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

by

G. P. Bolcourt

Los Alamos Scientific Laboratory of the University of California
LOS ALAMOS, NEW MEXICO 87545
An Affirmative Action/Equal Opportunity Employer
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 pre-
sent CTR systems occur predominantly in developmental com-
ponents being used in large quantities.

I. INTRODUCTION

The ability to predict failure rates is of increasing importance in controlled ther-
monuclear 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

\[
\frac{dr}{dt} = \frac{1}{R_s} \frac{dR_s}{dt},
\]

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

\[
R_s = \prod_{i=1}^{N} R_i,
\]

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

\[
R_i = \int_{0}^{\infty} f_i(t) dt,
\]

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

From (3),

\[
\frac{dR_i}{dt} = -f_i(t).
\]

Also,

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

\[
= - \sum_{i=1}^{N} \left( \prod_{j \neq i}^{N} R_j \right) f_i(t).
\]
Hence,

\[-\frac{1}{\bar{R}_S} \frac{d\bar{R}_S}{dt} = \frac{1}{\prod_{j=1}^N R_j} \left( \sum_{i=1}^N \left( \prod_{j \neq i}^N R_j \right) f_i(t) \right) \]

so, from (1),

\[x(t) = \sum_{i=1}^N f_i(t) \]

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

\[\text{ETNF}(t) = \frac{1}{x(t)} \]

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

\[\text{ETNF}(t) = 1 \left/ \sum_{j=1}^N \frac{f_j(t)}{1 - F_j(t)} \right. \]

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

\[F_j(t) = \int_0^t f_j(t') \, dt' \]

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

\[ J = \text{number of component types} \]
\[ N_k(t) = \text{number of original units of the } k^{\text{th}} \text{ type at time } t \]
\[ M_k(t) = \text{number of replacement units of the } k^{\text{th}} \text{ type at time } t \]
\[ f_{ok} = \text{probability density associated with all remaining original units of the } k^{\text{th}} \text{ type} \]
\[ f_{ik} = \text{probability density associated with the } i^{\text{th}} \text{ individual replacement unit of the } k^{\text{th}} \text{ type} \]
\[ F_{ok} = \text{unreliability of any remaining original unit of the } k^{\text{th}} \text{ type} \]
\[ F_{ik} = \text{unreliability of the } i^{\text{th}} \text{ individual replacement unit of the } k^{\text{th}} \text{ type} \]
\[ p_{ok} = \text{a posteriori failure probability of any original individual unit of the } k^{\text{th}} \text{ type} \]
\[ p_{ik} = \text{a posteriori failure probability of the } i^{\text{th}} \text{ replacement individual unit of the } k^{\text{th}} \text{ type} \]
\[ t_{ik} = \text{time at which the } i^{\text{th}} \text{ individual unit of the } k^{\text{th}} \text{ type began operation} \]
\[ t = \text{time of operation of the system} \]

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

\[ N_o = \sum_{k=1}^{J} \left( N_k(t) + M_k(t) \right) \]

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

\[ \text{ETNF}(t) = \frac{1}{\left( \sum_{k=1}^{J} \frac{N_k(t)f_{ok}(t)}{1 - F_{ok}(t)} + \sum_{k=1}^{J} \sum_{i=1}^{M_k} \frac{f_{ik}(t - t_{ik})}{1 - F_{ik}(t - t_{ik})} \right)} \tag{8} \]
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 \( \Delta t \),

\[
t_{ik} = t - n \Delta t
\]

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

\[
f_{ok}(nat) \bigg/ \left( 1 - F_{ok}(nat) \right)
\]

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 \( \Delta t \) 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 \( q_{ik} = 1 - p_{ik} \) 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

\[
p_{ok}(t) = \int_{t}^{t+\Delta t} \frac{f_{ok}(t')}{1 - F_{ok}(t')} dt'.
\]

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

\[
Q_{s} = \prod_{k=1}^{J} \left( \prod_{j=1}^{N_k} (1 - p_{ok}) \cdot \prod_{i=1}^{N_k} (1 - p_{ik}) \right).
\]
so the probability that a failure occurred is

\[ P(t) = \min \left\{ \left[ 1 - \prod_{k=1}^{J} \left( 1 - p_{\text{ok}}^{k} \right) \cdot \prod_{i=1}^{N} \left( 1 - p_{i}^{k} \right) \right]^{j-1}, 1 \right\} \quad (10) \]

Given the probability of failure during the time step \( \Delta t \), 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

\[ P(t) = \sum_{i=1}^{N} p_{i}^{1} \quad (11) \]

where the \( p_{i} \) are of the form

\[ p_{i}^{1} = p_{i} - 1/2 p_{i} \sum_{j \neq i} p_{j} + 1/3 p_{i} \sum_{j \neq i, k \neq i} p_{j} p_{k} + \cdots \]

\[ = p_{i} \cdot A_{i} \]

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
The use of Eq. (12) can also be justified by assuming that $\Delta t$ 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 $\Delta t$ 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

$$1 - \sum_{k=1}^{J} \left( \frac{N_k p_{ok}^* + \sum_{i=1}^{k} p_{ik}^*}{M_k} \right).$$

(12)

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
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 ad-
ditional GOTO branch is then required in the computed GOTO statements in sub-
routines 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.\textsuperscript{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

\[
f(t) = \begin{cases} 
0 & \text{if } t < \beta \\
\lambda e^{-\lambda(t-\beta)} & \text{if } t \geq \beta 
\end{cases}
\]

The parameter $\beta$ is the "guarantee" time. The failure rate is a constant

\[
r(t) = \begin{cases} 
0 & \text{if } t < \beta \\
\lambda & \text{if } t \geq \beta 
\end{cases}
\]

and the a posteriori failure probability in the interval $\Delta t$ is

\[
p(t) = \begin{cases} 
0 & \text{if } t < \beta \\
1 - e^{-\lambda \Delta t} & \text{if } t \geq \beta 
\end{cases}
\]
B. Weibull Distribution

The Weibull density function is

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

Here \(\gamma\) represents the guarantee time. For \(t = \gamma\), the value of \(f\) depends on \(\beta\). We have

