EGG-SSRE-9017
July 1990

TECHNICAL REPORT

User's Guide to PHAZE, a Computer Program for Parametric Hazard Function Estimation

Corwin L. Atwood

Idaho National Engineering Laboratory

Managed by the U.S. Department of Energy

EG&G Idaho

Work performed under DOE Contract No. DE-AC07-76ID01570

Prepared for the U.S. NUCLEAR REGULATORY COMMISSION

EGG-SSRE-9017

User's Guide to PHAZE, a Computer Program for Parametric Hazard Function Estimation

Corwin L. Atwood

Published July 1990

Idaho National Engineering Laboratory
EG&G Idaho
Idaho Falls, ID 83415-3421

Prepared for the
U.S. Nuclear Regulatory Commission
Washington, DC 20555
Under DOE Contract No. DE-AC07-76IDO1570

ABSTRACT

The program PHAZE (for Parametric HAZard function Estimation) performs statistical inference on a hazard function, based on reported failure times of components that are repaired and restored to service. The inference includes parameter estimation, testing of hypotheses, and checking of the model assumptions, under a choice of parametric models. This user's guide sketches only enough of the theory so that PHAZE can be used, with a full presentation of the theory being given in a companion report. A typical PHAZE session is described. The format of a data file is given, and all the PHAZE commands are listed and explained. The program has been verified and validated, and this work is summarized. Finally, some of the technical details are given, of interest to statisticians and programmers. An appendix shows an entire PHAZE session, both the user's commands and the program's responses.

FIN No. A6389—Aging Components and Systems IV:
Risk Evaluation and Aging Phenomena

SUMMARY

The program PHAZE (for Parametric HAZard function Estimation) performs statistical inference on a hazard function (also called a failure rate or intensity function), based on reported failure times of components that are repaired and restored to service. Three parametric models are allowed, the exponential, linear, and Weibull hazard models. The inference includes estimation (maximum likelihood estimators and confidence regions) of the parameters and of the hazard function itself, testing of hypotheses such as of increasing failure rate, and checking of the model assumptions.

This user's guide sketches only enough of the theory so that PHAZE can be used, with a full presentation of the theory being given in a companion report. A typical PHAZE session is described. This consists of an initial exploratory phase, in which the various model assumptions are checked, and a final estimation phase, in which the maximum likelihood estimator and a confidence interval are found for the hazard function at time(s) of interest. The format of a data file is given, with examples. PHAZE is an interactive command-based program; all the PHAZE commands are therefore listed and explained.

The program has been verified and validated, and this work is summarized. Finally, some of the technical details are given, of interest to statisticians and programmers. An appendix shows an entire PHAZE session, both the user's commands and the program's responses. This appendix illustrates virtually all of the PHAZE commands, and the resulting output.

ACKNOWLEDGMENTS

W. Scott Roesener used the program through many intermediate versions, discovering problems for me and showing patience while I corrected them. Andrew J. Wolford suggested features and options.

CONTENTS

ABSTRACT ii
SUMMARY iii
ACKNOWLEDGMENTS iv
1. INTRODUCTION 1
   1.1 Fundamental Assumptions 1
   1.2 Full and Conditional Likelihood 3
   1.3 Typical PHAZE Session 3
   1.4 Terminology and Formats 4
2. DATA FILES 5
3. PHAZE COMMANDS 9
   3.1 Command Syntax 9
   3.2 General Concepts 11
   3.3 Commands 12
4. VERIFICATION AND VALIDATION 28
   4.1 Overview 28
   4.2 Verification 30
   4.3 Validation 30
5. TECHNICAL DETAILS 33
   5.1 Language and Portability 33
   5.2 Calculational Methods 33
   5.3 Some Details of the Coding 37
6. REFERENCES 39
APPENDIX—EXAMPLE PHAZE SESSION A-1

TABLES

1. Summary of commands 10
2. Effect of switches on CI command 13

FIGURES

1. Data file with times entered as dates 7
2. Data file with single component, integer failure times 7
3. Data file with floating point failure times 7
4. Example output from CC B command 15
5. 90% confidence regions for (β, λ_0), based on exponential hazard model 19

6. 90% confidence regions for (β, λ_0), based on Weibull hazard model 19
7. 90% confidence regions for (β, λ_0), based on linear hazard model, uncentered data 20
8. 90% confidence regions for (β, λ_0), based on linear hazard model, centered data 20
9. Q-Q plot, based on exponential hazard model 25

User's Guide to PHAZE, a Computer Program for Parametric Hazard Function Estimation

1. INTRODUCTION

The program PHAZE carries out Parametric HAZard function Estimation, based on reported failure times of components that are repaired and restored to service. It is an interactive program, intended for a personal computer, but adaptable to any computer that runs Fortran. The theory behind the method is presented in a companion report *Estimating Hazard Functions for Repairable Components* (Atwood, 1990). That report, referred to from now on as EHF, is cited continually in this user's guide. Fuller technical explanation of the methods mentioned here can be found in EHF.

The outline of this guide is as follows. The data analysis setting is presented in Section 1, with just enough of the theory so that PHAZE can be used, and with a summary of a typical data analysis session. Section 2 describes the format for files that contain the failure data, files that are read as inputs by PHAZE. Section 3 explains all the PHAZE commands. Section 4 describes the work that was performed to verify and validate the correctness of PHAZE. Section 5 explains some of the technical details of the algorithms and the program; this section is intended only for statisticians and programmers. Section 6 lists the references, and Appendix A gives the complete record of an example PHAZE session.

1.1 Fundamental Assumptions

The concern is with the failure behavior of components. The failures are assumed to be governed by a Poisson process with *hazard function*, or *failure rate*, $\lambda(t)$. Here, t is time, typically measured from the component's installation. The Poisson assumption says that the probability of a failure in a short period $(t, t + \Delta t)$ asymptotically approaches $\lambda(t)\Delta t$, and failure counts in non-overlapping time intervals are independent. It is assumed that when a component fails, either it is immediately repaired (made as good as old) and placed back in service, or else it is replaced by a new component. Failures of distinct components are assumed to be independent. The data to be analyzed therefore consist of sequences of failure times of similar independent components.

Data for a component are called *time censored* if the component is observed for a fixed time period, or plant records covering a fixed time period are examined, and the failure times are recorded. The number of these failures is random. Data are called *failure censored* if the component is kept in service until a predetermined number of failures has occurred, at which time the component is removed from service. In this case the number of failures is fixed, but the end of the observation period equals the final failure time, and is random. The mathematical formulas used for statistical analysis differ slightly for time censored and failure censored data. They are given in EHF, but need not concern a user of PHAZE. To perform the analysis, however, PHAZE must know whether each component is time censored or failure censored. Therefore, the data file must state whether a component was replaced at the time of its final observed failure. If so, PHAZE treats the component as failure censored, and otherwise as time censored.

There are three parametric forms assumed by PHAZE for the hazard function $\lambda(t)$, all of the form $\lambda(t) = \lambda_0 h(t;\beta)$. The three forms are:

$\lambda(t) = \lambda_0 \exp(\beta t)$ (exponential hazard model)

$\lambda(t) = \lambda_0(1 + \beta t)$ (linear hazard model)

$\lambda(t) = \lambda_0(t/t_0)^\beta$ (Weibull hazard model)

In all three models, λ_0 has units 1/time. In the exponential and linear models, β has units 1/time, so that βt is dimensionless. Under these models, λ_0 is the value of the hazard function $\lambda(t)$ at time $t = 0$. In the Weibull model t_0 is an arbitrary normalizing time, and β is dimensionless. Under this model, λ_0 is the value of $\lambda(t)$ at $t = t_0$.

A typical PHAZE session consists of reading failure data from a file that was previously prepared, selecting one of the above three models, and performing data analysis. Often the analyst wants to decide whether $\lambda(t)$ is an increasing function, that is, whether failures tend to occur more frequently as time goes on. This is done in PHAZE by constructing a confidence interval for β, or by testing whether $\beta = 0$; the hazard function $\lambda(t)$ is increasing if $\beta > 0$, is constant if $\beta = 0$, and is decreasing if $\beta < 0$. In risk assessment it is also useful to estimate $\lambda(t)$ at any time t of interest, both by a point estimate and by a confidence interval. This is done in PHAZE by assuming one of the three models, estimating β and λ_0, and obtaining the corresponding estimate of $\lambda(t)$. In order to perform any of the above tasks, certain assumptions must be made; these assumptions should be checked

against the data. The PHAZE commands, described in Section 3, include commands for performing these checks of the assumptions.

