
13-18	Plot control. A one in column 18 indicates a plot is desired; otherwise no plot is made.
19-24	Probability print control. A one in column 24 will cause a print of the a posteriori failure probabilities for each component group. These prints occur with the same spacing as the ETNF output points.
25-36	Time at which last output point is desired. May not be greater than 1000*Δt*MSP unless the program storage is modified.
37-48	Time step, Δt.
COMPONENT CARDS	For each component group the following two cards must be present: Group title card Format (8A10) and Distribution card Format (2I12, 3E12.6).

Cols	
1-12	An integer indicating the type of distribution followed by the components in the group according to the following code: 1 - Exponential distribution 2 - Weibull distribution 3 - Normal distribution 4 - Log normal distribution 5 - Gamma distribution 6 - Uniform distribution 7 - Rayleigh distribution
13-24	An integer giving the number of components in the group.
25-36	α - First distribution parameter
37-48	β - Second distribution parameter
49-60	γ - Third distribution parameter.

The α, β, and γ required for the distributions must conform to the notation used in the test. If a second or third parameter is not required. the corresponding field on the distribution card may be left blank.

ACKNOWLEDGMENTS

I would like to thank R. H. Moore and G. L. Tietgen for their helpful advice.

REFERENCES

1. B. V. Gnedenko, Yu. K. Belyayev. and A. D. Solovyev, *Mathematical Methods of Reliability Theory* (Academic Press, New York and London. 1969).