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

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

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

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

\[
p(t) = \begin{cases} 
0 & t < \gamma \\
1 - e^{-\left(\frac{t}{\Delta t}\right)^{\beta} - (t - \gamma)^{\beta}} & t \geq \gamma 
\end{cases}
\]
C. Normal Distribution (Truncated Normal)

Two forms of the normal distribution are commonly used in reliability computations: the standard normal and the truncated normal. The density function of each is of the form

\[ f(t) = \left( \frac{C}{\sqrt{2\pi}} \right) \exp \left[ -\frac{(t - \mu)^2}{2\sigma^2} \right] \]

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

\[ r(t) = \frac{\left( \frac{\sqrt{2\pi}}{\sigma} \right) \exp \left[ -\frac{(t - \mu)^2}{2\sigma^2} \right]}{\text{erfc} \left[ \frac{(t - \mu)}{\sigma\sqrt{2}} \right]} \]

and

\[ p(t) = \frac{\sqrt{2\pi} \int_t^{t+At} \exp \left[ -\frac{(x - \mu)^2}{2\sigma^2} \right] \, dx}{\text{erfc} \left[ \frac{(t - \mu)}{\sigma\sqrt{2}} \right]} \]

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 \( At \). 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. We use a three-parameter distribution which includes a guaranteed life. The density function is
which reduces to the standard two-parameter distribution when \( \gamma = 0 \).

The failure rate is given by

\[
\lambda(t) = \begin{cases} 
0 & t \leq \gamma \\
\frac{1}{(t - \gamma) \beta \sqrt{2\pi}} \exp \left\{ - \left( \frac{\ln(t - \gamma) - \alpha}{2\beta^2} \right)^2 \right\} & t > \gamma 
\end{cases}
\]

and the a posteriori failure probability by

\[
p(t) = \begin{cases} 
0 & t < \gamma \\
\frac{\sqrt{2/\pi}}{\beta \text{erfc}\left( \frac{\ln(t - \gamma) - \alpha}{\beta \sqrt{2}} \right)} \int_{t}^{t + \Delta t} \frac{1}{(t' - \gamma)} e^{-\left( \frac{\ln(t' - \gamma) - \alpha}{2\beta^2} \right)^2} dt' & t > \gamma \\
\frac{1}{(1/2\beta) \sqrt{2/\pi}} \int_{t}^{t + \Delta t} \frac{1}{(t' - \gamma)} e^{-\left( \frac{\ln(t' - \gamma) - \alpha}{2\beta^2} \right)^2} dt' & t = \gamma 
\end{cases}
\]

A 41-point Simpson’s rule is also used to find this integral. In this case we also assume that \( \gamma \) is an integral multiple of \( \Delta t \).

E. Gamma Distribution

The gamma distribution in its three-parameter form has the density function,

\[
f(t) = \begin{cases} 
0 & t - \gamma < 0 \\
\frac{\alpha \left( \alpha(t - \gamma) \right)^{\beta - 1} e^{-\alpha(t - \gamma)}}{\Gamma(\beta)} & t - \gamma \geq 0 
\end{cases}
\]
The exponential and Erlang distributions are special cases of this distribution. The failure rate is given by

\[
x(t) = \begin{cases} 
0 & t - \gamma \leq 0 \\
\frac{\alpha \{ \alpha (t - \gamma) \}^{\beta - 1} e^{-\alpha (t - \gamma)}}{\Gamma(\beta, \alpha (t - \gamma))} & t - \gamma > 0 
\end{cases}
\]

and the a posteriori failure probability by

\[
p(t) = \begin{cases} 
0 & t - \gamma \leq 0 \\
\frac{\alpha \int_{t}^{t+\Delta t} \{ \alpha (t' - \gamma) \}^{\beta - 1} e^{-\alpha (t' - \gamma)} \, dt'}{\Gamma(\beta, \alpha (t - \gamma))} & t - \gamma > 0 
\end{cases}
\]

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

\[
\Gamma(\beta, u) = \int_{u}^{\infty} x^{\beta - 1} e^{-x} \, dx
\]

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 \( \Delta t \) is small. A 41-point Simpson’s rule is used instead and, as in the log normal case, \( \gamma \) is assumed to be an integral multiple of \( \Delta t \).

**F. Uniform Distribution**

The uniform distribution has the density function

\[
f(t) = \begin{cases} 
0 & t < a \text{ and } t \geq \beta \\
\frac{1}{\beta - a} & a \leq t < \beta 
\end{cases}
\]
The failure rate is

\[ r(t) = \begin{cases} 
0 & \text{if } t < \alpha \text{ and } t \geq \beta \\
\frac{1}{\beta - t} & \text{if } \alpha \leq t < \beta \\
1 & \text{if } t \geq \beta 
\end{cases} \]

and the a posteriori failure probability is

\[ p(t) = \begin{cases} 
0 & \text{if } t < \alpha \\
\frac{At}{\beta - t} & \text{if } \alpha \leq t < \beta \\
1 & \text{if } t \geq \beta 
\end{cases} \]

G. Rayleigh Distribution

The Rayleigh distribution has the density \(^6\)

\[ f(t) = \begin{cases} 
0 & \text{if } t \geq t_0 \\
\frac{(t-t_0)^2}{\sigma^2} & \text{if } t_0 \leq t < \infty 
\end{cases} \]

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

\[ \alpha = 2\sigma^2, \quad \beta = 2, \quad T = t_0. \]

To input a Rayleigh component type to the program, the first parameter is \( \sigma \) and the second parameter is \( t_0 \). 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.

<table>
<thead>
<tr>
<th>Subroutine</th>
<th>Function</th>
</tr>
</thead>
</table>
| EXPECT     | DRIVER FOR PROGRAM  
The program calls SETUP and initializes certain variables. A loop on the time step $\Delta t$ 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.

<table>
<thead>
<tr>
<th>Subroutine</th>
<th>Entry</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>PEXPON</td>
<td>PPEXPON</td>
<td>Used for components following exponential distributions.</td>
</tr>
<tr>
<td>PWEIB</td>
<td>PPWEIB</td>
<td>Used for components following Weibull distributions.</td>
</tr>
<tr>
<td>PNORM</td>
<td>PPNORM</td>
<td>Used for components following normal distributions.</td>
</tr>
<tr>
<td>PLNORM</td>
<td>PPLNORM</td>
<td>Used for components following log normal distributions.</td>
</tr>
<tr>
<td>PGAMMA</td>
<td>PPGAMMA</td>
<td>Used for components following gamma distribution.</td>
</tr>
<tr>
<td>PUNIF</td>
<td>PPUNIF</td>
<td>Used for components following uniform distributions.</td>
</tr>
</tbody>
</table>

The following subroutines are used to compute integrals.