1.2 Full and Conditional Likelihood

Before a typical data analysis session can be described, the following two approaches must be mentioned. Statistical inference is generally based on the likelihood, which depends on the data and on the parameter(s). Inference for β is of primary interest in many investigations of $\lambda(t)$, because it is β that determines whether $\lambda(t)$ is increasing. It is well known, and shown in Section 3 of EHF that the conditional likelihood can be used to perform inference for β, while not assuming that the components necessarily have a common value of λ_0, and without estimating λ_0 or the λ_0's. (The *conditional likelihood* is defined as the probability density of the nonreplacement failure times, given the failure counts for time censored components and given the final failure times for failure censored components. The *full likelihood* is this conditional likelihood times the probability of the failure counts for time censored components and the probability density of the final failure times of the failure censored components.)

Therefore the first exploratory analysis, used to verify assumptions of the model, should be based on the conditional likelihood. Later, when both parameters must be estimated simultaneously to produce an estimate of the hazard function $\lambda(t)$ at various times t, the full likelihood should be used.

1.3 Typical PHAZE Session

A typical analysis proceeds as follows. First the raw data (such as plant maintenance records) are read, interpreted, and encoded in a file that PHAZE can read. Then PHAZE is run to perform the following steps. The capital letters in parentheses below are the PHAZE commands used to perform each step. They are explained in Section 3.3, but are listed here to help guide a user who already has some familiarity with PHAZE. For a fuller treatment of any of the steps, see the explanations of the corresponding commands.

(1) Get and read a data file. (GD)

(2) Construct a cumulative failure plot, to get a preliminary graphical picture of any trends that may be present. (QQ)

Steps (3) through (7) are exploratory, and therefore should be based on the conditional likelihood, and not use an assumption of bivariate asymptotic normality. (FL, AN)

(3) Investigate whether the components have a common value of β. If they do, continue.

If they do not, try to find the reasons for the discrepancies, and consider splitting the data. (CC B)

(4) Investigate whether $\lambda(t)$ is increasing. That is, test $\beta = 0$ against the alternative $\beta > 0$. As supplemental information, construct a confidence interval for β, to see the range of plausible values. If $\lambda(t)$ does not seem to be increasing, in some situations the analyst would not be interested in continuing the study. (TE, CI B)

(5) Test whether the assumed model form is adequate. If it appears adequate, continue. If instead the data show statistically significant lack of fit to the assumed model, try to understand the reasons, and use a different model. (KS, QQ)

(6) Investigate whether the components have a common value of λ_0. If they do, continue. If they do not, try to find the reasons, and consider splitting the data. (CC L)

(7) Generate a two-dimensional confidence region for (β, λ_0), for comparison with the confidence ellipse to be produced in Step (9). (CI +)

The remaining steps should be based on the full likelihood, and on the assumption that $(\hat{\beta}, \log\hat{\lambda}_0)$, the maximum likelihood estimator (MLE), has approximately a bivariate normal distribution. (FL, AN)

(8) Find the MLE for (β, λ_0). (ML +)

(9) Investigate the adequacy of the normal approximation for the distribution of $(\hat{\beta}, \log\hat{\lambda}_0)$. (CI +)

(10) Get the MLE and a confidence interval for $\lambda(t)$ at various values of t. If the normal approximation is deemed adequate in Step (9), generate an approximate confidence interval; otherwise generate a conservative one. (ML H, CI H, HF)

A sample PHAZE session, following the above outline, is given in Appendix A.

1.4 Terminology and Formats

The following terminology is used throughout the later explanations, because all input for PHAZE is in free format. A *separator* is a blank, comma, or equal-sign. *Free format* means the following: Entries may appear anywhere on a line. However, every floating point number must be immediately followed by a separator, and every integer must be immediately followed by a non-numeric character. If successive character strings have a fixed length, they may optionally have separators between them. A character string that does not have a fixed length must be followed by a separator, and may not contain separators as part of the string.

Commands and their arguments may be entered in either upper case or lower case letters. The only restriction is that two-letter commands must be entirely upper case or entirely lower case. Letter arguments for a command do not have to be in the same case as the command.

2. DATA FILES

A data file records failure information that arose as follows. Each component was installed, or placed in service, at a certain time. The observation period, during which any failures were recorded, began on or after the installation time, and continued until some final time. The final observation time may simply be when no more data were recorded (for time censored data), or it may coincide with a failure that resulted in the component's being removed or replaced (in which case the data are treated as failure censored.)

The failure data are read from ASCII files, one file for each set of similar components. The file consists of a series of lines, with each component corresponding to one or more lines. The numbers for a component are in free format, but they must appear in a specified order. The elements that must be present for any component are, in order:

Component name (up to 14 characters, without embedded separators, not beginning with #)
Installation time (optional if times are dates and if installation date equals initial observation date, required otherwise)
Initial observation time
Final observation time
Number of failures in the observation period, an integer N
N failure times
The letter R (upper or lower case) if the component was removed or replaced at the time of the final failure.

The times may be entered as dates, integers, or floating point numbers, but all the times in the file must be of the same form. The form of the first time encountered in the file determines whether the times will all be interpreted as dates, integers, or floating point numbers. Dates are in the yymmdd format. For example 840329 means March 29, 1984. If the first time in the file is an integer

≥ 500101, then all times are interpreted as dates (Jan. 1, 1950 or later). If it is a smaller integer, all times are interpreted as integers. If it is a floating point number (that is, with a decimal point or in scientific notation), all times are interpreted as floating point numbers. All times other than dates are treated as if they are in units of hours.

Figure 1 shows an example of a data input file, using dates, with each line containing the elements listed above. For example, look at the third line. It says that the component MOV-1C was installed on Dec. 1, 1972. It was observed from Aug. 24, 1977, through October 1, 1987. During that time there were two events recorded as failures, on April 23, 1983 and on March 4, 1987. Note that three of the components were replaced as a result of their final failure; for those components, the final failure date and the end of the observation period agree, and the final 'R' tells PHAZE that this is not a coincidence.

If the inputs are dates, then the installation dates may be omitted. In that case, the installation date is set equal to the initial observation date. In any file, the syntax must be the same for all the components: either all have installation dates or none do.

A data record for a component may be continued on subsequent lines as follows. If fewer than N failure times have been entered, the last failure time on the line should be followed by (optional blanks and) a plus-sign. Then the next line will be interpreted as a continuation of the data for the component. Any non-continuation line that begins with a # will be interpreted as a comment, and ignored by PHAZE.

In Figure 1, note also that three of the component records are continued onto a second line, as indicated by the final plus signs. The installation dates are given; if the installation dates had all been equal to the beginning of observation, as they are in only a few cases, then all the installation dates could have been omitted. The comment at the end of the file is identified by the symbol #. The entries are arranged in neat columns for legibility, but this is not necessary.

Figure 2 shows an example data file with integer failure times. PHAZE treats the times as integers rather than dates because the first time is less than 500101. Even though the installation time equals the beginning of observation, the two numbers must both be entered.

```
MOV-1A      721201    770824 871001   1 810618
MOV-1B      721201    770824 871001   1 780706
MOV-1C      721201    770824 871001   2 830423 870304
MOV-1D      721201    770824 871001   7 830411 830520 840620 850814 860128 +
                                        860131 870304
MOV-1E      721201    770824 800219   1 800219 R
MOV-1E(R)   800220    800220 871001   3 810611 830313 870219
MOV-1F      721201    770824 820814   4 780605 810325 811001 820814 R
MOV-1F(R)   820815    820815 871001   1 850213
MOV-2A      730501    770824 871001   4 781015 850626 850628 851029
MOV-2B      730501    770824 871001   5 800826 801104 821218 850620 861204
MOV-2C      730501    770824 830426   2 811207 830426 r
MOV-2C(R)   830427    830427 871001   1 870226
MOV-2D      730501    770824 871001   6 780407 800513 800602 821218 850620+
                                        860715
MOV-2E      730501    770824 800322   0
MOV-2E(R)   800323    800323 871001   2 850620 850624
MOV-2F      730501    770824 871001   7 800509 821218 830424 830819 840412 +
                                        850620 850626
# File used as example by Atwood (1990)
```

Figure 1. Data file with times entered as dates

```
Comp1  1  1  1000 5  212  414  605  883  912
```

Figure 2. Data file with single component, integer failure times

```
validation-1  0.  3.  4.  1  3.7
validation-2  0.  3.  5.  3  3.6667 4.3333  5.  r
#Data for validation test of PHAZE, 2 components, 4 failures, 1 replacement
```

Figure 3. Data file with floating point failure times.