2. Igor Bazovsky, *Reliability Theory and Practice* (Prentice Hall, Inc., Englewood Cliffs, NJ, 1961).

3. R. E. Barlow and F. Proschan, *Mathematical Theory of Reliability* (Wiley, New York, 1965).

4. W. Weibull, "A Statistical Distribution Function of Wide Applicability," J. Appl. Mech. **18**, 293-297 (1951).

5. J. Aitchison and J. A. C. Brown, *The Log Normal Distribution* (Cambridge University Press, London, 1957).

6. A. M. Polovko, *Fundamentals of Reliability Theory* (Academic Press, New York and London, 1968).

APPENDIX A

FORTRAN LISTING OF XPECT PROGRAM

LASL identification number : LP-0534

```
      PROGRAM XPECT (INP,OUT,FILM)
C     THIS PROGRAM COMPUTES THE EXPECTED NUMBER OF SHOTS BETWEEN FAILURE
C     OR MALFUNCTIONS FOR A SYSTEM HAVING UP TO 10 TYPES OF COMPONENTS
C     THESE COMPONENT TYPES MAY FOLLOW ANY OF THE FOLLOWING FAILURE
C     DISTRIBUTIONS
C         1---EXPONENTIAL DISTRIBUTION
C         2---WEIBULL DISTRIBUTION
C         3---NORMAL DISTRIBUTION
C         4---LOG NORMAL DISTRIBUTION
C         5---GAMMA DISTRIBUTION
C         6---UNIFORM DISTRIBUTION
C         7---RAYLEIGH DISTRIBUTION
C     THE INPUT REQUIREMENTS ARE
C     A TITLE CARD         FORMAT 8A10
C     A SINGLE CARD GIVING
C         THE NUMBER OF DIFFERENT COMPONENT TYPES--FORMATI6
C         THE SPACING BETWEEN OUTPUT VALUES          FORMATI6
C         A ONE IN COLUMN 18 IF A PLOT IS DESIRED
C         A ONE IN COLUMN 24 IF PROBABILITIES ARE DESIRED
C         THE LAST TIME OUTPUT IS NEEDED             FORMAT E12.6
C         TIME STEP                                  FORMAT E12.6
C     FOR EACH TYPE THE FOLLOWING DATA
C         CARD 1--NAME OF COMPONENT                          FORMAT 10A10
C         CARD 2--COMPONENT DISTRIBUTION TYPE  (ITYPE(J))    FORMAT I12
C                 NUMBER OF COMPONENTS OF TYPE (NORIG(J))    FORMAT I12
C                 1ST DISTRIBUTION PARAMETER   (ALPHA(J))    FORMAT E12.6
C                 2ND DISTRIBUTION PARAMETER   (BETA(J))     FORMAT E12.6
C                 3ED DISTRIBUTION PARAMETER   (GAMMA(J))    FORMAT E12.6
C     IN CASES WHERE ONLY 1 OR 2 PARAMETERS ARE USED LEAVE SPACE BLANK
      COMMON /XP1/ TYPE(8,10),TITLE(8),EXPECT(1001),X(1001),LABELY(3)
      COMMON /XP2/ ALPHA(10),BETA(10),GAMMA(10),RETNF(10)
      COMMON /XP3/ NORIG(10),NREPLAC(10),IREPL(1000,10),PSUM,IPROB,MSP
      COMMON /XP4/ NGROUPS,PROB(10),P(1000,10),IGROUP(10),PTEST(10)
      COMMON /XP5/ TLAST,TDELTA,NSHOT,SHOTS,IPLOT,LASTSHT,NREP(10)
      DATA LABELX/10HTIME      /
      DATA LABELY/29HEXPECTED TIME TO NEXT FAILURE/
      CALL SETUP
      KPRINT=0
      NSHOT=0
      SHOTS=0.
      CALL CETNF (ETNF,IREASON)
      IF (IREASON.EQ.2) PRINT 14
      PRINT 13, NSHOT,ETNF
      EXPECT(1)=ETNF
      X(1)=0.
C     START LOOP ON SHOTS
          DO 6 NSHOT=1,LASTSHT
          SHOTS=FLOAT(NSHOT)*TDELTA
          CALL DIDITFL (IFAIL)
          GO TO (2,1,11), IFAIL
C     A FAILURE OCCURED BRANCH TO ROUTINE TO DECIDE WHICH TYPE FAILED
1         CALL FAILURE
C     ADD A SHOT TO ALL REPLACEMENT UNITS
2             DO 4 J=1,NGROUPS
              KSTOP=NREPLAC(J)
              IF (KSTOP.EQ.0) GO TO 4
              IF (KSTOP.GT.1000) GO TO 7
                  DO 3 K=1,KSTOP
3                 IREPL(K,J)=IREPL(K,J)+1
4             CONTINUE
          IF (MSP.EQ.1) GO TO 5
          IPRINT=NSHOT+1
          IF (MOD(IPRINT,MSP).NE.1) GO TO 6
5         CALL CETNF (ETNF,IREASON)
          KPRINT=KPRINT+1
          X(KPRINT)=SHOTS
          EXPECT(KPRINT)=ETNF
          IF (IREASON.EQ.2) PRINT 14
          PRINT 13, SHOTS,ETNF
6         CONTINUE
      PRINT 18, ((NREP(I),I),I=1,NGROUPS)
      GO TO 8
7     PRINT 16, J
```

```
8     IF (IPLOT.NE.1) GO TO 12
C     PLOT IF DESIRED
9     IF (KPRINT.GT.1001) GO TO 10
      CALL PLOJB (X,EXPECT,KPRINT,1,0,46,0,10.,6.,TITLE,80,LABELX,10,LAB
     1ELY,29)
      GO TO 12
10    PRINT 15
      GO TO 12
C     PROBABILITY OF FAILURE GREATER THAN OR EQUAL TO 1.
11    PRINT 17
      IF (IPLOT.EQ.1.AND.KPRINT.GT.1) GO TO 9
12    CONTINUE
      RETURN
C
C
C
13    FORMAT (1H ,* AT TIME *,E13.6,*      EXPECTED TIME TO NEXT FAILURE
     1=*,E13.6)
14    FORMAT (1H ,* FAILURE RATE IS ZERO SO ETNF WOULD BE INFINITE.*/* R
     1UN CONTINUES.*)
15    FORMAT (1H ,* NUMBER OF POINTS DESIRED PLOTED GREATER THAN 1000. V
     1ECTOR EXPECT HAS OVERFLOWED. NO PLOT MADE.*)
16    FORMAT (1HO,* NUMBER OF REPLACEMENTS OF COMPONENT TYPE *,I3,* EXCE
     1EDS ALLOWED STORAGE*/*  RUN TERMINATED.*)
17    FORMAT (1H ,* RUN TERMINATED TO GIVE YOU TIME TO THINK.*)
18    FORMAT (1HO,//10(I5,* UNITS OF GROUP*,I3,* WERE REPLACED*/))
      END
```

```
      SUBROUTINE SETUP
      COMMON /XP1/ TYPE(8,10),TITLE(8),EXPECT(1001),X(1001),LABELY(3)
      COMMON /XP2/ ALPHA(10),BETA(10),GAMMA(10),RETNF(10)
      COMMON /XP3/ NORIG(10),NREPLAC(10),IREPL(1000,10),PSUM,IPROB,MSP
      COMMON /XP4/ NGROUPS,PROB(10),P(1000,10),IGROUP(10),PTEST(10)
      COMMON /XP5/ TLAST,TDELTA,NSHOT,SHOTS,IPLOT,LASTSHT,NREP(10)
      DIMENSION YN(2), NAME(2)
      DATA NAME(1),NAME(2)/10H LOGNORMAL,10H    GAMMA /
      DATA YN/10H  NO.     ,10H YES.     /
      READ 8, (TITLE(I),I=1,8)
      PRINT 5, (TITLE(I),I=1,8)
      READ 6, NGROUPS,MSP,IPLOT,IPROB,TLAST,TDELTA
      IP=1
      IPB=1
      IF (IPLOT.EQ.1) IP=2
      IF (IPROB.EQ.1) IPB=2
      PRINT 7, NGROUPS,MSP,TLAST,TDELTA,YN(IP),YN(IPB)
           DO 1 I=1,NGROUPS
           READ 8, (TYPE(J,I),J=1,8)
           READ 9, IGROUP(I),NORIG(I),ALPHA(I),BETA(I),GAMMA(I)
           PRINT 10, I,(TYPE(J,I),J=1,8),IGROUP(I),NORIG(I),ALPHA(I),BET
     1     A(I),GAMMA(I)
1          CONTINUE
C     ZERO. THE REPLACEMENT VECTOR
           DO 2 I=1,NGROUPS
           NREP(I)=0
2          NREPLAC(I)=0
      LASTSHT=IFIX(TLAST/TDELTA)+1
           DO 3 I=1,NGROUPS
           IF (IGROUP(I).NE.7) GO TO 3
           IGROUP(I)=2
           ALPHA(I)=2.*ALPHA(I)**2
           GAMMA(I)=BETA(I)
           BETA(I)=2.
3          CONTINUE
           DO 4 I=1,NGROUPS
           IF (IGROUP(I).NE.4.AND.IGROUP(I).NE.5) GO TO 4
           TEMP=GAMMA(I)/TDELTA
           ITEMP=INT(TEMP)
           TEMP=(TEMP-FLOAT(ITEMP))*TDELTA
           GAMMA(I)=GAMMA(I)-TEMP
           J=MOD(IGROUP(I),3)
```

```
            IF (TEMP.NE.0.) PRINT 12, IGROUP(I),NAME(J),GAMMA(I)              SETUP 44
4           CONTINUE                                                          SETUP 45
        PRINT 11                                                              SETUP 46
        RETURN                                                                SETUP 47
C                                                                             SETUP 48
C                                                                             SETUP 49
5       FORMAT (1H ,8A10)                                                     SETUP 50
6       FORMAT (4I6,2E12.6)                                                   SETUP 51
7       FORMAT (1H0,* NUMBER OF GROUPS OF COMPONENTS CONSIDERED-----------SETUP 52
       1-*,I5/*  SPACING DESIRED BETWEEN OUTPUT DATA------------------*,I5SETUP 53