<table>
<thead>
<tr>
<th>Subroutine</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERK</td>
<td>LOGERK</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Subroutine</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>GAMPROB</td>
<td></td>
</tr>
</tbody>
</table>

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 \( \Delta t \) 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 \( 1000 \cdot \Delta t \cdot \text{MSP} \) unless the program storage is modified.

37-48 Time step, \( \Delta t \).

**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).
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

An integer giving the number of components in the group.

α - First distribution parameter

β - Second distribution parameter

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 dis-
tribution 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

2. Igor Bazovsky, Reliability Theory and Practice (Prentice Hall, Inc., Englewood

York, 1965).


APPENDIX A

FORTRAN LISTING OF XPECT PROGRAM
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
2---WIENULL DISTRIBUTION
3---NORMAL DISTRIBUTION
4---LOG NORMAL DISTRIBUTION
5---GAMMA DISTRIBUTION
6---UNIFORM DISTRIBUTION
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--FORMATE12.6
C TIME STEP--FORMATE12.6
C FOR EACH TYPE THE FOLLOWING DATA
C CARD 1--NAME OF COMPONENT
CARD 2--COMPONENT DISTRIBUTION TYPE (ITYPE(J)) FORMAT I12
CARD 3--NUMBER OF COMPONENTS OF TYPE (NORIG(J)) FORMAT I12
1ST DISTRIBUTION PARAMETER (ALPHA(J)) FORMAT E12.6
2ND DISTRIBUTION PARAMETER (BETA(J)) FORMAT E12.6
3ED DISTRIBUTION PARAMETER (PANNA(J)) FORMAT E12.6

C IN CASES WHERE ONLY 1 OR 2 PARAMETERS ARE USED LEAVE SPACE BLANK
COMMON /XP1/ TYPE(9,10), TITLE(8), XPECT(1001), X(1001), LABELY(3)
COMMON /XP2/ ALPHAT(10), BETA(10), GAMMA(10), RETNF(10)
COMMON /XP3/ NORG(10), REPLAC(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, IPILOT, LASTSHOT, NREP(10)
DATA LABELX/10HTIME /
DATA LABELY/29XEXPECTED 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(I)=ETNF
X(I)=0.
C START LOOP ON SHOTS
DO 6 NSHOT=1,LASTSHOT
SHOTS=FLOAT(NSHOT)*TDELTA
CALL DIDITFL (IFAIL)
GO TO (2,1,11), IFAIL
C A FAILURE OCCURRED BRANCH TO ROUTINE TO DECIDE WHICH TYPE FAILED
C CALL FAILURE
C ADD A SHOT TO ALL REPLACEMENT UNITS
C DO 4 J=1,NGROUPS
KSTOP=IREPL(J)
IF (KSTOP.EQ.0) GO TO 4
IF (KSTOP.GT.1000) GO TO 7
DO 3 K=1,KSTOP
IREPL(K,J)=IREPL(K,J)+1
C CONTINUE
IF (MSP.EQ.1) GO TO 5
IPRINT=NSHOT+1
IF (MOD(IPRINT,MSP).NE.1) GO TO 6
CALL CETNF (ETNF, IREASON)
KPRINT=KPRINT+1
X(KPRINT)=SHOTS
EXPECT(KPRINT)=ETNF
IF (IREASON.EQ.2) PRINT 14
PRINT 13, SHOTS, ETNF
CONTINUE
PRINT 18, ((NREP(I),I),I=1,NGROUPS)
GO TO 8
PRINT 16, J
C
IF (IPLT. NE. 1) GO TO 12
CALL PLOJB (X, EXPECT, KPRINT, 1, 0, 46, 0, 10, .6., TITLE, 80, LABELX, 10, LABY}
GO TO 16
PRINT 15
GO TO 12
CONTINUE
RETURN

FORMAT (1H, * AT TIME *, E13.6, * EXPECTED TIME TO NEXT FAILURE *)
1=*.E13.6)
FORMAT (1H, * FAILURE RATE IS ZERO SO ETNF WOULD BE INFINITE. ** RXPECT 
UN CONTINUES. *)
FORMAT (1H, * NUMBER OF POINTS DESIRED PLOTED GREATER THAN 1000. VXPECT 
VECTOR EXPECT HAS OVERFLOWED. NO PLOT MADE. *)
FORMAT (1HO, * NUMBER OF REPLACEMENTS OF COMPONENT TYPE *, I3, * EXCEPT *)
1EDS ALLOWED STORAGE** RUN TERMINATED. *)
FORMAT (1H, * RUN TERMINATED TO GIVE YOU TIME TO THINK. *)
FORMAT (1HO, //10(I5, *, UNITS OF GROUP*, I3, * WERE REPLACED */))
END

SUBROUTINE SETUP
COMMON /XP4/ TYPE(8, 10), TITLE(8), EXPECT(1001), X(1001), LABELY(3)
COMMON /XP3/ ALPHA(10), BETA(10), GAMMA(10), RETNF(10)
COMMON /XP3/ NOrig(10), NREPLAC(10), TREPL(1000, 10), PSUM, IPB, MISP
COMMON /XP3/ NGROUPS, IPB(10), P(1000, 10), IGROUP(10), PTEST(10)
COMMON /XP3/ TAST, TDELTA, NSHOT, SHOTS, IPLT, LASTH, 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, IPLT, IPB, TAST, TDELTA
IF=1
IPB=1
IF (IPLT. EQ. 1) IP=2
IF (IPB. EQ. 1) IPB=2
PRINT 7, NGROUPS, MSP, TAST, TDELTA, YN(IP), YN(IPB)
DO 1 I=1, NGROUPS
READ 8, (TYPE(J, I), J=1, 8)
DO 9 J=1, 8
IF (IGROUP(I, J). NE. 7) GO TO 3
IGROUP(I)=2
ALPHA(I)=2.*ALPHA(I)**2
BETA(I)=BETA(I)
GAMMA(I)=GAMMA(I)
SETUP 21
CONTINUE
1
ZERO THE REPLACEMENT VECTOR
DO 2 I=1, NGROUPS
NREP(I)=0
NREPLAC(I)=0
LASTH=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
BETA(I)=BETA(I)
CONTINUE
DO 4 I=1, NGROUPS
IF (IGROUP(I). NE. 4 AND IGROUP(I). NE. 5) GO TO 4
CONTINUE
DO 5 I=1, NGROUPS
IF (IGROUP(I). NE. 4 AND IGROUP(I). NE. 5) GO TO 4
CONTINUE
SUBROUTINE CETNF(ETNF, IREASON)