Figure 3 shows an example of a data input file with floating point failure times. The first time entered, in this example 0., must contain a decimal point or be in scientific notation (0E0). The subsequent times are not required to have decimal points, but will be interpreted as floating point numbers. For example, the entry 4. could have been entered as an integer 4 without changing the way that PHAZE interprets the entry.

The exact times of failure are interpreted slightly differently, depending on whether the input times are dates, integers, or floating point numbers. Suppose for example that the observation period was from 850101 to 851231, i.e., from January 1, 1985, to December 31, 1985. In this case, the observation period is considered to be 365 days long (1985 was not a leap year). That is, the observation period goes from the beginning of the day January 1 (just after midnight) to the end of the day December 31 (just before midnight). Likewise, the installation time is interpreted as the beginning of the installation date. Failures are assumed to occur at the midpoint (noon) of the failure date; the only exception to this is that replacement failures are counted as occurring at the end of the day, coinciding with the end of the observation period.

When the inputs are integers, the times are interpreted similarly. In the example of Figure 2, suppose for simplicity that the integers stand for hours. The installation time and the beginning of observation are at the beginning of hour 1, the end of the observation period is at the end of hour 1000, and the observation period lasts for 1000 hours (not for $1000 - 1 = 999$). The failure times are assumed to be at the middle of the hour. Therefore, for example, the first failure occurs 211.5 hours after the start of the observation period.

When the inputs are floating point numbers, the times are interpreted just as they are written. In the example of Figure 3, the second component has an observation period that lasts 2 hours ($= 5. - 3.$), and the first failure occurs 0.6667 hours after the start of the observation period.

3. PHAZE COMMANDS

3.1 Command Syntax

PHAZE has a command structure rather than a menu structure. That is, there is a set of commands that the user must know. The program prompts the user for a command, executes this command, and then prompts the user for the next command.

Most of the commands consist of two letters; only H and Q (for help and quit) consist of a single letter. Some of the commands require arguments, either letters or numbers depending on the command. Most of these arguments can be entered on the command line. If they are not entered, PHAZE will request them from the user. Input is in free format.

The commands are listed in Table 1, and presented below in alphabetical order. The commands are written in BOLD CAPITALS, and the arguments are written in *italics*. In each case, the command, with its possible arguments, is given at the left, and the command in English is given in the center. The bold capital letters in the English name of the command highlight the mnemonic relation to the PHAZE command. Values for the arguments must be substituted by the user. For example,

ML *parameter*

is the command to find the maximum likelihood estimate (MLE) of a parameter to be specified. If the user enters

ml b

MLB

ml,b

or any other variety of the command, PHAZE will find the MLE of β.

The commands CC, CI, and ML all take a parameter as an argument. With some restrictions depending on the command, the parameter generally may take the values

B for β

L for λ_0

+ for both β and λ_0

H for the hazard function $\lambda(t)$.

Table 1. Summary of commands

Parameter	Meaning
AN	Toggle option to treat ($\hat{\beta}$, $\log\hat{\lambda}_0$) as having a bivariate Asymptotically Normal distribution
CC *parameter*	Compare Components for *parameter*
CD	Toggle option to Center Data, for linear hazard model
CI *parameter conf*	Find *conf* % Confidence Interval for *parameter*
DD	Describe Data
DU	DUmp values in storage, for code development only
FL	Toggle option to use Full Likelihood
GD *filename*	Get Data stored in *filename*
H *class*	Show Help messages for a *class* of commands
HF *time*	Find Hazard Function at *time*
HF R *time1 time2*	Find Hazard Function in Range *time1* to *time2*
KS	Kolmogorov-Smirnov test of the model
LE	Toggle option to assume λ_0's Equal for all components
ML *parameter*	Find the Maximum Likelihood estimate (MLE) of *parameter*
OU *filename*	Define OUtput file, *filename*, for echo of session
PF	Toggle option to allow creation of Plot Files
Q	Quit
QQ	Calculate values for a Q-Q (quantile-quantile) plot
SB *value*	Set β_0 to *value*
SD	See Data in concise display
SM *model*	Select Model *model*
SN *time*	Set Normalizing t_0 to *time*, for Weibull model
ST	Show STatus
TE *value*	TEst β = *value*
#	Comment
*	For code development only

Certain commands have as an argument the name of a file, when this file is to be read or written. The filename may contain disk and directory information, and must contain the DOS extension, if any. Examples could be a:\data\mov.txt and \valves\chkval.dat. The filename may be at most 22 characters long. Three filenames are reserved for special meanings: NONE, HELP, and QUIT, in upper case or lower case. If a command is given with the filename NONE, no file is used. (This can be used to turn off writing to an output file, or to cancel the search for an input file which cannot be found for some reason.) If the filename HELP is entered, PHAZE briefly explains the need for a filename, and the user's possible responses. If the name QUIT is entered, PHAZE stops executing and returns control to DOS.

3.2 General Concepts

PHAZE stores the current environment. This environment consists of the choices listed below. In parentheses the initial value is shown, the value in effect at the start of the session. Also shown in capital letters are the commands that directly affect that choice.

- the data set most recently read (no initial data set; GD)
- the selected model
  - exponential, linear, or Weibull hazard model (no initial model; SM)
  - components assumed to have common λ_0 or not (initially no common λ_0; LE)
  - data centered or not, for linear hazard model (initially not centered; CD)
  - value of normalizing t_0, for Weibull hazard model (see SN for initial value; SN)
- analysis options
  - use conditional likelihood or full likelihood (initially conditional; FL)
  - use bivariate asymptotic normality of $(\hat{\beta}, \log\hat{\lambda}_0)$ or not (initially do not; AN)
- output options
  - echo session to a disk file or not (initially do not; OU)
  - allow writing of plot information to disk files or not (initially do not; PF)

PHAZE also stores a current value, denoted β_0, as the default value of β. This default value is used when λ_0 is to be estimated by the MLE or by a confidence interval. (An estimate of λ_0 depends on the assumed value of β.) This default β_0 also specifies the hypothesized value for the commands TE (test whether $\beta = \beta_0$), KS (perform a Kolmogorov-Smirnov test of the assumed model, with

$\beta = \beta_0$), and QQ (generate a Q-Q plot for the assumed model, with $\beta = \beta_0$). Whenever a data set, model, or analysis option changes, the value of β_0 is set to 0; the only exception is that changing t_0 does not change the meaning of β, and so does not cause β_0 to be reset. Whenever the MLE $\hat{\beta}$ is found, β_0 is reset equal to $\hat{\beta}$; most applications, such as finding the MLE of λ_0, or checking the goodness of fit of the model to the data, will be done when $\beta_0 = \hat{\beta}$. The user may also set β_0 manually by the command SB. Sometimes the calculation of $\hat{\beta}$, which results in resetting β_0, is invisible to the user. For example, to compare components to see if they have a common value of β, the MLE $\hat{\beta}$ must be found. Therefore the command CC B changes β_0 from its previous value to $\hat{\beta}$. Commands that use β_0 will print the value used. To be sure in advance of the value of β_0, use the command SB or ML B.

Some of the commands are *toggle* commands, that is they turn a switch on or off. The switch determines which of two possible options PHAZE uses. For example, PF is such a command. Each time the command PF is entered, the plot-file switch changes, either from off to on or from on to off.

3.3 Commands

AN — Toggle option to treat ($\hat{\beta}$, $\log\hat{\lambda}_0$) as having a bivariate Asymptotically Normal distribution

This toggles the bivariate-asymptotic-normality switch on or off. If the switch is on, the confidence region for (β, $\log\lambda_0$) is constructed to have an elliptical shape. If the switch is off, the region is based on a confidence interval for β, and a confidence interval for λ_0 for each value in the confidence interval for β. The effect of AN on confidence regions is detailed in Table 2. The default value of the bivariate-asymptotic-normality switch is off.

Turning on the bivariate-asymptotic-normality switch may reset other switches: The equal-lambdas switch (command LE) will be turned on if it is off, and the full-likelihood switch (command FL) will be turned on if it is off. These changes are announced to the user when they happen.