       2/*  FINAL TIME DESIRED-----------------------------------*,E12.6/*SETUP 54
       3  TIME STEP----------------------------------------*,E12.6/*  SETUP 55
       4IS A PLOT DESIRED-------------------------------------*,A10/*  ARE SETUP 56
       5PROBABILITY PRINTS DESIRED------------------------*,A10)          SETUP 57
8       FORMAT (8A10)                                                         SETUP 58
9       FORMAT (2I12,3E12.6)                                                  SETUP 59
10      FORMAT (1H0,* GROUP*,I3,/2X,8A10/*  DISTRIBUTION TYPE NUMBER*,I3/*SETUP 60
       1  NUMBER OF UNITS*,I6,/*  ALPHA=*,E12.6,* BETA=*,E13.6,* GAMMA=*,ESETUP 61
       212.6)                                                                 SETUP 62
11      FORMAT (1H1)                                                          SETUP 63
12      FORMAT (1H0,/* FOR COMPONENT GROUP*,I3,*, OBEYING*,A10,* DISTRIBUTSETUP 64
       1ION, GAMMA PARAMETER IS A NONINTEGRAL MULTIPLE OF DELTA T.*/*  GAMSETUP 65
       2MA PARAMETER HAS BEEN CHANGED TO*,E14.6)                              SETUP 66
        END                                                                   SETUP 67
```

```
        SUBROUTINE  CETNF(ETNF,IREASON)                                       CETNF  2
        COMMON /XP2/ ALPHA(10),BETA(10),GAMMA(10),RETNF(10)                   CETNF  3
        COMMON /XP3/ NORIG(10),NREPLAC(10),IREPL(1000,10),PSUM,IPROB,MSP      CETNF  4
        COMMON /XP4/ NGROUPS,PROB(10),P(1000,10),IGROUP(10),PTEST(10)         CETNF  5
        COMMON /XP5/ TLAST,TDELTA,NSHOT,SHOTS,IPLOT,LASTSHT,NREP(10)          CETNF  6
C       FORM THE SUM OF F(X)/(1-INT(F(X)) FOR EACH COMPONENT OF EACH TYPE.CETNF  7
C       EXPECTED NUMBER OF SHOTS TO NEXT FAILURE IS RECIPROCOL OF THIS SUMCETNF  8
        IREASON=1                                                             CETNF  9
            DO 7 I=1,NGROUPS                                                  CETNF 10
            IWORK=IGROUP(I)                                                   CETNF 11
            GO TO (1,2,3,4,5,6), IWORK                                        CETNF 12
1           CALL PEXPON (I)                                                   CETNF 13
            GO TO 7                                                           CETNF 14
2           CALL PWEIB (I)                                                    CETNF 15
            GO TO 7                                                           CETNF 16
3           CALL PNORM (I)                                                    CETNF 17
            GO TO 7                                                           CETNF 18
4           CALL PLNORM (I)                                                   CETNF 19
            GO TO 7                                                           CETNF 20
5           CALL PGAMMA (I)                                                   CETNF 21
            GO TO 7                                                           CETNF 22
6           CALL PUNIFM (I)                                                   CETNF 23
7           CONTINUE                                                          CETNF 24
C       SUM THE INDIVIDUAL FAILURE RATES AND TAKE RECIPROCAL                  CETNF 25
        SUM=0.                                                                CETNF 26
            DO 8 I=1,NGROUPS                                                  CETNF 27
8           SUM=SUM+RETNF(I)                                                  CETNF 28
        IF (SUM.EQ.0.) GO TO 9                                                CETNF 29
        ETNF=1./SUM                                                           CETNF 30
        RETURN                                                                CETNF 31
9       IREASON=2                                                             CETNF 32
        ETNF=1.E+300                                                          CETNF 33
        RETURN                                                                CETNF 34
C                                                                             CETNF 35
        END                                                                   CETNF 36
```

```
        SUBROUTINE  DIDITFL(IFAIL)                                            DIDITFL2
        COMMON /XP2/ ALPHA(10),BETA(10),GAMMA(10),RETNF(10)                   DIDITFL3
        COMMON /XP3/ NORIG(10),NREPLAC(10),IREPL(1000,10),PSUM,IPROB,MSP      DIDITFL4
        COMMON /XP4/ NGROUPS,PROB(10),P(1000,10),IGROUP(10),PTEST(10)         DIDITFL5
        COMMON /XP5/ TLAST,TDELTA,NSHOT,SHOTS,IPLOT,LASTSHT,NREP(10)          DIDITFL6
        COMMON /XP6/ PS(10),PPROB                                             DIDITFL7
```

```
      LCM /XP7/ PSAVE(10000,10)
C     FOR EACH COMPONENT TYPE COMPUTE THE PROBABILITY OF FAILURE
C     THIS PROBABILITY IS THE A POSTERIORI PROBABILITY SINCE ALL
C     COMPONENTS WERE OPERATING ON ENTRY TO SUBROUTINE.
      PPROB=1.
           DO 7 I=1,NGROUPS
           IWORK=IGROUP(I)
           GO TO (1,2,3,4,5,6), IWORK
 1         CALL PPEXPON (I)
           GO TO 7
 2         CALL PPWEIB (I)
           GO TO 7
 3         CALL PPNORM (I)
           GO TO 7
 4         CALL PPLNORM (I)
           GO TO 7
 5         CALL PPGAMMA (I)
           GO TO 7
 6         CALL PPUNIFM (I)
 7         CONTINUE
      Y=RANDOM(DUMMY)
      PSUM=1.-PPROB
      IFAIL=1
      IF (PSUM.GE.1.) IFAIL=3
      IF (IPROB.NE.1) GO TO 10
C     PRINT PROBABILITIES IF DESIRED.
      IF (MSP.EQ.1) GO TO 8
      IPRINT=NSHOT+1
      IF (MOD(IPRINT,MSP).NE.1) GO TO 10
 8    PRINT 14, SHOTS
           DO 9 IMPRINT=1,NGROUPS
 9         PRINT 16, IMPRINT,PS(IMPRINT)
 10   IF (IFAIL.NE.3) GO TO 12
      PRINT 15, NSHOT,SHOTS
           DO 11 I=1,NGROUPS
 11        PRINT 16, I,PS(I)
      GO TO 13
 12   IF (Y.LE.PSUM) IFAIL=2
 13   RETURN
C
C
C
 14   FORMAT (1H ,* A POSTERIORI COMPONENT GROUP FAILURE PROBABILITY AT
     1TIME *,E13.6)
 15   FORMAT (1H0,* ON SHOT*,I6,* AT TIME *,E13.6,* PROBABILITY OF FAILU
     1RE TOO LARGE. YOUR SYSTEM WONT WORK.*/*  WE WILL PRINT THE PROBABI
     2LITIES SO YOU CAN SEE WHICH COMPONENT DID IT.*)
 16   FORMAT (1H ,* COMPONENT*,I2,* PROB OF FAILURE=*,E15.7)
      END


      SUBROUTINE FAILURE
      COMMON /XP3/ NORIG(10),NREPLAC(10),IREPL(1000,10),PSUM,IPROB,MSP
      COMMON /XP4/ NGROUPS,PROB(10),P(1000,10),IGROUP(10),PTEST(10)
      COMMON /XP5/ TLAST,TDELTA,NSHOT,SHOTS,IPLOT,LASTSHT,NREP(10)
C     FIND SUM OF INDIVIDUAL FAILURE PROBABILITIES FOR NORMALIZATION.
C     DURING SUMMATION FIND AND SAVE CONTRIBUTIONS OF EACH GROUP.
C     PTEST HOLDS TOTAL FOR GROUP, PROB HOLDS CONTRIBUTION OF ORIGINAL
C     UNITS AND P HOLDS CONTRIBUTION OF REPLACEMENT UNITS.
      SUM=0.
           DO 1 I=1,NGROUPS
           NTEMP=NORIG(I)-NREPLAC(I)
           IF (NTEMP.LT.1) PROB(I)=0.
           PROB(I)=PROB(I)*FLOAT(NTEMP)
 1         SUM=SUM+PROB(I)
           DO 3 I=1,NGROUPS
           ISTOP=NREPLAC(I)
           IF (ISTOP.EQ.0) GO TO 3
                DO 2 J=1,ISTOP
 2              SUM=SUM+P(J,I)
 3         CONTINUE
```

```