COMMON /XP2/ ALPHA(10), BETA(10), GAMMA(10), RETNF(10)
COMMON /XP3/ NORG(10), NREPLAC(10), IREPL(1000, 10), PSUM, IPROB, MSP
COMMON /XP4/ NGROUPS, P(1000, 10), IGROUP(10), PST(10), TST(10)
COMMON /XP5/ TLAST, TDELTA, NSHOT, SHOTS, IPLOT, LASTSHOT, NREP(10)
COMMON /XP6/ PS(10), PP(10)

C FORM THE SUM OF F(X) + [1 - INT(F(X)) FOR EACH COMPONENT OF EACH TYPE. CETNF
C EXPECTED NUMBER OF SHOTS TO NEXT FAILURE IS RECIPROCAL OF THIS SUM.CETNF
C IREASON = 1
DO 7 I=1, NGROUPS
   IWORK = IGROUP(I)
   CALL PEXF(I)
   GO TO 7
1 CALL PWEIB(I)
   GO TO 7
2 CALL PNORM(I)
   GO TO 7
3 CALL PLNORM(I)
   GO TO 7
4 CALL PGAMMA(I)
   GO TO 7
5 CALL PUNIFM(I)
   CONTINUE
6 CALL PUNIFM(I)
   CONTINUE
7 SUM = SUM + RETNF(I)
C SUM THE INDIVIDUAL FAILURE RATES AND TAKE RECIPROCAL
C SUM = 0
DO 8 I = 1, NGROUPS
   IF (SUM.EQ.0.) GO TO 9
   ETNF = 1./SUM
9 RETURN
C IREASON = 2
ETNF = 1.E+300
RETURN
END

SUBROUTINE DIDITFL(IFAIL)

COMMON /XP2/ ALPHA(10), BETA(10), GAMMA(10), RETNF(10)
COMMON /XP3/ NORG(10), NREPLAC(10), IREPL(1000, 10), PSUM, IPROB, MSP
COMMON /XP4/ NGROUPS, P(1000, 10), IGROUP(10), PST(10), TST(10)
COMMON /XP5/ TLAST, TDELTA, NSHOT, SHOTS, IPLOT, LASTSHOT, NREP(10)
COMMON /XP6/ PS(10), PP(10)

DIDITFL
LCM / XP7 / PSAVE(10000, 10)

FOR EACH COMPONENT TYPE COMPUTE THE PROBABILITY OF FAILURE

THIS PROBABILITY IS THE A POSTERIORI PROBABILITY SINCE ALL

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 PPPLNORM (I)

GO TO 7

5 CALL PPGAMMA (I)

GO TO 7

6 CALL PPUNIFFM (I)

CONTINUE

7

Y=RANDOM(DUMMY)

PSUM=1.-PPROB

IFAIL=1

IF (PSUM.GE.1.) IFAIL=3

IF (IFPROC.NE.1.) GO TO 10

PRINT PROBABILITIES IF DESIRED.

IF (MSP.EQ.1) GO TO 7

IF (MOD(IPRINT,MSP).NE.1) GO TO 10

PRINT 14, SHOTS

DO 9 IPRINT=1,NGROUPS

PRINT 16, IPRINT, PS(IPRINT)

9

IF (IFAIL.NE.3) GO TO 12

PRINT 15, NSHOT, SHOTS

DO 11 I=1,NGROUPS

10 PRINT 16, I, PS(I)

11 GO TO 13

12 IF (Y.LE.PSUM) IFAIL=2

13 RETURN

CALL NPUNIFFM(I)

14 FORMAT (1H * A POSTERIORI COMPONENT GROUP FAILURE PROBABILITY AT

TIME *,E13.6)

15 FORMAT (1H, 16* , 12, 12, E15.7)

16 FORMAT (1H, 16* , 12, 12, E15.7)

END

SUBROUTINE FAILURE

COMMON / XP3 / NORIG(10), NREPLAC(10), REPL(10000, 10), PSUM, IPRC, MSP

COMMON / XP4 / NGROUPS, P(10000, 10), IGROUP(10), PTEST(10)

COMMON / XP5 / LAST, TDELT, SHOTS, IPRC, LSTH, NREP(10)

FIND SUM OF INDIVIDUAL FAILURE PROBABILITIES FOR NORMALIZATION.

DURING SUMMATION FIND AND SAVE CONTRIBUTIONS OF EACH GROUP.

PTEST HOLDS TOTAL FOR GROUP. PROB HOLDS CONTRIBUTION OF ORIGINAL

UNITS AND P HOLDS CONTRIBUTION OF REPLACEMENT UNITS.

SUM=0

DO 1 I=1,NGROUPS

NTEMP=NORIG(I)-NREPLAC(I)

IF (NTEMP.LE.1.) PROB(I)=0.