CC *parameter* — Compare Components for *parameter*

The allowed parameter values are B, L, and +. This command compares the components to see if they have a common value of β (when the parameter is B), or a common value of λ_0 assuming a common value of β (when the parameter is L). Using + as the parameter is equivalent to the calling CC B

Table 2. Effect of switches on CI command

Parameter	Asymptotic-Normality Switch Off Full-Likelihood Switch Off	Asymptotic-Normality Switch On Full-Likelihood Switch On
β	Use asymptotic normality of conditional log-likelihood.	Use asymptotic normality of $(\hat{\beta}, \log\hat{\lambda}_0)$, get confidence interval from marginal distribution of $\hat{\beta}$.
λ_0	Use exact distribution of $\hat{\lambda}_0$, assuming $\beta = \beta_0$. NOTE, this is done whether full-likelihood switch is on or off.	Use asymptotic normality of $(\hat{\beta}, \log\hat{\lambda}_0)$, get confidence interval from marginal distribution of $\log\hat{\lambda}_0$.
(β, λ_0)	Find confidence interval for β, then for each β_0 in interval find confidence interval for λ_0. Use $(1-\alpha/2)$ confidence intervals to get $(1-\alpha)$ region, for example 95% intervals to get 90% 2-dimensional region.	Use asymptotic bivariate normality of $(\hat{\beta}, \log\hat{\lambda}_0)$ to get confidence ellipse for $(\beta, \log\lambda_0)$.
$\lambda(t)$	Find maximum and minimum values of $\lambda(t)$ for (β, λ_0) in 2-dimensional confidence region, yielding conservative interval for $\lambda(t)$.	Use approximate normality of $\log\hat{\lambda}(t)$.

followed by CC L. If the parameter is not specified, the value used is the one remembered from the most recent use of CC, CI, or ML.

Figure 4 shows a sample print-out generated by the command CC B. It is explained here. For this comparison it is not assumed that the components all have the same value of β. Let β_j denote β for the jth component. The estimate of β_j is $\hat{\beta}_j$; this MLE uses only the data from the jth component. Similarly, let $\beta_{j,LO}$ and $\beta_{j,UP}$ denote the lower and upper ends of a 95% confidence interval for β_j, based only on data from the jth component. Finally, suppose that all the components *except* the jth have a common value of β, denoted by β_{-j} and estimated by the MLE $\hat{\beta}_{-j}$. The first page of Figure 4 shows, for each component, the numerical values of $(\beta_{j,LO}, \hat{\beta}_j, \beta_{j,UP})$ and of $\hat{\beta}_{-j}$. The second page of Figure 4 shows the significance level, based on the difference $\hat{\beta}_j - \hat{\beta}_{-j}$ and on the variance of $\hat{\beta}_j$ and $\hat{\beta}_{-j}$. The exact method of calculation is explained in Section 6.1 of EHF, and in Section 5.2 of this report. The significance level is the probability that $\hat{\beta}_j$ and $\hat{\beta}_{-j}$ would be as far apart as actually observed, if in fact all the components have the same β. A small value indicates strong evidence that the jth component has a different β from the other components. Finally the print-out shows a plotted 95% confidence interval, enclosed by parentheses, with $\hat{\beta}_j$ shown by a star (*), and $\hat{\beta}_{-j}$ shown by an I. For several components in Figure 4, not all the symbols are shown. In these cases, one symbol was printed over another symbol, thereby erasing it. The hierarchy is *, I, ().

Although Figure 4 does not show this, there may be plots in which an interval continues off the area shown; this is indicated at the end of the plotted portion by < or >. This occurs if a confidence interval is infinite (as can happen with the linear hazard model). In such a case, the numerical printout of the intervals, as illustrated in the first page of Figure 4, should make clear what has happened.

At the bottom of Figure 4 is the overall MLE and the overall 95% confidence interval, calculated under the assumption that all the components have a common β. This is shown just as for the individual components, except that $\hat{\beta}_{-j}$ is now meaningless, and the significance level is not shown. To get an overall significance level, we must recognize that some components will appear extreme from chance alone. This can be accounted for by using the Bonferroni inequality, discussed in many texts, and by Alt (1982). In the present context, for any number c it says that

P(at least one of k significance levels is $\leq c$) $\leq kc$.

```
-> ccb
  95.0% conf. intervals with MLEs for beta, by component and MLE without
comp.
 MOV-1A          (-1.10E-04,-1.72E-05, 6.48E-05)          1.42E-05
 MOV-1B          (-4.85E-04,-1.32E-04, 2.55E-05)          1.54E-05
 MOV-1C          (-1.77E-05, 4.09E-05, 1.08E-04)          1.21E-05
 MOV-1D          ( 4.06E-06, 3.90E-05, 7.52E-05)          8.57E-06
 MOV-1E            not estimable
 MOV-1E(R)       (-5.41E-05,-4.31E-07, 5.32E-05)          1.41E-05
 MOV-1F          (-6.39E-05, 1.88E-05, 1.03E-04)          1.33E-05
 MOV-1F(R)       (-1.67E-04,-3.35E-06, 1.59E-04)          1.35E-05
 MOV-2A          (-2.04E-05, 1.67E-05, 5.46E-05)          1.30E-05
 MOV-2B          (-2.35E-05, 9.12E-06, 4.20E-05)          1.40E-05
 MOV-2C          (-8.71E-05, 7.48E-05, 3.04E-04)          1.29E-05
 MOV-2C(R)       (-7.88E-05, 1.91E-04, 6.82E-04)          1.28E-05
 MOV-2D          (-3.47E-05,-4.86E-06, 2.49E-05)          1.71E-05
 MOV-2E            not estimable
 MOV-2E(R)       (-3.18E-05, 3.99E-05, 1.19E-04)          1.26E-05
 MOV-2F          (-1.55E-05, 1.29E-05, 4.16E-05)          1.35E-05
 OVERALL         ( 5.32E-07, 1.34E-05, 2.62E-05)
```

Figure 4. Example output from CC B command. (Page 1 of 2)

```
Comparison of beta values for components
Component     Signif    95.0% Confidence Interval
MOV-1A        0.48                    (----*I-)
MOV-1B        0.09  (---------------*------I)
MOV-1C        0.44                        (I*--)
MOV-1D        0.14                         I*-)
MOV-1E
MOV-1E(R)     0.64                      (-*I-)
MOV-1F        0.91                     (---*---)
MOV-1F(R)     0.82                 (------*I------)
MOV-2A        0.87                        (*-)
MOV-2B        0.80                       (-*)
MOV-2C        0.61                    (----I--*----------)
MOV-2C(R)     0.33                     (---I-------*----------------------)
MOV-2D        0.21                       (*I)
MOV-2E
MOV-2E(R)     0.54                       (-I*---)
MOV-2F        0.97                        (*)

OVERALL                                   (*)
Overall significance level for testing equality of betas = 1.00

Signif for jth component means estimated probability that
beta-hat(j) - beta-hat(others) is at least as extreme as observed

Overall significance level is P[at least one signif(j) <= observed min],
and is computed with Bonferroni inequality
```

Figure 4. (Page 2 of 2)

The inequality is close to equality when kc is small. Therefore the overall significance level for testing equality of the β_j's is the number of components having calculated significance levels times the minimum significance level calculated for a component. A small value of the attained overall significance level (say 0.05 or smaller) shows that there is strong evidence against the hypothesis that all the components have the same value of β. The approximate overall attained significance level in Figure 4 is 14 (the number of components having calculated significance levels) times 0.09 (the smallest significance level attained), which is larger than 1.0. Therefore it is printed as 1.00 in Figure 4. This means that there is no evidence against the assumption that all the components have the same value of β.

PHAZE uses the Bonferroni multiplier k when $k \geq 3$, that is, when three or more components have significance levels printed. When $k = 2$, the smallest number for which a comparison of components is possible, the Bonferroni multiplier is not used, and the overall significance level reported is the smaller significance level of the two. The reason for this is as follows: Normally, the significance level for the first component is based on the difference $\hat{\beta}_1 - \hat{\beta}_2$, while the significance level for the second is based on $\hat{\beta}_2 - \hat{\beta}_1$. Therefore there is only one significance level, corresponding to the difference, and multiplying it by two would be incorrect. Normally the significance levels printed for the two components are the same, although they can be calculated as different if one component has one failure while the other has more, as explained in Section 5.2.

CD Toggle option to Center Data, for linear hazard model

For the linear hazard model, time can be measured from any point, not only from the component's installation time. A useful time to use as the origin is the center of the observation period, t_{mid}, defined precisely in Section 5.2. This causes the estimators $\hat{\beta}$ and $\hat{\lambda}_0$ to be independent of each other, as explained in Section 4.3 of EHF. The CD command toggles the value of the center-data switch between on and off. If on, the data are centered. If off, the data are uncentered, and times are measured from the component's installation. The default value is off.