      RECPSUM=1./SUM
           DO 4 I=1,NGROUPS
4          PROB(I)=PROB(I)*RECPSUM
           DO 6 I=1,NGROUPS
           SUM=0.
           ISTOP=NREPLAC(I)
           IF (ISTOP.EQ.0) GO TO 6
                DO 5 J=1,ISTOP
                P(J,I)=P(J,I)*RECPSUM
5               SUM=SUM+P(J,I)
6          PTEST(I)=PROB(I)+SUM
C     FIND FAILED UNIT
      Y=RANDOM(DUMMY)
C     1ST FIND TYPE
      SUM=0.
           DO 7 I=1,NGROUPS
           SUM=SUM+PTEST(I)
7          IF (Y.LE.SUM) GO TO 8
C     FAILURE WAS OF TYPE I
C     DETERMINE IF FAILURE WAS ORIGINAL UNIT OR REPLACEMENT
8     SUM=SUM-PTEST(I)
      IF (NREPLAC(I).EQ.0) GO TO 10
      JSTOP=NREPLAC(I)
           DO 9 J=1,JSTOP
           SUM=SUM+P(J,I)
9          IF (Y.LE.SUM) GO TO 11
C     IF PROGRAM REACHES THIS POINT FAILURE WAS AN ORIGINAL UNIT OF
C     TYPE I. ADD 1 TO THE REPLACEMENT INDEX AND SET SHOT COUNT ON THE
C     NEW UNIT TO -1.
10    NREPLAC(I)=NREPLAC(I)+1
      NREP(I)=NREP(I)+1
      IDUMMY=NREPLAC(I)
      IREPL(IDUMMY,I)=-1
      RETURN
C     FAILED UNIT WAS REPLACEMENT UNIT J OF TYPE I. SET SHOT COUNT ON IT
C     TO -1
11    IREPL(J,I)=-1
      NREP(I)=NREP(I)+1
      RETURN
      END



      SUBROUTINE  PEXPON(I)
C     FOR COMPONENTS FOLLOWING EXPONENTIAL STATISTICS.
C     THE FAILURE RATE IS INDEPENDENT OF THE NUMBER OF SHOTS
      COMMON /XP2/ ALPHA(10),BETA(10),GAMMA(10),RETNF(10)
      COMMON /XP3/ NORIG(10),NREPLAC(10),IREPL(1000,10),PSUM,IPROB,MSP
      COMMON /XP4/ NGROUPS,PROB(10),P(1000,10),IGROUP(10),PTEST(10)
      COMMON /XP5/ TLAST,TDELTA,NSHOT,SHOTS,IPLOT,LASTSHT,NREP(10)
      COMMON /XP6/ PS(10),PPROB
      LCM /XP7/ PSAVE(10000,10)
C     CALCULATE FAILURE RATE
      INDEX=NREPLAC(I)
      RETNF(I)=ALPHA(I)*FLOAT(NORIG(I))
C     REMOVE CONTRIBUTIONS OF UNITS WITH LESS THAN BETA SHOTS
      TEMP=SHOTS-BETA(I)
      IF (TEMP.LT.0.) GO TO 2
      IF (INDEX.EQ.0) RETURN
           DO 1 J=1,INDEX
           TEMP=FLOAT(IREPL(J,I))*TDELTA-BETA(I)
1          IF (TEMP.LT.0.) RETNF(I)=RETNF(I)-ALPHA(I)
      RETURN
2     RETNF(I)=0.
      RETURN
      ENTRY PPEXPON
C     CALCULATE THE A POSTERIORI PROBABILITY
      INDEX=NREPLAC(I)
      TEMP=SHOTS-BETA(I)
      IF (TEMP.GE.0.) GO TO 3
      PROB(I)=0.
      PS(I)=0.
```

```
      PSAVE(NSHOT,I)=0.                                                 PEXPON31
      RETURN                                                            PEXPON32
    3 PROB(I)=1.-EXP(-ALPHA(I)*TDELTA)                                  PEXPON33
      PSAVE(NSHOT,I)=PROB(I)                                            PEXPON34
      PS(I)=1.                                                          PEXPON35
      MULTTO=NORIG(I)-INDEX                                             PEXPON36
      IF (MULTTO.LT.1) GO TO 5                                          PEXPON37
      PS(I)=1.-PROB(I)                                                  PEXPON38
      IF (MULTTO.EQ.1) GO TO 5                                          PEXPON39
          DO 4 J=2,MULTTO                                               PEXPON40
    4     PS(I)=PS(I)*(1.-PROB(I))                                      PEXPON41
    5 IF (INDEX.EQ.0) GO TO 7                                           PEXPON42
          DO 6 J=1,INDEX                                                PEXPON43
          K=IREPL(J,I)+1                                                PEXPON44
          P(J,I)=PSAVE(K,I)                                             PEXPON45
    6     PS(I)=PS(I)*(1.-P(J,I))                                       PEXPON46
    7 PPROB=PPROB*PS(I)                                                 PEXPON47
      PS(I)=1.-PS(I)                                                    PEXPON48
      RETURN                                                            PEXPON49
      END                                                               PEXPON50


      SUBROUTINE  PWEIB(I)                                              PWEIB  2
C     FOR COMPONENTS FOLLOWING WEIBULL STATISTICS                       PWEIB  3
      COMMON /XP2/ ALPHA(10),BETA(10),GAMMA(10),RETNF(10)               PWEIB  4
      COMMON /XP3/ NORIG(10),NREPLAC(10),IREPL(1000,10),PSUM,IPROB,MSP  PWEIB  5
      COMMON /XP4/ NGROUPS,PROB(10),P(1000,10),IGROUP(10),PTEST(10)     PWEIB  6
      COMMON /XP5/ TLAST,TDELTA,NSHOT,SHOTS,IPLOT,LASTSHT,NREP(10)      PWEIB  7
      COMMON /XP6/ PS(10),PPROB                                         PWEIB  8
      LCM /XP7/ PSAVE(10000,10)                                         PWEIB  9
      INDEX=NREPLAC(I)                                                  PWEIB 10
      TEMP=SHOTS-GAMMA(I)                                               PWEIB 11
      IF (TEMP.LT.0.) GO TO 4                                           PWEIB 12
      IF (TEMP.EQ.0.) GO TO 5                                           PWEIB 13
      NTEMP=NORIG(I)-INDEX                                              PWEIB 14
      RETNF(I)=0.                                                       PWEIB 15
      IF (NTEMP.LT.1) GO TO 1                                           PWEIB 16
      RETNF(I)=BETA(I)*(TEMP**(BETA(I)-1.))*FLOAT(NTEMP)/ALPHA(I)       PWEIB 17
    1 IF (INDEX.EQ.0) RETURN                                            PWEIB 18
          DO 3 J=1,INDEX                                                PWEIB 19
          TEMP=FLOAT(IREPL(J,I))*TDELTA-GAMMA(I)                        PWEIB 20
          IF (TEMP.LT.0.) GO TO 3                                       PWEIB 21
          IF (TEMP.EQ.0.) GO TO 2                                       PWEIB 22
          RETNF(I)=RETNF(I)+BETA(I)*(TEMP**(BETA(I)-1.))/ALPHA(I)       PWEIB 23
          GO TO 3                                                       PWEIB 24
    2     IF (BETA(I).NE.1.) GO TO 3                                    PWEIB 25
          RETNF(I)=RETNF(I)+1./ALPHA(I)                                 PWEIB 26
    3     CONTINUE                                                      PWEIB 27
      RETURN                                                            PWEIB 28
C     NUMBER OF SHOTS LESS THAN GAMMA NO FAILURES CAN OCCUR             PWEIB 29
    4 RETNF(I)=0.                                                       PWEIB 30
      RETURN                                                            PWEIB 31
C     NUMBER OF SHOTS EQUALS GAMMA                                      PWEIB 32
    5 IF (BETA(I).LE.1.) GO TO 6                                        PWEIB 33
      RETNF(I)=0.                                                       PWEIB 34
      RETURN                                                            PWEIB 35
    6 IF (BETA(I).LT.1.) GO TO 7                                        PWEIB 36
      RETNF(I)=1./ALPHA(I)                                              PWEIB 37
      RETURN                                                            PWEIB 38
    7 PRINT 13                                                          PWEIB 39
      TEMP=.01                                                          PWEIB 40
      RETNF(I)=BETA(I)*(TEMP**(BETA(I)-1.))*FLOAT(NORIG(I)-INDEX)/ALPHA(PWEIB 41