PROB(I)=PROB(I)*FLOAT(NTEMP)

1 SUM=SUM+PROB(I)

DO 3 I=1,NGROUPS

ISTOP=NREPLAC(I)

GO TO 7

3 IF (ISTOP.EQ.0) GO TO 3

DO 2 J=1,ISTOP

SUM=SUM+P(J,I)

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

ENTRY PEXPON(I)
C FOR COMPONENTS FOLLOWING EXPONENTIAL STATISTICS.
C THE FAILURE RATE IS INDEPENDENT OF THE NUMBER OF SHOTS
C COMMON /XP2/ ALPHA(10),BETA(10),GAMMA(10),RETNF(10)
C COMMON /XP3/ NORIG(10),NREPLAC(10),IREPL(1000,10),PSUM,IPROR,MSP
C COMMON /XP4/ NGROUPS,PROB(10),P(1000,10),IGROUP(10),PTEST(10)
C COMMON /XP5/ TLAST,TDELTA,NSHOT,SHOTS,IPLOT,LASTSHI,NREP(10)
C COMMON /XP6/ PS(10),PPROR
C COMMON /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)
    IF (TEMP.LT.0.) RETNF(I)=RETNF(I)-ALPHA(I)
  1
  RETURN
  RETNF(I)=0.
  RETURN
C ENTRY PEXPON
C CALCULATE THE A POSTERIORI PROBABILITY
C INDEX=NREPLAC(I)
C TEMP=SHOTS-BETA(I)
  IF (TEMP.GE.0.) GO TO 3
  PROB(I)=0.
  PS(I)=0.
SUBROUTINE PWEIB(I)
FOR COMPONENTS FOLLOWING WEIBULL 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/ NOROUPS, PROB(10), P(1000,10), IGROUP(10), PTEST(10)
COMMON /XP5/ TLAST, TDELTA, NSHOT, SHOT$,IR, PROB(MSP)
COMMON /XP6/ PS(10), PPROB
INDEX=NREPLAC(I)
TEMP=SHOTS-GAMMA(I)
IF (TEMP.LT.0.) GO TO 4
IF (TEMP.EQ.0.) GO TO 5
NTEMP=NORIG(I)-INDEX
RETNF(I)=0.
IF (NTEMP.LT.1.) GO TO 1
RETNF(I)=BETA(I)*(TEMP**(BETA(I)-1.))*(NTEMP)/ALPHA(I)
RETURN
1 IF (INDEX.EQ.0.) RETURN
DO 3 J=1,INDEX
TEMP=WRITE(IREPL(J,1))**TDELTA-Gamma(I)
IF (TEMP.LT.0.) GO TO 3
IF (TEMP.EQ.0.) GO TO 2
RETNF(I)=RETNF(I)+BETA(I)*(TEMP**(BETA(I)-1.))/ALPHA(I)
GO TO 3
2 IF (BETA(I).LE.1.) GO TO 3
RETNF(I)=RETNF(I)+1./ALPHA(I)
CONTINUE
RETURN
C NUMBER OF SHOTS LESS THAN GAMMA NO FAILURES CAN OCCUR
4 RETNF(I)=0.
RETURN
C NUMBER OF SHOTS EQUALS GAMMA
5 IF (BETA(I).LE.1.) GO TO 6
RETNF(I)=0.
RETURN
6 IF (BETA(I).LT.1.) GO TO 7
RETNF(I)=1./ALPHA(I)
RETURN
7 PRINT 13
TEMP=.01
RETNF(I)=BETA(I)*(TEMP**(BETA(I)-1.))*(NORIG(I)-INDEX)/ALPHA(I)
RETURN
ENTRY PWEIB
C CALCULATE THE A POSTERIORI PROBABILITY
INDEX=NREPLAC(I)
TEMP=SHOTS-GAMMA(I)
IF (TEMP.LT.1.) GO TO 12
TEMP=((TEMP-TDELTA)**BETA(I)-TEMP**BETA(I))/ALPHA(I)
PROB(I)=1.-EXP(TEMP)
RETURN
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), REPL(1000, 10), PSUM, IPROB, MSP
COMMON /XP4/ NGROUPS, PROB(10), P(1000, 10), IGROUP(10), PTTEST(10)
COMMON /XP5/ TLAST, TDELTA, NSHOT, SHOTS, IPLOT, LASTSHOT, NREP(10)
COMMON /XP6/ PSAVE(10060, 10)
DATA C1, C2 /1.41421356, 0.7978845608/
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)
WTMP = NORIG(I) - INDEX
RETNF(I) = 0.
IF (NTMP .LT. 1) GO TO 1
RETNF(I) = FLOAT(NTMP) * DF / FBAR
1 IF (INDEX .EQ. 0) RETURN
DO 2 J = 1, INDEX
TEST = (FLOAT(REPL(J, I)) * TDELTA - ALPHA(I)) / (BETA(I) * C1)
FBAR = BETA(I) * ERFC(TEST)
DF = C2 * EXP(-TEST**2)
RETNF(I) = RETNF(I) + DF / FBAR
2 RETURN
C IF PROGRAM REACHES THIS POINT FAILURE IS VIRTUALLY CERTAIN
C WE ARBITRARILY SET RETNF=1.E+100 AND RETURN
3 RETNF(I) = 1.E+100
RETURN
C ENTRY PPROM
C CALCULATE THE A POSTERIORI PROBABILITY
INDEX = NREPLAC(I)
CALL ERK (PROB(I), ALPHA(I), BETA(I))
PSAVE(NSHORT, 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
PS(I) = 1. - PROB(I)
FORMAT (1HO /* FOR THE WEIBULL DISTRIBUTION BETA LESS THAN 1 AND TIME-GAMMA = 0 CAUSES THE FAILURE RATE TO APPROACH INFINITY. */ /* SINCE THE WEIBULL IS 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. */ /*)
END
SUBROUTINE PLNORM(I)
FOR COMPONENTS FOLLOWING LOG NORMAL STATISTICS
COMMON /XP2/ ALPHA(10),BETA(10),GAMMA(10),RETNF(10)
COMMON /XP3/ NORIG(10),NREPLAC(10),IREPL(10000,10),PSUM,IPROB,MSP
COMMON /XP4/ NGROUPS,PROB(10),P(10000,10),NGROUPS,PTEST(10)
COMMON /XP5/ TLAST,TDDELTA,NSHOT,NSHOTS,PLOT,LASTSHOT,NREP(10)
COMMON /XP6/ PS(10),PPROB
LCM /XP7/ PSAVE(10000,10)
DATA C1,C2/1.41421356,3731,0.79788456080286/
INDEX=NREPLAC(I)
RETNF(I)=0.
TEMP1=SHOTS-GAMMA(I)
IF (TEMP1.LE.0.) RETURN
D=1./(C1*BETA(I))
TEMP2=(ALOG(TEMP1)-ALPHA(I))*D
IF (TEMP2.GE.26.) GO TO 3
NTEMP=NORIG(I)-INDEX
IF (NTEMP.LT.1) GO TO 1
RETNF(I)=C2*EXP(-TEMP2**2)/(TEMP1*BETA(I)*ERFC(TEMP2))
RETNF(I)=FLOAT(NTEMP)*RETNF(I)
IF (INDEX.EQ.0) RETURN
DO 2 J=1 INDEX
TEMP1=FLOAT(IREPL(J,I))*TDDELTA-GAMMA(I)
IF (TEMP1.LE.0.) GO TO 2
TEMP2=(ALOG(TEMP1)-ALPHA(I))*D
RETNF(I)=RETNF(I)+C2*EXP(-TEMP2**2)/(TEMP1*BETA(I)*ERFC(TEMP2))
1 CONTINUE
RETURN
C IF TEMP2.GE.26. ERFC WILL UNDERFLOW. ARBITRARILY SET
C RETNF(I)=1.E+100 AND RETURN
3 RETNF(I)=1.E+100 AND RETURN
RETURN
ENTRY PPLNORM
C CALCULATE THE A POSTERIORI PROBABILITY
INDEX=NREPLAC(I)
CALL LOGERK (PROB(I),ALPHA(I),BETA(I),GAMMA(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
PS(I)=PS(I)*(1.-PROB(I))
4 CONTINUE
IF (INDEX.EQ.0) GO TO 7
DO 6 J=1,INDEX
K=IREPL(J,I)+1
P(J,I)=PSAVE(K,I)
PS(I)=PS(I)*(1.-P(J,I))
6 CONTINUE
PPROB=PPROB*PS(I)
PS(I)=1.-PS(I)
RETURN
END
SUBROUTINE PGAMMA(I)
FOR COMPONENTS FOLLOWING GAMMA DISTRIBUTIONS
GAMMA IN COMMON/XP2/ HAS BEEN CHANGED TO ZAMMA TO ALLOW THE USE 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,MSPOR
COMMON /XP4/ NGROUPS,PROB(10),P(1000,10),IGROUP(10),IFTES(10)
COMMON /XP5/ TLAST,TDDELTA,NSHOT,SHOTS,IPLOT,LASTSHT,NREP(10)
COMMON /XP6/ NCPOE(10),PPROB
LCM /XP7/ PSAVE(10000,10)
INDEX=NREPLAC(I)
U=ALPHA(I)*(SHOTS-ZAMMA(I))
RETNF(I)=0.
IF (UCLE.0.0) RETURN
NTEMP=NORIG(I)-INDEX
IF (NTEMP.LT.1) GO TO 1
RETNF(I)=ALPHA(I)*U**(BETA(I)-1.)*EXP(-U)
RETNF(I)=RETNF(I)+RATE
CONTINUE
RETURN
ENTRY PPGAMMA
CALCULATE THE A POSTERIORI PROBABILI
INDEX=NREPLAC(I)
IF (SHOTS-ZAMMA(I).LT.0.0) 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
IF (MULTTO.EQ.O GO TO 4
DO 3 J=2,MULTTO
PS(I)=PS(I)*(1.-PROB(I))
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)+I
PS(I)=PS(I)*(1.-P(J,I))
5 PS(I)=PS(I)*(1.-P(J.I))
6 PPPO=PPOB*P(I)
PS(I)=1.-PS(I)
RETURN
PROBABILITY OF FAILURE IS ZERO, SHOTS LESS THAN GAMMA
PROB(I)=0.
PSAVE(NSHOT,I)=0.
PS(I)=0.
RETURN
END