In a future version of PHAZE, this command may be extended to CD *value*. The input for *value* will be a user-defined origin, from which all times are to be measured. The command CD without any argument will work as before, centering the data at t_{mid} if the data were previously uncentered, and uncentering the data if they were previously centered at any value other than 0. The command will then work for both the exponential and the linear hazard models. When this extension is implemented, it will be shown in the output printed by the Help command.

CI *parameter conf* Find *conf* % Confidence Interval for *parameter*

The parameter may be B, L, +, or H. The confidence level *conf* may be any number between 0 and 100. If the parameter is B or L, then a confidence interval is found for β or λ_0, respectively. If the parameter is H, the command may be entered as

CI H *conf time*

PHAZE will read t, the value of *time*, or will prompt the user for t, and then will produce a confidence interval for the hazard function $\lambda(t)$ at t. If the parameter is +, then a two-dimensional confidence region is found for (β, λ_0). This region is also plotted, with λ_0 plotted on a logarithmic scale. If the PF switch is on, the information for the two-dimensional region may be written to a disk file for later analysis or plotting. If the arguments are not all specified, PHAZE will request them.

The form of all the confidence intervals and regions depends on whether the full-likelihood and asymptotic-normality switches are turned on or off. The allowed combinations are shown in Table 2 with their effects.

The command CI + can be used to check the assumption that $(\hat{\beta}, \log\hat{\lambda}_0)$ has approximately a bivariate normal distribution. First use the conditional likelihood and no assumption of bivariate asymptotic normality, and generate a two-dimensional confidence region. Then use the full likelihood and the asymptotic normality assumption to generate a confidence ellipse. The bivariate normal approximation appears adequate if the two confidence regions cover mostly the same territory, that is, they overlap well, and if the confidence ellipse is not truncated by some theoretical limit. Figures 5 through 8, taken from EHF, illustrate the concept. Figures 5 and 6, based on the exponential and Weibull hazard models, respectively, show very good agreement between the two regions, indicating that the bivariate normal approximation is quite acceptable for those models and that data set. Figure 7 shows the two regions based on the linear model and uncentered data. The overlap is terrible, and the ellipse is truncated at the theoretical lower bound for β. Figure 8 shows the two regions based on the linear model with centered data. The overlap is good, but the ellipse is truncated at the theoretical upper bound for β. These two pictures show that the bivariate normal approximation is inadequate for the linear model, whether the data are centered or not.

Such overlays are not printed directly by PHAZE; output disk files from PHAZE were used as inputs for a graphics package, which printed Figures 5 through 8. In a future version of PHAZE, the command CI may be extended to allow the parameter to be O. (The O stands for overlay.) The