     1I)                                                                PWEIB 42
      RETURN                                                            PWEIB 43
      ENTRY PPWEIB                                                      PWEIB 44
C     CALCULATE THE A POSTERIORI PROBABILITY                            PWEIB 45
      INDEX=NREPLAC(I)                                                  PWEIB 46
      TEMP=SHOTS-GAMMA(I)                                               PWEIB 47
      IF (TEMP.LT.1.) GO TO 12                                          PWEIB 48
      TEMP=((TEMP-TDELTA)**BETA(I)-TEMP**BETA(I))/ALPHA(I)              PWEIB 49
      PROB(I)=1.-EXP(TEMP)                                              PWEIB 50
```

```
      PSAVE(NSHOT,I)=PROB(I)
      PS(I)=1.
      MULTTO=NORIG(I)-INDEX
      IF (MULTTO.LT.1) GO TO 9
      PS(I)=1.-PROB(I)
      IF (MULTTO.EQ.1) GO TO 9
          DO 8 J=2,MULTTO
8         PS(I)=PS(I)*(1.-PROB(I))
9     IF (INDEX.EQ.0) GO TO 11
          DO 10 J=1,INDEX
          K=IREPL(J,I)+1
          P(J,I)=PSAVE(K,I)
10        PS(I)=PS(I)*(1.-P(J,I))
11    PPROB=PPROB*PS(I)
      PS(I)=1.-PS(I)
      RETURN
12    PROB(I)=0.
      PS(I)=0.
      PSAVE(NSHOT,I)=0.
      RETURN
C
C
13    FORMAT (1HO,/* FOR THE WEIBULL DISTRIBUTION, BETA LESS THAN 1 AND
     1TIME-GAMMA=0 CAUSES THE FAILURE RATE TO APPROACH INFINITY.*/* SINC
     2E IT WILL BE WELL BEHAVED FOR TIME-GAMMA GREATER THAN ZERO, TIME-G
     3AMMA IS GIVEN A SMALL POSITIVE VALUE */* AND THE FAILURE RATE IS C
     4ALCULATED FOR THIS VALUE.  THE INFINITIES DUE TO REPLACEMENTS ARE
     5IGNORED.*/* IT IS POSSIBLE THAT THIS MAY CAUSE DISCONTINUITIES IN
     6THE OVERALL ETNF.*//)
      END
```

```
      SUBROUTINE  PNORM(I)
C     FOR COMPONENTS FOLLOWING NORMAL STATISTICS
      COMMON /XP2/ ALPHA(10),BETA(10),GAMMA(10),RETNF(10)
      COMMON /XP3/ NORIG(10),NREPLAC(10),IREPL(1000,10),PSUM,IPROB,MSP
      COMMON /XP4/ NGROUPS,PROB(10),P(1000,10),IGROUP(10),PTEST(10)
      COMMON /XP5/ TLAST,TDELTA,NSHOT,SHOTS,IPLOT,LASTSHT,NREP(10)
      COMMON /XP6/ PS(10),PPROB
      LCM /XP7/ PSAVE(10000,10)
      DATA C1,C2/1.4142135623731,0.79788456080286/
      INDEX=NREPLAC(I)
      TEST=(SHOTS-ALPHA(I))/(BETA(I)*C1)
      IF (TEST.GT.26.) GO TO 3
      FBAR=BETA(I)*ERFC(TEST)
      DF=C2*EXP(-TEST**2)
      NTEMP=NORIG(I)-INDEX
      RETNF(I)=0.
      IF (NTEMP.LT.1) GO TO 1
      RETNF(I)=FLOAT(NTEMP)*DF/FBAR
1     IF (INDEX.EQ.0) RETURN
          DO 2 J=1,INDEX
          TEST=(FLOAT(IREPL(J,I))*TDELTA-ALPHA(I))/(BETA(I)*C1)
          FBAR=BETA(I)*ERFC(TEST)
          DF=C2*EXP(-TEST**2)
2         RETNF(I)=RETNF(I)+DF/FBAR
      RETURN
C     IF PROGRAM REACHES THIS POINT FAILURE IS VIRTUALY CERTAIN
C     WE ARBITRARILY SET RETNF=1.E+100 AND RETURN
3     RETNF(I)=1.E+100
      RETURN
      ENTRY PPNORM
C     CALCULATE THE A POSTERIORI PROBABILITY
      INDEX=NREPLAC(I)
      CALL ERK (PROB(I),ALPHA(I),BETA(I))
      PSAVE(NSHOT,I)=PROB(I)
      PS(I)=1.
      MULTTO=NORIG(I)-INDEX
      IF (MULTTO.LT.1) GO TO 5
      PS(I)=1.-PROB(I)
      IF (MULTTO.EQ.1) GO TO 5
```

```
          DO 4 J=2,MULTTO                                                PNORM 41
4         PS(I)=PS(I)*(1.-PROB(I))                                       PNORM 42
5     IF (INDEX.EQ.0) GO TO 7                                            PNORM 43
          DO 6 J=1,INDEX                                                 PNORM 44
          K=IREPL(J,I)+1                                                 PNORM 45
          P(J,I)=PSAVE(K,I)                                              PNORM 46
6         PS(I)=PS(I)*(1.-P(J,I))                                        PNORM 47
7     PPROB=PPROB*PS(I)                                                  PNORM 48
      PS(I)=1.-PS(I)                                                     PNORM 49
      RETURN                                                             PNORM 50
      END                                                                PNORM 51


      SUBROUTINE  PLNORM(I)                                              PLNORM 2
C     FOR COMPONENTS FOLLOWING LOG NORMAL STATISTICS                     PLNORM 3
      COMMON /XP2/ ALPHA(10),BETA(10),GAMMA(10),RETNF(10)                PLNORM 4
      COMMON /XP3/ NORIG(10),NREPLAC(10),IREPL(1000,10),PSUM,IPROB,MSP   PLNORM 5
      COMMON /XP4/ NGROUPS,PROB(10),P(1000,10),IGROUP(10),PTEST(10)      PLNORM 6
      COMMON /XP5/ TLAST,TDELTA,NSHOT,SHOTS,IPLOT,LASTSHT,NREP(10)       PLNORM 7
      COMMON /XP6/ PS(10),PPROB                                          PLNORM 8
      LCM /XP7/ PSAVE(10000,10)                                          PLNORM 9
      DATA C1,C2/1.4142135623731,0.79788456080286/                       PLNORM10
      INDEX=NREPLAC(I)                                                   PLNORM11
      RETNF(I)=0.                                                        PLNORM12
      TEMP1=SHOTS-GAMMA(I)                                               PLNORM13
      IF (TEMP1.LE.0.) RETURN                                            PLNORM14
      D=1./(C1*BETA(I))                                                  PLNORM15
      TEMP2=(ALOG(TEMP1)-ALPHA(I))*D                                     PLNORM16
      IF (TEMP2.GE.26.) GO TO 3                                          PLNORM17
      NTEMP=NORIG(I)-INDEX                                               PLNORM18
      IF (NTEMP.LT.1) GO TO 1                                            PLNORM19
      RETNF(I)=C2*EXP(-TEMP2**2)/(TEMP1*BETA(I)*ERFC(TEMP2))             PLNORM20
      RETNF(I)=FLOAT(NTEMP)*RETNF(I)                                     PLNORM21
1     IF (INDEX.EQ.0) RETURN                                             PLNORM22
          DO 2 J=1,INDEX                                                 PLNORM23
          TEMP1=FLOAT(IREPL(J,I))*TDELTA-GAMMA(I)                        PLNORM24
          IF (TEMP1.LE.0.) GO TO 2                                       PLNORM25
          TEMP2=(ALOG(TEMP1)-ALPHA(I))*D                                 PLNORM26
          RETNF(I)=RETNF(I)+C2*EXP(-TEMP2**2)/(TEMP1*BETA(I)*ERFC(TEMP2PLNORM27
     1    ))                                                             PLNORM28
2         CONTINUE                                                       PLNORM29
      RETURN                                                             PLNORM30
C     IF TEMP2.GE.26. ERFC WILL UNDERFLOW.  ARBITRARILY SET              PLNORM31
C     RETNF(I)=1.E+100 AND RETURN                                        PLNORM32
3     RETNF(I)=1.E+100                                                   PLNORM33
      RETURN                                                             PLNORM34
      ENTRY PPLNORM                                                      PLNORM35
C     CALCULATE THE A POSTERIORI PROBABILITY                             PLNORM36
      INDEX=NREPLAC(I)                                                   PLNORM37
      CALL LOGERK (PROB(I),ALPHA(I),BETA(I),GAMMA(I))                    PLNORM38
      PSAVE(NSHOT,I)=PROB(I)                                             PLNORM39