SUBROUTINE PUNIFM(I)
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,MSPOR
COMMON /XP4/ NGROUPS,PROB(10),P(1000,10),IGROUP(10),IFTES(10)
COMMON /XP5/ TLAST,TDDELTA,NSHOT,SHOTS,IPLOT,LASTSHT,NREP(10)
COMMON /XP6/ NCPOE(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)
SUBROUTINE ERK(P, ALPHA, BETA, GAMMA)
  ERK COMPUTES THE A POSTERIORI FAILURE PROBABILITY
  FOR A NORMALLY DISTRIBUTED COMPONENT. IT COMPUTES THE INTEGRAL
  OF THE DISTRIBUTION FUNCTION FROM SHOT N-1 TO SHOT N USING A 41
  POINT SIMPSONS RULE.
  ENTRY LOGERK DOES THE SAME FOR A LOG NORMAL COMPONENT.
  COMMON /XP5/ TLAST, TDELTA, NSHOT, SHOTS, IPLOT, LASTSH, NREP(10)
  DATA C1, C2 /1.4142135623731, 0.797884568062867/
  B=SHOTS-ALPHA
  A=B-TDELTA
  STEP=.025*TDELTA
  Q=-.5/(BETA**2)
  P=EXP(Q*A*A)
  DO 1 I=2, 40, 2
       P=P*EXP(Q*(A+FLOAT(I-1)*STEP)**2)+2.*EXP(Q*(A+FLOAT(I)*STEP))
  1 P=P-EXP(Q**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((-ALPHA-ALPHA)**2*Q)/A
  DO 3 I=2, 40, 2
       X=I*A+FLOAT(I-1)*STEP
       P=P*EXP(-Q*X)**2
  3 CONTINUE
  RETURN
END
X2=A+FLOAT(I)*STEP
P=P+4.*EXP(-(ALOG(X1)-ALPHA)**2/Q)/X1+2.*EXP(-(ALOG(X2)-ALPHA)**2/Q)/X2

P=P-EXP(-(ALOG(B)-ALPHA)**2/Q)/B
IF (A.EQ.0.) GO TO 4
P=P*STEP*C2/(3.*BETA*ERFC((ALOG(A)-ALPHA)/(BETA*C1)))
RETURN

IF A.EQ.0.) GO TO 4
RETURN

SUBROUTINE GAMPROB(ALPHA,BETA,ZAMMA,PROB)
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.