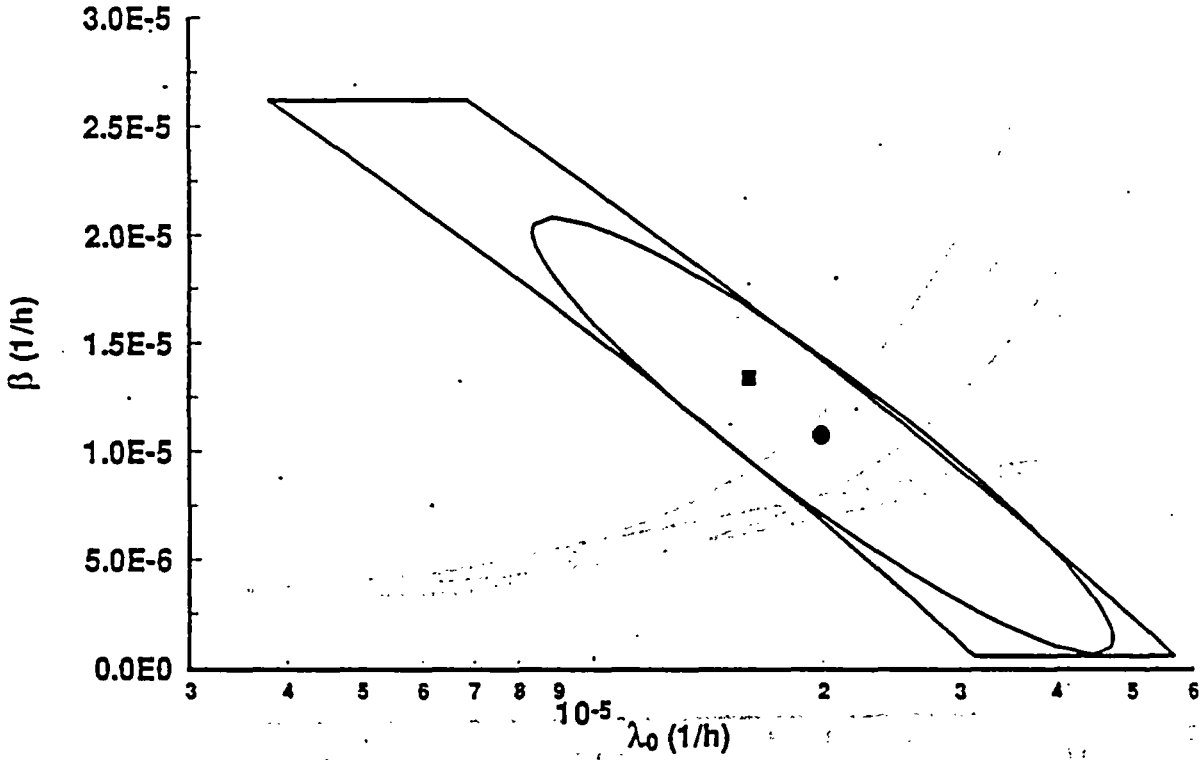


Figure 5. 90% confidence regions for (β, λ_0), based on exponential hazard model

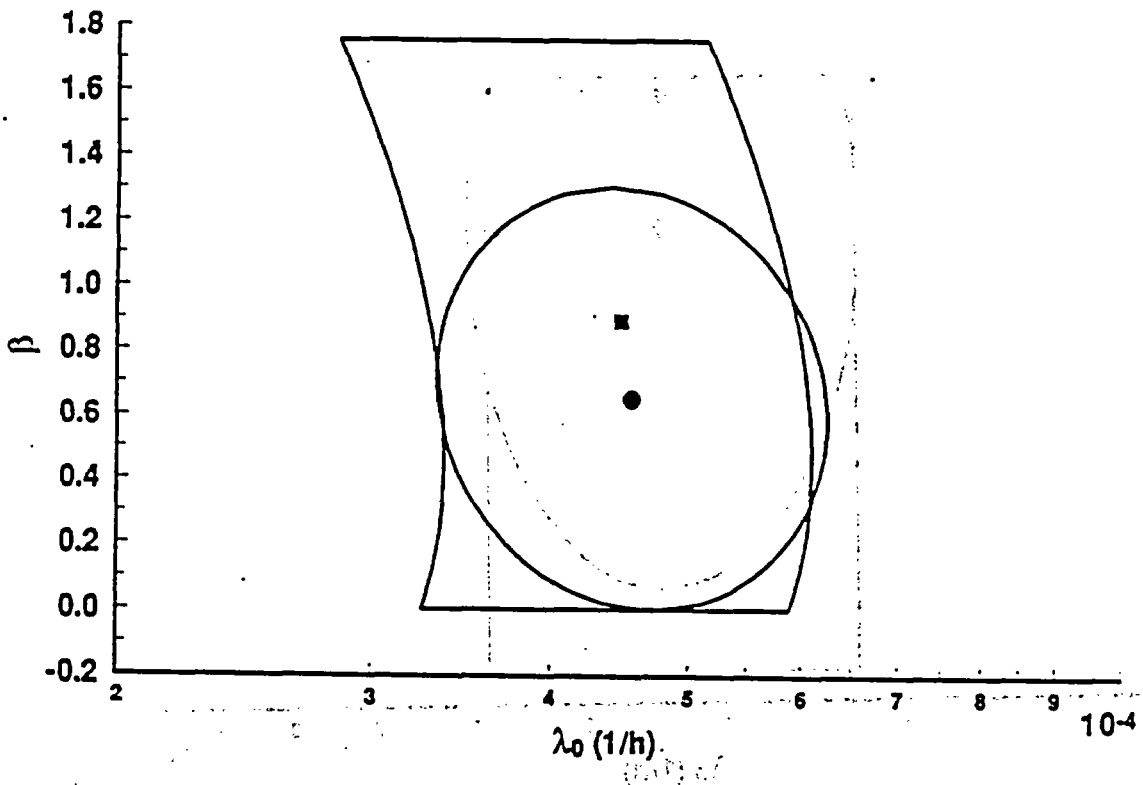


Figure 6. 90% confidence regions for (β, λ_0), based on Weibull hazard model

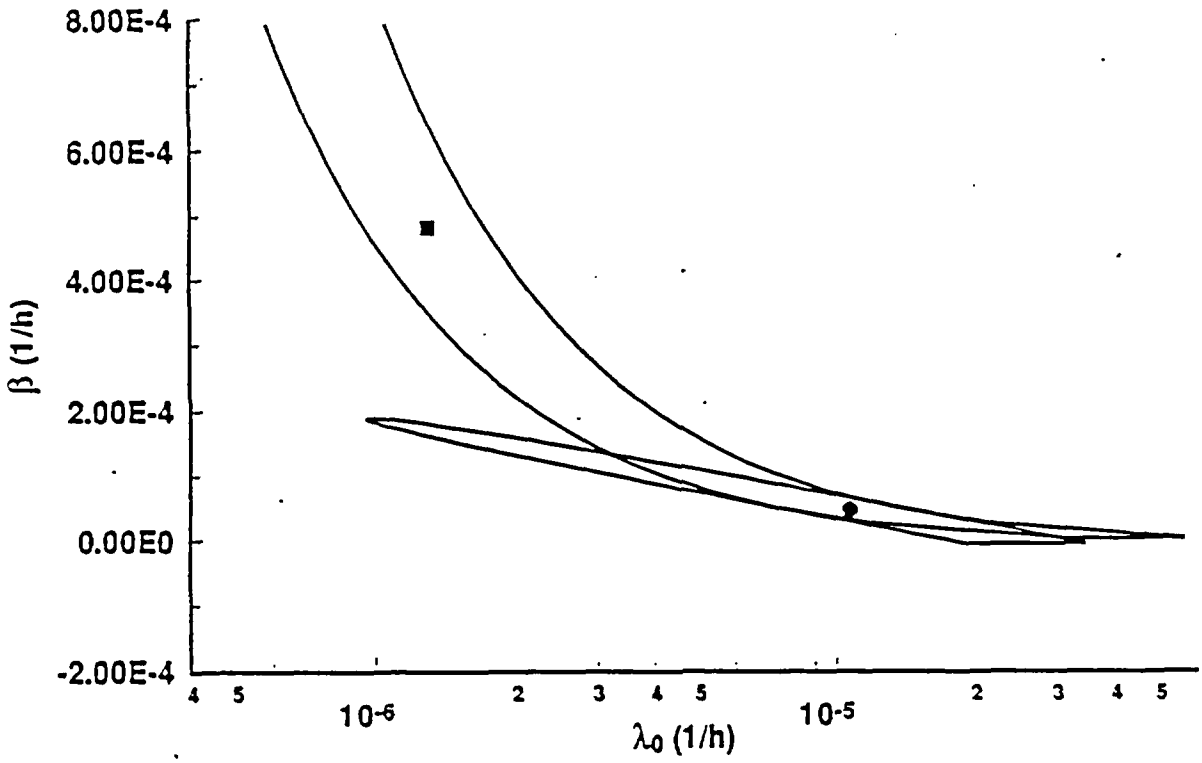


Figure 7. 90% confidence regions for (β, λ_0), based on linear hazard model, uncentered data

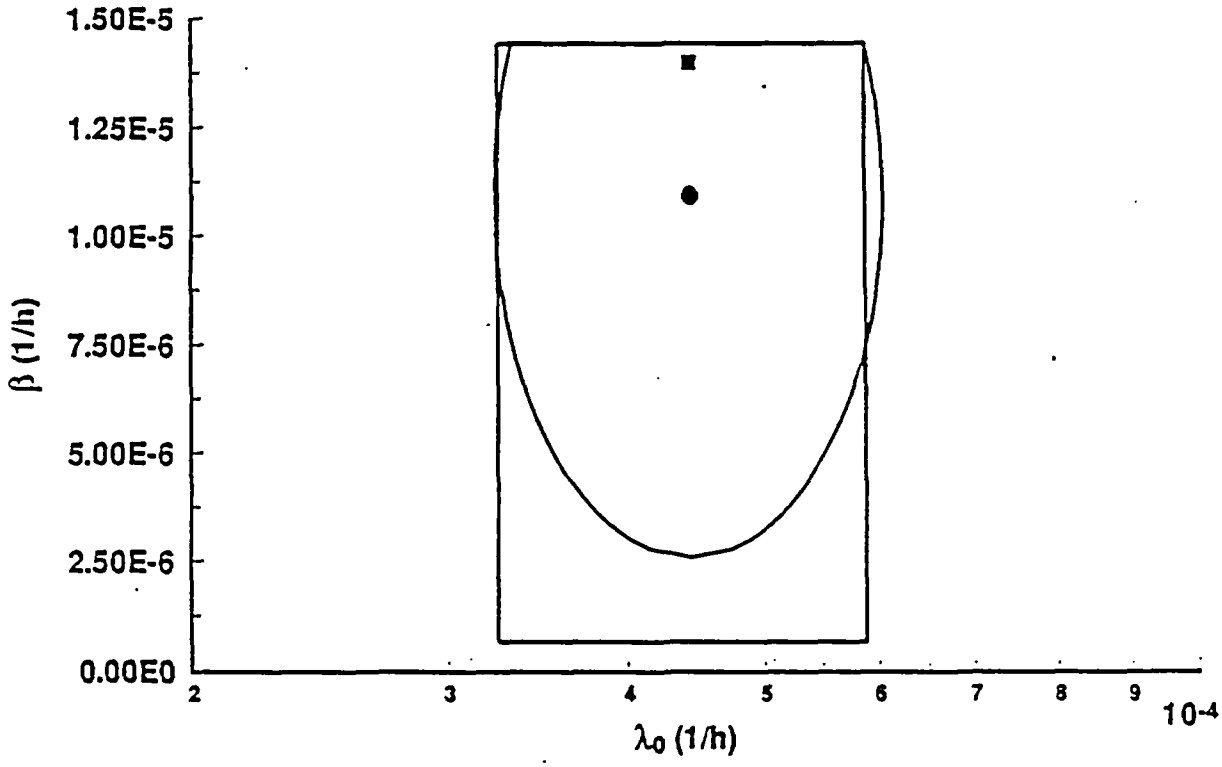


Figure 8. 90% confidence regions for (β, λ_0), based on linear hazard model, centered data

command CI O *conf* will result in two calls of CI + *conf*, first with bivariate asymptotic normality turned off, then with it turned on. The two resulting confidence regions will be overlaid in a printer plot on the terminal screen, instead of only being printed separately as they are now. When this change is implemented, the message printed by the Help command will show this.

DD Describe Data

This command generates summary statistics for the current data set. For each component, PHAZE prints the component's name, the number of nonreplacement failures, the length of the observation period, the normalized mean of the nonreplacement failure times, whether the component was replaced upon failure at the end of the observation period, and the component's age at the start of the observation period. The normalized mean of the nonreplacement failure times means the following. Let t be a nonreplacement failure time, and let s_0 and s_1 be the start and end of the observation period, respectively. Then the normalized replacement time is $(t - s_0)/(s_1 - s_0)$, and the mean is the arithmetic average of these terms. The output from DD is briefer than the output from SD, only one line per component.

DU DUmp values in storage, for code development only

This command was used in the code development. It writes values stored in a multi-purpose common block to the terminal. It is a command for programmers and debuggers only.

FL Toggle option to use Full Likelihood

When the full-likelihood switch is on, the full likelihood is used for maximum likelihood estimation of β and confidence intervals for β. When it is off, the conditional likelihood is used for these estimates of β. Estimation of λ_0 is always based on substituting the default β_0 into the full likelihood. (The meaning of β_0 is explained in Section 3.2.) The initial value of the full-likelihood switch is off.

The full and conditional likelihood are explained briefly in Section 1.2, and more thoroughly in EHF. The conditional likelihood is appropriate when it is not certain whether all the components have the same value of λ_0, and when the primary interest of analysis is in β. Therefore it is appropriate when the user initially is trying to decide if β is positive, and is not sure if the components have a common λ_0 or if the model form is correct. The full likelihood is appropriate when $\hat{\beta}$ and $\hat{\lambda}_0$ are to be used together to estimate the hazard function.