      PS(I)=1.                                                           PLNORM40
      MULTTO=NORIG(I)-INDEX                                              PLNORM41
      IF (MULTTO.LT.1) GO TO 5                                           PLNORM42
      PS(I)=1.-PROB(I)                                                   PLNORM43
      IF (MULTTO.EQ.1) GO TO 5                                           PLNORM44
          DO 4 J=2,MULTTO                                                PLNORM45
4         PS(I)=PS(I)*(1.-PROB(I))                                       PLNORM46
5     IF (INDEX.EQ.0) GO TO 7                                            PLNORM47
          DO 6 J=1,INDEX                                                 PLNORM48
          K=IREPL(J,I)+1                                                 PLNORM49
          P(J,I)=PSAVE(K,I)                                              PLNORM50
6         PS(I)=PS(I)*(1.-P(J,I))                                        PLNORM51
7     PPROB=PPROB*PS(I)                                                  PLNORM52
      PS(I)=1.-PS(I)                                                     PLNORM53
      RETURN                                                             PLNORM54
      END                                                                PLNORM55
```

```
      SUBROUTINE  PGAMMA(I)
C     FOR COMPONENTS FOLLOWING GAMMA DISTRIBUTIONS
C     GAMMA IN COMMON/XP2/ HAS BEEN CHANGED TO ZAMMA TO ALLOW THE  USE
C     OF A FUNCTION ON DISC
      COMMON /XP2/ ALPHA(10),BETA(10),ZAMMA(10),RETNF(10)
      COMMON /XP3/ NORIG(10),NREPLAC(10),IREPL(1000,10),PSUM,IPROB,MSP
      COMMON /XP4/ NGROUPS,PROB(10),P(1000,10),IGROUP(10),PTEST(10)
      COMMON /XP5/ TLAST,TDELTA,NSHOT,SHOTS,IPLOT,LASTSHT,NREP(10)
      COMMON /XP6/ PS(10),PPROB
      LCM /XP7/ PSAVE(10000,10)
      INDEX=NREPLAC(I)
      U=ALPHA(I)*(SHOTS-ZAMMA(I))
      RETNF(I)=0.
      IF (U.LE.0.) RETURN
      NTEMP=NORIG(I)-INDEX
      IF (NTEMP.LT.1) GO TO 1
      RETNF(I)=ALPHA(I)*U**(BETA(I)-1.)*EXP(-U)
      RETNF(I)=FLOAT(NTEMP)*RETNF(I)/GAMMA(BETA(I),U)
    1 IF (INDEX.EQ.0) RETURN
          DO 2 J=1,INDEX
          U=ALPHA(I)*(FLOAT(IREPL(J,I))*TDELTA-ZAMMA(I))
          IF (U.LE.0.) GO TO 2
          RATE=ALPHA(I)*U**(BETA(I)-1.)*EXP(-U)
          RATE=RATE/GAMMA(BETA(I),U)
          RETNF(I)=RETNF(I)+RATE
    2     CONTINUE
      RETURN
      ENTRY PPGAMMA
C     CALCULATE THE A POSTERIORI PROBABILITY
      INDEX=NREPLAC(I)
      IF (SHOTS-ZAMMA(I).LT.0.) GO TO 7
      CALL GAMPROB (ALPHA(I),BETA(I),ZAMMA(I),PROB(I))
      PSAVE(NSHOT,I)=PROB(I)
      PS(I)=1.
      MULTTO=NORIG(I)-INDEX
      IF (MULTTO.LT.1) GO TO 4
      PS(I)=1.-PROB(I)
      IF (MULTTO.EQ.1) GO TO 4
          DO 3 J=2,MULTTO
    3     PS(I)=PS(I)*(1.-PROB(I))
    4 IF (INDEX.EQ.0) GO TO 6
          DO 5 J=1,INDEX
          K=IREPL(J,I)+1
          P(J,I)=PSAVE(K,I)
    5     PS(I)=PS(I)*(1.-P(J,I))
    6 PPROB=PPROB*PS(I)
      PS(I)=1.-PS(I)
      RETURN
C     PROBABILITY OF FAILURE IS ZERO, SHOTS LESS THAN GAMMA
    7 PROB(I)=0.
      PSAVE(NSHOT,I)=0.
      PS(I)=0.
      RETURN
      END
```

```
      SUBROUTINE  PUNIFM(I)
C     FOR COMPONENTS FOLLOWING UNIFORM STATISTICS
      COMMON /XP2/ ALPHA(10),BETA(10),GAMMA(10),RETNF(10)
      COMMON /XP3/ NORIG(10),NREPLAC(10),IREPL(1000,10),PSUM,IPROB,MSP
      COMMON /XP4/ NGROUPS,PROB(10),P(1000,10),IGROUP(10),PTEST(10)
      COMMON /XP5/ TLAST,TDELTA,NSHOT,SHOTS,IPLOT,LASTSHT,NREP(10)
      COMMON /XP6/ PS(10),PPROB
      LCM /XP7/ PSAVE(10000,10)
      INDEX=NREPLAC(I)
      RETNF(I)=0.
      IF (SHOTS.LT.ALPHA(I)) RETURN
      IF (SHOTS.GE.BETA(I)) RETURN
      NTEMP=NORIG(I)-INDEX
      IF (NTEMP.LT.1) GO TO 1
      RETNF(I)=FLOAT(NTEMP)/(BETA(I)-SHOTS)
```

```
1     IF (INDEX.EQ.0) RETURN
         DO 2 J=1,INDEX
         TEMP=TDELTA*FLOAT(IREPL(J,I))
         IF (TEMP.LT.ALPHA(I)) GO TO 2
         RETNF(I)=RETNF(I)+1./(BETA(I)-TEMP)
2        CONTINUE
      RETURN
      ENTRY PPUNIFM
C     CALCULATE THE A POSTERIORI PROBABILITY
      INDEX=NREPLAC(I)
      PROB(I)=0.
      PSAVE(NSHOT,I)=0.
      PS(I)=0.
      IF (SHOTS.LT.ALPHA(I)) RETURN
      IF (SHOTS.GE.BETA(I)) GO TO 7
      PROB(I)=TDELTA/(BETA(I)-SHOTS)
      PROB(I)=AMIN1(1.,PROB(I))
      PSAVE(NSHOT,I)=PROB(I)
      PS(I)=1.
      MULTTO=NORIG(I)-INDEX
      IF (MULTTO.LT.1) GO TO 4
      PS(I)=1.-PROB(I)
      IF (MULTTO.EQ.1) GO TO 4
         DO 3 J=2,MULTTO
3        PS(I)=PS(I)*(1.-PROB(I))
4     IF (INDEX.EQ.0) GO TO 6
         DO 5 J=1,INDEX
         K=IREPL(J,I)+1
         P(J,I)=PSAVE(K,I)
5        PS(I)=PS(I)*(1.-P(J,I))
6     PPROB=PPROB*PS(I)
      PS(I)=1.-PS(I)
      RETURN
C     PROBABILITY OF FAILURE IS 1. IF NUMBER OF SHOTS .GE. BETA
7     PROB(I)=1.
      PS(I)=1.
      PPROB=0.
      RETURN
      END
```

```
      SUBROUTINE  ERK(P,ALPHA,BETA,GAMMA)
C     ERK COMPUTES THE A POSTERIORI FAILURE PROBABILITY
C     FOR A NORMALLY DISTRIBUTED COMPONENT. IT COMPUTES THE INTEGRAL
C     OF THE DISTRIBUTION FUNCTION FROM SHOT N-1 TO SHOT N USING A 41
C     POINT SIMPSONS RULE.
C     ENTRY LOGERK DOES THE SAME FOR A LOG NORMAL COMPONENT.
      COMMON /XP5/ TLAST,TDELTA,NSHOT,SHOTS,IPLOT,LASTSHT,NREP(10)
      DATA C1,C2/1.414213562373,0.797884560802/
      B=SHOTS-ALPHA
      A=B-TDELTA
      STEP=.025*TDELTA
      Q=-.5/(BETA**2)
      P=EXP(Q*A**2)
         DO 1 I=2,40,2
1        P=P+4.*EXP(Q*(A+FLOAT(I-1)*STEP)**2)+2.*EXP(Q*(A+FLOAT(I)*STE
     1   P)**2)
      P=P-EXP(Q*B**2)
      P=STEP*C2*P/(3.*BETA*ERFC(A/(BETA*C1)))
      RETURN
      ENTRY LOGERK
      P=0.
      B=SHOTS-GAMMA
      IF (B.LE.0.) RETURN
      STEP=.025*TDELTA
      A=B-TDELTA
      Q=1./(2.*BETA*BETA)
      IF (A.EQ.0.) GO TO 2
      P=EXP(-(ALOG(A)-ALPHA)**2*Q)/A
2        DO 3 I=2,40,2
         X1=A+FLOAT(I-1)*STEP
```

```
          X2=A+FLOAT(I)*STEP
3         P=P+4.*EXP(-(ALOG(X1)-ALPHA)**2*Q)/X1+2.*EXP(-(ALOG(X2)-ALPHA
     1    )**2*Q)/X2
      P=P-EXP(-(ALOG(B)-ALPHA)**2*Q)/B
      IF (A.EQ.O.) GO TO 4
      P=P*STEP*C2/(3.*BETA*ERFC((ALOG(A)-ALPHA)/(BETA*C1)))
      RETURN
4     P=P*C2*STEP/(6.*BETA)
      RETURN
      END


      SUBROUTINE  GAMPROB(ALPHA,BETA,ZAMMA,PROB)
C     THIS ROUTINE COMPUTES THE A POSTERIORI PROBABILITY OF FAILURE
C     BETWEEN T AND T+DELTA T OF A COMPONENT WHICH FOLLOWS A GAMMA
C     DISTRIBUTION.  IT USES A 41-POINT SIMPSONS RULE.
      COMMON /XP5/ TLAST,TDELTA,NSHOT,SHOTS,IPLOT,LASTSHT,NREP(10)
      B=SHOTS-ZAMMA
      A=B-TDELTA
      STEP=.025*TDELTA
      IF (B.LE.O.) GO TO 4
      IF (A.EQ.O.) GO TO 1
      SUM=(ALPHA*A)**(BETA-1.)*EXP(-ALPHA*A)
      GO TO 2
1     SUM=O.
2         DO 3 I=2,40,2
          AIM1=(A+FLOAT(I-1)*STEP)*ALPHA
          AI=(A+FLOAT(I)*STEP)*ALPHA
          SUM=SUM+4.*(AIM1)**(BETA-1.)*EXP(-AIM1)+2.*AI**(BETA-1.)*EXP(
     1    -AI)
3         CONTINUE
      SUM=SUM-(ALPHA*B)**(BETA-1.)*EXP(-ALPHA*B)
      U=ALPHA*A
      Z=GAMMA(BETA,U)
      PROB=SUM*STEP*ALPHA/(3.*Z)
      RETURN
C     SHOTS ARE BELOW GUARANTEED LIFE. BY ASSUMPTION NO FAILURES OCCUR.
4     PROB=O.
      RETURN
      END
```