<table>
<thead>
<tr>
<th>TEST PROBLEM</th>
<th>7</th>
<th>40</th>
<th>1</th>
<th>2000</th>
<th>0.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXPONENTIAL</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WIEBULL</td>
<td>2</td>
<td></td>
<td></td>
<td>1000</td>
<td>0.75</td>
</tr>
<tr>
<td>NORMAL</td>
<td>3</td>
<td></td>
<td></td>
<td>130</td>
<td>0.0</td>
</tr>
<tr>
<td>LOG NORMAL</td>
<td>4</td>
<td></td>
<td></td>
<td>180</td>
<td>-3.7</td>
</tr>
<tr>
<td>GAMMA</td>
<td>5</td>
<td></td>
<td></td>
<td>100</td>
<td>-5.2</td>
</tr>
<tr>
<td>UNIFORM</td>
<td>6</td>
<td></td>
<td></td>
<td>1000</td>
<td></td>
</tr>
<tr>
<td>RAYLEIGH</td>
<td>7</td>
<td></td>
<td></td>
<td>2000</td>
<td></td>
</tr>
</tbody>
</table>

Fig. B-1.
Input to program.
**TEST PROBLEM**

<table>
<thead>
<tr>
<th>Component</th>
<th>Type</th>
<th>Distribution Type Number</th>
<th>Number of Units</th>
<th>Alpha</th>
<th>Beta</th>
<th>Gamma</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>EXPONENTIAL</td>
<td>1</td>
<td>100</td>
<td>0.80000E-05</td>
<td>1.00000E+03</td>
<td>Gamma = 0.00000E+00</td>
</tr>
<tr>
<td>2</td>
<td>WEIHULL</td>
<td>2</td>
<td>100</td>
<td>0.70000E-05</td>
<td>7.50000E+00</td>
<td>Gamma = 0.00000E+00</td>
</tr>
<tr>
<td>3</td>
<td>NORMAL</td>
<td>3</td>
<td>100</td>
<td>0.25000E-04</td>
<td>4.00000E+03</td>
<td>Gamma = 0.00000E+00</td>
</tr>
<tr>
<td>4</td>
<td>LOG-NORMAL</td>
<td>4</td>
<td>100</td>
<td>0.15000E+02</td>
<td>5.00000E+02</td>
<td>3.70000E+01</td>
</tr>
<tr>
<td>5</td>
<td>GAMMA</td>
<td>5</td>
<td>100</td>
<td>1.00000E-01</td>
<td>2.00000E+02</td>
<td>5.20000E+01</td>
</tr>
<tr>
<td>6</td>
<td>UNIFORM</td>
<td>6</td>
<td>100</td>
<td>0.50000E+03</td>
<td>4.00000E+06</td>
<td>Gamma = 0.00000E+00</td>
</tr>
<tr>
<td>7</td>
<td>RAYLEIGH</td>
<td>7</td>
<td>200</td>
<td>1.00000E+05</td>
<td>4.00000E+03</td>
<td>Gamma = 0.00000E+00</td>
</tr>
</tbody>
</table>

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.
SINCER 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.

<table>
<thead>
<tr>
<th>AT TIME</th>
<th>EXPECTED TIME TO NEXT FAILURE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2.3333E+01</td>
</tr>
<tr>
<td>200000E-02</td>
<td>1.6191E+02</td>
</tr>
<tr>
<td>400000E-02</td>
<td>2.8650E+02</td>
</tr>
<tr>
<td>600000E-02</td>
<td>4.3714E+02</td>
</tr>
<tr>
<td>800000E-02</td>
<td>4.4717E+02</td>
</tr>
<tr>
<td>1000000E-03</td>
<td>3.9675E+02</td>
</tr>
<tr>
<td>1200000E-03</td>
<td>4.4077E+02</td>
</tr>
<tr>
<td>1400000E-03</td>
<td>4.7924E+02</td>
</tr>
<tr>
<td>1600000E-03</td>
<td>5.1217E+02</td>
</tr>
<tr>
<td>1800000E-03</td>
<td>5.2071E+02</td>
</tr>
<tr>
<td>2000000E-03</td>
<td>5.5767E+02</td>
</tr>
<tr>
<td>2200000E-03</td>
<td>6.0597E+02</td>
</tr>
<tr>
<td>2400000E-03</td>
<td>6.3294E+02</td>
</tr>
<tr>
<td>2600000E-03</td>
<td>6.5992E+02</td>
</tr>
<tr>
<td>2800000E-03</td>
<td>7.3274E+02</td>
</tr>
<tr>
<td>3000000E-03</td>
<td>8.8625E+02</td>
</tr>
<tr>
<td>3200000E-03</td>
<td>6.9937E+02</td>
</tr>
<tr>
<td>3400000E-03</td>
<td>7.7123E+02</td>
</tr>
<tr>
<td>3600000E-03</td>
<td>7.2234E+02</td>
</tr>
<tr>
<td>3800000E-03</td>
<td>7.3684E+02</td>
</tr>
<tr>
<td>4000000E-03</td>
<td>7.7544E+02</td>
</tr>
<tr>
<td>4200000E-03</td>
<td>7.6351E+02</td>
</tr>
<tr>
<td>4400000E-03</td>
<td>7.6382E+02</td>
</tr>
<tr>
<td>4600000E-03</td>
<td>6.4231E+02</td>
</tr>
<tr>
<td>4800000E-03</td>
<td>6.6546E+02</td>
</tr>
<tr>
<td>5000000E-03</td>
<td>6.7499E+02</td>
</tr>
<tr>
<td>5200000E-03</td>
<td>6.8529E+02</td>
</tr>
<tr>
<td>5400000E-03</td>
<td>6.6521E+02</td>
</tr>
<tr>
<td>5600000E-03</td>
<td>6.7694E+02</td>
</tr>
<tr>
<td>5800000E-03</td>
<td>6.5479E+02</td>
</tr>
<tr>
<td>6000000E-03</td>
<td>6.5407E+02</td>
</tr>
<tr>
<td>6200000E-03</td>
<td>6.6394E+02</td>
</tr>
<tr>
<td>6400000E-03</td>
<td>6.6429E+02</td>
</tr>
<tr>
<td>6600000E-03</td>
<td>6.4239E+02</td>
</tr>
<tr>
<td>6800000E-03</td>
<td>6.4469E+02</td>
</tr>
<tr>
<td>7000000E-03</td>
<td>6.4516E+02</td>
</tr>
<tr>
<td>7200000E-03</td>
<td>6.6424E+02</td>
</tr>
<tr>
<td>7400000E-03</td>
<td>6.4826E+02</td>
</tr>
<tr>
<td>7600000E-03</td>
<td>6.4824E+02</td>
</tr>
<tr>
<td>7800000E-03</td>
<td>6.6250E+02</td>
</tr>
<tr>
<td>8000000E-03</td>
<td>6.6250E+02</td>
</tr>
<tr>
<td>8200000E-03</td>
<td>6.8025E+02</td>
</tr>
<tr>
<td>8400000E-03</td>
<td>6.6248E+02</td>
</tr>
<tr>
<td>8600000E-03</td>
<td>6.6250E+02</td>
</tr>
<tr>
<td>8800000E-03</td>
<td>6.6157E+02</td>
</tr>
<tr>
<td>9000000E-03</td>
<td>6.6053E+02</td>
</tr>
<tr>
<td>9200000E-03</td>
<td>6.5932E+02</td>
</tr>
<tr>
<td>9400000E-03</td>
<td>6.5798E+02</td>
</tr>
<tr>
<td>9600000E-03</td>
<td>6.5627E+02</td>
</tr>
<tr>
<td>9800000E-03</td>
<td>6.5452E+02</td>
</tr>
<tr>
<td>10000000E-04</td>
<td>6.5257E+02</td>
</tr>
<tr>
<td>10200000E-04</td>
<td>6.5098E+02</td>
</tr>
<tr>
<td>10400000E-04</td>
<td>6.4846E+02</td>
</tr>
<tr>
<td>10600000E-04</td>
<td>6.4688E+02</td>
</tr>
</tbody>
</table>