GD *filename* Get Data stored in *filename*

This command causes PHAZE to find and read the specified file. If the filename is not given, PHAZE will prompt the user for it. The file must be in the format described in the Section 3, "Data Files for Input." The data file is checked for internal consistency in many ways, and the results are announced to the user. If fatal errors are found, PHAZE will not use the file.

H *class* Show Help messages for a *class* of commands

This command causes PHAZE to write information that approximately fills a terminal screen. The printed information is a list of certain commands, each with its possible arguments, and a brief explanation of the command. Use class D for commands that control Data or files, I for commands that perform statistical Inference, and O for commands that set or show Options.

HF *time* Find Hazard Function at *time*

This command produces the MLE and 90% confidence interval for the hazard function $\lambda(t)$, where t is the specified time. It is a short command to accomplish the same result as the sequence

ML H *time*

CI H 90 *time*

As mentioned for the CI command, the confidence intervals depend on the settings of certain switches, as shown in Table 2.

HF R *time1* *time2* Find Hazard Function in Range *time1* to *time2*

This executes the command

HF *t*

for evenly spaced values of t in the range *time1* to *time2*. The default range is the total range of the component observation periods, so that *time1* equals the minimum observed age and *time2* equals the maximum observed age. If no times are entered on the command line, PHAZE reports the default values, and lets the user select them or other values. The number of increments in the range defaults to 20, but is controlled by the user. If the bivariate-asymptotic-normality switch (command AN) is on, the distribution of the MLE $\hat{\lambda}(t)$ is approximately lognormal. In this case the mean and the error factor (defined as the 95th percentile divided by the median) of the distribution are also printed, for each requested time t in the range. If the PF switch is on, the user has the option of writing this information to a disk file, so that the hazard function $\lambda(t)$ (MLE and 90% confidence interval, and the median and error factor in the lognormal case) can be plotted as a function of t or used as inputs to other programs.

KS Kolmogorov-Smirnov test of the model

This performs a Kolmogorov-Smirnov test of the hypothesis that the data come from the assumed model, with $\beta = \beta_0$. This formal statistical test corresponds to the informal graphical test given by the QQ command. Here β_0 is as described in Section 3.1 "Command Syntax," either the initial value of 0, or the MLE $\hat{\beta}$, or a value that the user has defined with the SB command. The test uses the conditional distribution of the data, given the failure counts and the final observation times for the components. This conditional distribution depends only on β, not on the (possibly different) values of the λ_0's; it is written out at the beginning of Section 6.3 of EHF. As discussed in EHF, when KS is used to test the adequacy of the model form, setting $\beta_0 = \hat{\beta}$ results in not rejecting the model often enough, because the best fitting β_0 has been used; that is, using the best fitting β_0 violates the theoretical assumptions, and therefore the true significance level is smaller than the printed significance level. On the other hand, no better method is known.

When $\beta_0 = 0$, a second Kolmogorov-Smirnov test is also performed, of the hypothesis that all the failures result from a Poisson process with constant failure rate. This hypothesis is true when all the components have $\beta = 0$ and a common value of λ_0, and the number of components being observed is the same at all times (that is, no components are introduced part way through the observation period, and any component replacements are performed instantaneously.) For this hypothesis, times are thought of in terms of the observation date, not in terms of the components' ages.

As discussed in Section 4.3 below, the significance level calculated by PHAZE may be inaccurate by $\pm 1\%$ or more. This means that the theoretical significance level is not calculated accurately; this is not to be confused with the above comment that the actual significance level is smaller than the theoretical one.

When the PF switch is turned on, the command KS offers the user the option of printing output to a plot file.

LE Toggle option to assume λ_0's Equal for all components

If the equal-lambdas switch is on, the components are assumed to have a common value of λ_0. If the switch is off, they are not assumed to necessarily have a common value of λ_0. The initial value is off.

Turning off the equal-lambdas switch may reset other switches: The bivariate-asymptotic-normality switch (command AN) will be turned off if it is on, and the full-likelihood switch (command FL) will be turned off if it is on. Thus, turning off the equal-lambdas switch resets the analysis options to their original values.

ML *parameter* Find the Maximum Likelihood estimate (MLE) of *parameter*

Here *parameter* may be B, L, +, or H, to obtain the MLE for β, λ_0, the pair (β, λ_0), or the hazard function $\lambda(t)$, respectively. If the bivariate-asymptotic-normality switch is turned on, via the AN command, then ML + produces not only the MLE for both parameters, but also the Fisher information matrix and the asymptotic covariance matrix for ($\hat{\beta}$, $\log\hat{\lambda}_0$). The formulas used, and the resulting MLEs, depend on whether the full likelihood or the conditional likelihood is used, and whether the λ_0's are assumed to be equal or not. These options are controlled by the commands FL, AN, and LE. If the parameter is not specified, the value used is the one remembered from the most recent use of CC, CI, or ML.

OU *filename* Define OUtput file, *filename*, for echo of session

This command allows the user to echo the current session to a disk file, for later printing or for documentation. To change the output file from a current file to no disk output, enter OU and enter NONE for the filename. Initially there is no disk output file.

PF Toggle option to allow creation of Plot Files

The commands CC, CI +, HF R, and QQ all produce output that could be plotted graphically. If the plot-file switch is turned on, PHAZE will ask the user if this output should be written to a disk file, for possible later use by a graphics program. If the plot-file switch is turned off, PHAZE does not offer the possibility of writing such a file. The initial value for the plot-file switch is off.

Q Quit

This command terminates the PHAZE session, and returns control to DOS.

QQ Calculate values for a Q-Q (quantile-quantile) plot

Q-Q plots are described in Section 6.3 of EHF and by Snee and Pfeifer (1983). A Q-Q plot provides a graphical check of the correctness of the model, corresponding to the quantitative test of the KS command. Figure 9, taken from EHF, shows an example of a Q-Q plot. The points fall approximately on a straight line, indicating that the model fits the data well.

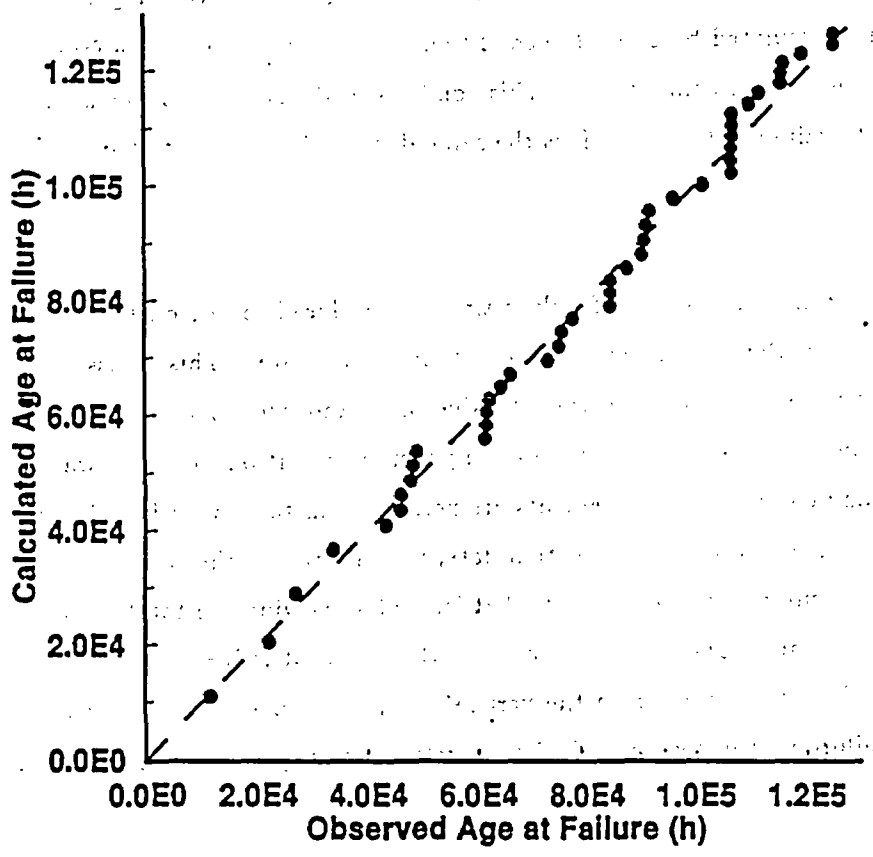


Figure 9. Q-Q plot, based on exponential hazard model

The output from the command QQ consists of a list of the observed failure times, and a corresponding list of the expected failure times under the assumed model with $\beta = \beta_0$. These lists give the x and y coordinates of the points in the Q-Q plot. Here β_0 is as described in the Section 3.2, either the initial value of 0, or the MLE $\hat{\beta}$, or a value that the user has defined with the SB command. The expected failure times are calculated based on the conditional distribution, given the failure counts and the final observation times for the components. This conditional distribution depends only on β, not on the (possibly different) values of the λ_0's. It is the same distribution that is tested by the command KS.