APPENDIX B

EXAMPLE OF ETNF CALCULATION

An example of the expected-time-to-next-failure computation is given for seven groups of hypothetical components that represent the seven distribution types the program accepts. The distribution type is used as the group name and the parameters used are those given in the test problem printout below. These parameters were chosen to illustrate the use of the program and do not, in general, correspond to known components. Probability prints for this example were not requested so that the output listing would be shorter.

```
12345678901234567890123456789012345678901234567890123456789012345678901234567890
00000000011111111112222222222333333333344444444445555555555666666666677777777778


TEST PROBLEM
     7    40       1          2000.          0.5
EXPONENTIAL
               1        1000        .000098    100.
WEIBULL
               2        10007000.                .75           0.0
NORMAL
               3         100 2500.             400.
LOG NORMAL
               4         100    15.             50.           -3.7
GAMMA
               5         100        .01         20.           -5.2
UNIFORM
               6        1000   500.         400000.
RAYLEIGH
               7         20010000.              00.
```

Fig. B-1.
Input to program.

```
TEST PROBLEM

 NUMBER OF GROUPS OF COMPONENTS CONSIDERED------------     7
 SPACING DESIRED BETWEEN OUTPUT DATA------------------    40
 FINAL TIME DESIRED-----------------------------------  .200000E+04
 TIME STEP--------------------------------------------  .500000E+00
 IS A PLOT DESIRED------------------------------------  YES.
 ARE PROBABILITY PRINTS DESIRED-----------------------  NO.

 GROUP  1
 EXPONENTIAL
 DISTRIBUTION TYPE NUMBER  1
 NUMBER OF UNITS  1000
 ALPHA= .800000E-05 BETA=  .100000E+03 GAMMA=0.

 GROUP  2
 WEIBULL
 DISTRIBUTION TYPE NUMBER  2
 NUMBER OF UNITS  1000
 ALPHA= .700000E+05 BETA=  .750000E+00 GAMMA=0.

 GROUP  3
 NORMAL
 DISTRIBUTION TYPE NUMBER  3
 NUMBER OF UNITS   100
 ALPHA= .250000E+04 BETA=  .400000E+03 GAMMA=0.

 GROUP  4
 LOG NORMAL
 DISTRIBUTION TYPE NUMBER  4
 NUMBER OF UNITS   100
 ALPHA= .150000E+02 BETA=  .500000E+02 GAMMA=-.370000E+01

 GROUP  5
 GAMMA
 DISTRIBUTION TYPE NUMBER  5
 NUMBER OF UNITS   100
 ALPHA= .100000E-01 BETA=  .200000E+02 GAMMA=-.520000E+01

 GROUP  6
 UNIFORM
 DISTRIBUTION TYPE NUMBER  6
 NUMBER OF UNITS  1000
 ALPHA= .500000E+03 BETA=  .400000E+06 GAMMA=0.

 GROUP  7
 RAYLEIGH
 DISTRIBUTION TYPE NUMBER  7
 NUMBER OF UNITS   200
 ALPHA= .100000E+05 BETA=  .400000E+03 GAMMA=0.

FOR COMPONENT GROUP  4, OBEYING LOGNORMAL DISTRIBUTION, GAMMA PARAMETER IS A NONINTEGRAL MULTIPLE OF DELTA T.
 GAMMA PARAMETER HAS BEEN CHANGED TO  -.350000E+01  .

FOR COMPONENT GROUP  5, OBEYING    GAMMA  DISTRIBUTION, GAMMA PARAMETER IS A NONINTEGRAL MULTIPLE OF DELTA T.
 GAMMA PARAMETER HAS BEEN CHANGED TO  -.500000E+01  .
```

Fig. B-2.
Program output page 1.

FOR THE WEIBULL DISTRIBUTION, BETA LESS THAN 1 AND TIME-GAMMA=0 CAUSES THE FAILURE RATE TO APPROACH INFINITY. SINCE IT WILL BE WELL BEHAVED FOR TIME-GAMMA GREATER THAN ZERO, TIME-GAMMA IS GIVEN A SMALL POSITIVE VALUE AND THE FAILURE RATE IS CALCULATED FOR THIS VALUE. THE INFINITIES DUE TO REPLACEMENTS ARE IGNORED. IT IS POSSIBLE THAT THIS MAY CAUSE DISCONTINUITIES IN THE OVERALL ETNF.

```
AT TIME  0.              EXPECTED TIME TO NEXT FAILURE=  .253338E+01
AT TIME  .200000E+02     EXPECTED TIME TO NEXT FAILURE=  .161916E+02
AT TIME  .400000E+02     EXPECTED TIME TO NEXT FAILURE=  .286802E+02
AT TIME  .600000E+02     EXPECTED TIME TO NEXT FAILURE=  .373147E+02
AT TIME  .800000E+02     EXPECTED TIME TO NEXT FAILURE=  .484776E+02
AT TIME  .100000E+03     EXPECTED TIME TO NEXT FAILURE=  .389625E+02
AT TIME  .120000E+03     EXPECTED TIME TO NEXT FAILURE=  .440377E+02
AT TIME  .140000E+03     EXPECTED TIME TO NEXT FAILURE=  .479247E+02
AT TIME  .160000E+03     EXPECTED TIME TO NEXT FAILURE=  .512217E+02
AT TIME  .180000E+03     EXPECTED TIME TO NEXT FAILURE=  .522071E+02
AT TIME  .200000E+03     EXPECTED TIME TO NEXT FAILURE=  .557670E+02
AT TIME  .220000E+03     EXPECTED TIME TO NEXT FAILURE=  .583863E+02
AT TIME  .240000E+03     EXPECTED TIME TO NEXT FAILURE=  .605971E+02
AT TIME  .260000E+03     EXPECTED TIME TO NEXT FAILURE=  .6[illegible]5555E+02
AT TIME  .280000E+03     EXPECTED TIME TO NEXT FAILURE=  .643204E+02
AT TIME  .300000E+03     EXPECTED TIME TO NEXT FAILURE=  .658923E+02
AT TIME  .320000E+03     EXPECTED TIME TO NEXT FAILURE=  .673274E+02
AT TIME  .340000E+03     EXPECTED TIME TO NEXT FAILURE=  .686825E+02
AT TIME  .360000E+03     EXPECTED TIME TO NEXT FAILURE=  .699373E+02
AT TIME  .380000E+03     EXPECTED TIME TO NEXT FAILURE=  .711038E+02
AT TIME  .400000E+03     EXPECTED TIME TO NEXT FAILURE=  .722336E+02
AT TIME  .420000E+03     EXPECTED TIME TO NEXT FAILURE=  .730388E+02
AT TIME  .440000E+03     EXPECTED TIME TO NEXT FAILURE=  .737708E+02
AT TIME  .460000E+03     EXPECTED TIME TO NEXT FAILURE=  .744365E+02
AT TIME  .480000E+03     EXPECTED TIME TO NEXT FAILURE=  .750418E+02
AT TIME  .500000E+03     EXPECTED TIME TO NEXT FAILURE=  .635318E+02
AT TIME  .520000E+03     EXPECTED TIME TO NEXT FAILURE=  .638827E+02
AT TIME  .540000E+03     EXPECTED TIME TO NEXT FAILURE=  .642317E+02
AT TIME  .560000E+03     EXPECTED TIME TO NEXT FAILURE=  .645481E+02
AT TIME  .580000E+03     EXPECTED TIME TO NEXT FAILURE=  .647991E+02
AT TIME  .600000E+03     EXPECTED TIME TO NEXT FAILURE=  .650177E+02
AT TIME  .620000E+03     EXPECTED TIME TO NEXT FAILURE=  .652137E+02
AT TIME  .640000E+03     EXPECTED TIME TO NEXT FAILURE=  .653298E+02
AT TIME  .660000E+03     EXPECTED TIME TO NEXT FAILURE=  .654072E+02
AT TIME  .680000E+03     EXPECTED TIME TO NEXT FAILURE=  .630540E+02
AT TIME  .700000E+03     EXPECTED TIME TO NEXT FAILURE=  .642399E+02
AT TIME  .720000E+03     EXPECTED TIME TO NEXT FAILURE=  .646251E+02
AT TIME  .740000E+03     EXPECTED TIME TO NEXT FAILURE=  .647406E+02
AT TIME  .760000E+03     EXPECTED TIME TO NEXT FAILURE=  .646946E+02
AT TIME  .780000E+03     EXPECTED TIME TO NEXT FAILURE=  .645161E+02
AT TIME  .800000E+03     EXPECTED TIME TO NEXT FAILURE=  .642426E+02
AT TIME  .820000E+03     EXPECTED TIME TO NEXT FAILURE=  .638250E+02
AT TIME  .840000E+03     EXPECTED TIME TO NEXT FAILURE=  .632482E+02
AT TIME  .860000E+03     EXPECTED TIME TO NEXT FAILURE=  .625502E+02
AT TIME  .880000E+03     EXPECTED TIME TO NEXT FAILURE=  .616572E+02
AT TIME  .900000E+03     EXPECTED TIME TO NEXT FAILURE=  .605635E+02
AT TIME  .920000E+03     EXPECTED TIME TO NEXT FAILURE=  .593236E+02
AT TIME  .940000E+03     EXPECTED TIME TO NEXT FAILURE=  .579[illegible]E+02
AT TIME  .960000E+03     EXPECTED TIME TO NEXT FAILURE=  .562754E+02
AT TIME  .980000E+03     EXPECTED TIME TO NEXT FAILURE=  .545281E+02
AT TIME  .100000E+04     EXPECTED TIME TO NEXT FAILURE=  .526279E+02
AT TIME  .102000E+04     EXPECTED TIME TO NEXT FAILURE=  .505978E+02
AT TIME  .104000E+04     EXPECTED TIME TO NEXT FAILURE=  .484625E+02
AT TIME  .106000E+04     EXPECTED TIME TO NEXT FAILURE=  .463884E+02
```