Fig. B-3. 
Program output page 2.
<table>
<thead>
<tr>
<th>AT TIME</th>
<th>EXPECTED TIME TO NEXT FAILURE</th>
</tr>
</thead>
<tbody>
<tr>
<td>.100000E+04</td>
<td>.441336E+02</td>
</tr>
<tr>
<td>.110000E+04</td>
<td>.415066E+02</td>
</tr>
<tr>
<td>.117000E+04</td>
<td>.395107E+02</td>
</tr>
<tr>
<td>.114000E+04</td>
<td>.373391E+02</td>
</tr>
<tr>
<td>.116000E+04</td>
<td>.351492E+02</td>
</tr>
<tr>
<td>.118000E+04</td>
<td>.331969E+02</td>
</tr>
<tr>
<td>.120000E+04</td>
<td>.313187E+02</td>
</tr>
<tr>
<td>.122000E+04</td>
<td>.295422E+02</td>
</tr>
<tr>
<td>.124000E+04</td>
<td>.276949E+02</td>
</tr>
<tr>
<td>.126000E+04</td>
<td>.261159E+02</td>
</tr>
<tr>
<td>.128000E+04</td>
<td>.246645E+02</td>
</tr>
<tr>
<td>.130000E+04</td>
<td>.233198E+02</td>
</tr>
<tr>
<td>.132000E+04</td>
<td>.208753E+02</td>
</tr>
<tr>
<td>.134000E+04</td>
<td>.208252E+02</td>
</tr>
<tr>
<td>.136000E+04</td>
<td>.198636E+02</td>
</tr>
<tr>
<td>.138000E+04</td>
<td>.179567E+02</td>
</tr>
<tr>
<td>.140000E+04</td>
<td>.162486E+02</td>
</tr>
<tr>
<td>.142000E+04</td>
<td>.156193E+02</td>
</tr>
<tr>
<td>.144000E+04</td>
<td>.156547E+02</td>
</tr>
<tr>
<td>.146000E+04</td>
<td>.147956E+02</td>
</tr>
<tr>
<td>.148000E+04</td>
<td>.139833E+02</td>
</tr>
<tr>
<td>.150000E+04</td>
<td>.132235E+02</td>
</tr>
<tr>
<td>.152000E+04</td>
<td>.126253E+02</td>
</tr>
<tr>
<td>.154000E+04</td>
<td>.120678E+02</td>
</tr>
<tr>
<td>.156000E+04</td>
<td>.115483E+02</td>
</tr>
<tr>
<td>.158000E+04</td>
<td>.110655E+02</td>
</tr>
<tr>
<td>.160000E+04</td>
<td>.106125E+02</td>
</tr>
<tr>
<td>.162000E+04</td>
<td>.103333E+02</td>
</tr>
<tr>
<td>.164000E+04</td>
<td>.100834E+02</td>
</tr>
<tr>
<td>.166000E+04</td>
<td>.994125E+01</td>
</tr>
<tr>
<td>.168000E+04</td>
<td>.980429E+01</td>
</tr>
<tr>
<td>.170000E+04</td>
<td>.965070E+01</td>
</tr>
<tr>
<td>.172000E+04</td>
<td>.921319E+01</td>
</tr>
<tr>
<td>.174000E+04</td>
<td>.889252E+01</td>
</tr>
<tr>
<td>.176000E+04</td>
<td>.856299E+01</td>
</tr>
<tr>
<td>.178000E+04</td>
<td>.823140E+01</td>
</tr>
<tr>
<td>.180000E+04</td>
<td>.790087E+01</td>
</tr>
<tr>
<td>.182000E+04</td>
<td>.757035E+01</td>
</tr>
<tr>
<td>.184000E+04</td>
<td>.723983E+01</td>
</tr>
<tr>
<td>.186000E+04</td>
<td>.690931E+01</td>
</tr>
<tr>
<td>.188000E+04</td>
<td>.657880E+01</td>
</tr>
<tr>
<td>.190000E+04</td>
<td>.624829E+01</td>
</tr>
<tr>
<td>.192000E+04</td>
<td>.591788E+01</td>
</tr>
<tr>
<td>.194000E+04</td>
<td>.558747E+01</td>
</tr>
<tr>
<td>.196000E+04</td>
<td>.525706E+01</td>
</tr>
<tr>
<td>.198000E+04</td>
<td>.492665E+01</td>
</tr>
<tr>
<td>.200000E+04</td>
<td>.459624E+01</td>
</tr>
</tbody>
</table>

Fig. B-4.
Program output page 3.

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-5.
Film output.