Fig. B-3.
Program output page 2.

```
AT TIME   .108000E+04      EXPECTED TIME TO NEXT FAILURE=  .441336E+02
AT TIME   .110000E+04      EXPECTED TIME TO NEXT FAILURE=  .418566E+02
AT TIME   .112000E+04      EXPECTED TIME TO NEXT FAILURE=  .395807E+02
AT TIME   .114000E+04      EXPECTED TIME TO NEXT FAILURE=  .373391E+02
AT TIME   .116000E+04      EXPECTED TIME TO NEXT FAILURE=  .351492E+02
AT TIME   .118000E+04      EXPECTED TIME TO NEXT FAILURE=  .331968E+02
AT TIME   .120000E+04      EXPECTED TIME TO NEXT FAILURE=  .313187E+02
AT TIME   .122000E+04      EXPECTED TIME TO NEXT FAILURE=  .295422E+02
AT TIME   .124000E+04      EXPECTED TIME TO NEXT FAILURE=  .276949E+02
AT TIME   .126000E+04      EXPECTED TIME TO NEXT FAILURE=  .261150E+02
AT TIME   .128000E+04      EXPECTED TIME TO NEXT FAILURE=  .244694E+02
AT TIME   .130000E+04      EXPECTED TIME TO NEXT FAILURE=  .230819E+02
AT TIME   .132000E+04      EXPECTED TIME TO NEXT FAILURE=  .208753E+02
AT TIME   .134000E+04      EXPECTED TIME TO NEXT FAILURE=  .202520E+02
AT TIME   .136000E+04      EXPECTED TIME TO NEXT FAILURE=  .190863E+02
AT TIME   .138000E+04      EXPECTED TIME TO NEXT FAILURE=  .179567E+02
AT TIME   .140000E+04      EXPECTED TIME TO NEXT FAILURE=  .162486E+02
AT TIME   .142000E+04      EXPECTED TIME TO NEXT FAILURE=  .160193E+02
AT TIME   .144000E+04      EXPECTED TIME TO NEXT FAILURE=  .156547E+02
AT TIME   .146000E+04      EXPECTED TIME TO NEXT FAILURE=  .147956E+02
AT TIME   .148000E+04      EXPECTED TIME TO NEXT FAILURE=  .139838E+02
AT TIME   .150000E+04      EXPECTED TIME TO NEXT FAILURE=  .132235E+02
AT TIME   .152000E+04      EXPECTED TIME TO NEXT FAILURE=  .126254E+02
AT TIME   .154000E+04      EXPECTED TIME TO NEXT FAILURE=  .120678E+02
AT TIME   .156000E+04      EXPECTED TIME TO NEXT FAILURE=  .115483E+02
AT TIME   .158000E+04      EXPECTED TIME TO NEXT FAILURE=  .110656E+02
AT TIME   .160000E+04      EXPECTED TIME TO NEXT FAILURE=  .108126E+02
AT TIME   .162000E+04      EXPECTED TIME TO NEXT FAILURE=  .103833E+02
AT TIME   .164000E+04      EXPECTED TIME TO NEXT FAILURE=  .100884E+02
AT TIME   .166000E+04      EXPECTED TIME TO NEXT FAILURE=  .990412E+01
AT TIME   .168000E+04      EXPECTED TIME TO NEXT FAILURE=  .954168E+01
AT TIME   .170000E+04      EXPECTED TIME TO NEXT FAILURE=  .921319E+01
AT TIME   .172000E+04      EXPECTED TIME TO NEXT FAILURE=  .889252E+01
AT TIME   .174000E+04      EXPECTED TIME TO NEXT FAILURE=  .860269E+01
AT TIME   .176000E+04      EXPECTED TIME TO NEXT FAILURE=  .840607E+01
AT TIME   .178000E+04      EXPECTED TIME TO NEXT FAILURE=  .831240E+01
AT TIME   .180000E+04      EXPECTED TIME TO NEXT FAILURE=  .823174E+01
AT TIME   .182000E+04      EXPECTED TIME TO NEXT FAILURE=  .825935E+01
AT TIME   .184000E+04      EXPECTED TIME TO NEXT FAILURE=  .830745E+01
AT TIME   .186000E+04      EXPECTED TIME TO NEXT FAILURE=  .837245E+01
AT TIME   .188000E+04      EXPECTED TIME TO NEXT FAILURE=  .815128E+01
AT TIME   .190000E+04      EXPECTED TIME TO NEXT FAILURE=  .815367E+01
AT TIME   .192000E+04      EXPECTED TIME TO NEXT FAILURE=  .819602E+01
AT TIME   .194000E+04      EXPECTED TIME TO NEXT FAILURE=  .793978E+01
AT TIME   .196000E+04      EXPECTED TIME TO NEXT FAILURE=  .797989E+01
AT TIME   .198000E+04      EXPECTED TIME TO NEXT FAILURE=  .780262E+01
AT TIME   .200000E+04      EXPECTED TIME TO NEXT FAILURE=  .763425E+01


16 UNITS OF GROUP  1 WERE REPLACED
 4 UNITS OF GROUP  2 WERE REPLACED.
13 UNITS OF GROUP  3 WERE REPLACED
12 UNITS OF GROUP  4 WERE REPLACED
63 UNITS OF GROUP  5 WERE REPLACED
 4 UNITS OF GROUP  6 WERE REPLACED
 4 UNITS OF GROUP  7 WERE REPLACED
```

Fig. B-4.
Program output page 3.

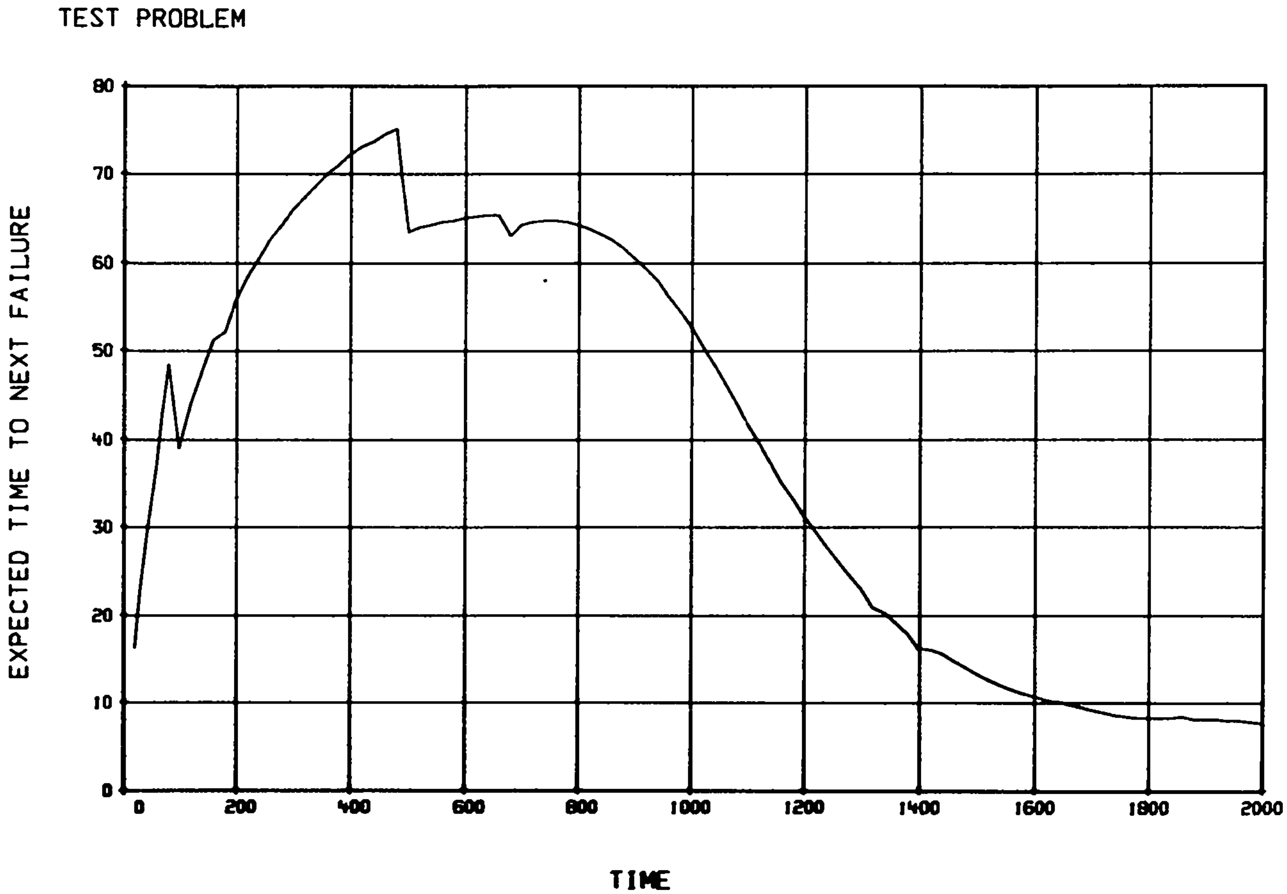

Fig. B-5.
Film output.