When $\beta_0 = 0$, a second pair of lists of failure times is produced, based on the assumption that all the failures result from a Poisson process with constant failure rate. This is assumption is true when all the components have $\beta = 0$ and a common value of λ_0 and when the number of components being observed is the same at all times (that is, no components are introduced part way through the observation period, and any component replacements are performed instantaneously.) For this assumption, times are thought of in terms of the observation date, not in terms of the components' ages. This Q-Q plot is equivalent to a cumulative failure plot, that is, a plot showing the total number of failures that have occurred by any time. Under the assumption, the expected failure times are uniform over the observation period. If they are plotted on the vertical axis, they have equal spaces between them. To make a cumulative failure plot instead of a Q-Q plot, relabel the expected failure times as 1, 2, etc. instead of as times.

When the plot-file switch (command PF) is turned on, the command QQ offers the user the option of printing the lists to a plot file.

At present, the command QQ does not cause a Q-Q plot to be printed on the terminal screen: it only prints a pair of lists for use by a graphics package. A Q-Q plot on the terminal screen may be added to a future version of PHAZE. When this capability is added, the command structure of QQ will not change at all, only the form of the output.

SB *value* Set β_0 to *value*

This command sets β_0 to the specified number *value*. See the explanation of the use of β_0 in Section 3.2. The number β_0 is used by the commands CC L, CI L, KS, ML L, QQ, and TE.

SD See Data in concise display

This command takes the current data set and prints the failure information for each component, with all times measured from the component's installation. Dates are converted to elapsed hours. The information printed consists of: the component's name, age at the beginning and at the end of the observation period, the age at the time of each nonreplacement failure, the average of these ages, and the age at the component's replacement if the component was replaced upon failure at the end of the observation period. This command gives more output than DD, because it lists the individual failure times.

SM *model* Select Model *model*

The value entered for *model* must be E, L, or W, for the exponential, linear, or Weibull hazard model. There is no default.

SN *time* Set Normalizing t_0 to *time*, for Weibull model

This sets t_0 to the specified time. The default value of t_0 is $\max_j r_j$, the largest observation range of any component, where the observation range r_j is defined as the the end time of the observation period for the jth component minus the starting time of the observation period. A value for t_0 such as t_{mid}, defined for the command CD in Section 5.2, results in MLEs $\hat{\beta}$ and $\hat{\lambda}_0$ that are less strongly correlated. The SN command is meaningful only for the Weibull hazard model.

ST Show STatus

This command causes PHAZE to show the status of the current environment, as defined in Section 3.2. The current values of the data, the model, and the analysis options are all shown. The values of the output options are shown if they are not the (write-nothing) defaults.

TE *value* TEst $\beta = value$

This performs a statistical test of the hypothesis

H_0: $\beta = value$

against the one-sided alternative

H_1: $\beta > value$.

To test for the presence of aging (β positive), *value* should be entered as 0. If no number is entered for *value*, PHAZE uses the current value of β_0. Note, this command does not reset β_0, regardless of the number entered as *value*. Regardless of the current value of the full-likelihood switch (command FL), the test is based on the conditional likelihood, as explained in Section 5.2 below.

A test result is shorthand for part of the information given by a confidence interval. The relation is that the test accepts the values that are in a confidence interval, and rejects all other values. More precisely, a test rejects β_0 at the significance level α if β_0 is not in a $100(1-\alpha)\%$ confidence interval. Two-sided tests correspond to two-sided confidence intervals. The one-sided test performed by the command TE corresponds to a one-sided confidence interval (β_{LO}, ∞). Therefore, as a supplement to the command TE, it is helpful to construct a (two-sided) confidence interval for β, with the command CI B. This shows whether the range of plausible β is wide or narrow.

Comment

Any line beginning with # is treated as a comment. It is ignored as a command, but is echoed to any output file that was established by the OU command. If the terminal is being copied to the printer (via Ctrl-PrintScreen), the comment will be printed there.

* For code development only

This character was used for testing the code. By changing the coding corresponding to the command *, PHAZE can be directed to execute various commands. The present coding merely causes PHAZE to read the next command, so * can be used for a comment line.

4. VERIFICATION AND VALIDATION

This section describes the work that was done to verify and validate the correctness of PHAZE.

4.1 Overview

ANSI/IEEE Std 730 (IEEE, 1984) states that software is logically produced in three steps:

- software requirements specification (SRS)
- software design
- code (in some computer language)

The SRS describes the requirements (functions, performances, design constraints, and attributes) of the software and the external interfaces, while the software design specifies the algorithms, data bases, internal interfaces, etc. "Verification" then means the ascertainment that the requirements of the SRS are implemented in the software design, and that the design is correctly implemented in the code.

"Validation" means the demonstration that the code, when executed, complies with the requirements of the SRS.

The SRS is that the program should read data and then carry out the methods given by EHF to analyze the data. All results should be accurate to at least two or three significant digits.

The software design was chosen for simplicity as follows. The main program consists primarily of a long series of if-then-else blocks, with each block corresponding to one of the allowed commands. When the user enters a command, the main program proceeds through the series of blocks until it locates the command, at which point it calls the appropriate subprogram to execute the command. When the command has been executed, control passes to the top of the main program, where the next command is read. Where possible, the subroutines were taken from existing programs that had been run successfully for years, or from published sources. About 5% of the lines of code were obtained this way, with the rest being written especially for PHAZE.

The subprograms in PHAZE fall into several classes:

Class (1) Subprograms that evaluate formulas; for example, the function DV evaluates the derivative v' according to the formulas in Table 1 of EHF.

Class (2) Subprograms that use these calculations to produce the desired output; for example, the subroutine CONFB calculates a confidence interval for β, and prints it if so instructed.

Class (3) Subprograms that calculate standard quantities; for example, the function ANORDF calculates the normal cumulative distribution function at a specified point.

Class (4) Subprograms that decipher input and manipulate files; for example RDINT starts reading the current input line at a specified location and finds the next integer on the line; GETNAM asks the user for the name of a file, and either finds it or creates it, as instructed.

The verification and validation activities differed somewhat for subprograms in different classes, as described below.

4.2 Verification

The mathematical formulas used are developed in EHF. In principle, they have long been known; the only new developments are some of the exact algebraic forms taken for the various models. These expressions were all either derived or confirmed using the symbolic computer program Mathematica (Wolfram, 1988). The formulas are given in Table 1 of EHF.

Subprograms in Classes (1) and (2) were verified by comparing them, line by line, with the formulas or the methods given in EHF. Subprograms in Class (3) were taken either from printed sources or directly from Fortran programs on disk; those copied from printed sources were verified by line-by-line inspection of the code. The subprograms in Class (4) were originally written in 1982. They perform tasks for which wrong behavior would be apparent. Because they have been used frequently in various applications since 1982, evolving only slightly, they are believed to be correct. Before they were included in PHAZE, they were briefly inspected but not carefully verified. The verification activity for all the classes of subprograms took place from December 1989 through March 1990. A few typographical errors that had been missed during the above verification were uncovered during the validation exercises described below, and during a separate static analysis of the code.

4.3 Validation

When considering the accuracy of the calculations, we must distinguish between quantities that are found by numerical iteration and quantities that are calculated from formulas. Typically, the MLE and confidence limits for β are found by numerical iteration. The accuracy required can be specified in advance, with more accuracy requiring more computing time. In PHAZE, these confidence bounds are required to have a relative error of at most 0.001, i.e., a slight error in the third significant digit. The MLE $\hat{\beta}$ is required to have relative error 0.00001, for reasons given below. To pass the validation tests described below, PHAZE must correctly iterate to find the MLE and confidence limits for β to the required accuracy. In addition, the calculations that are simple evaluation of formulas must be accurate enough so that the results of iterations are based on correct values.

Class (1)

The subprograms in Class (1) all involve only evaluation of formulas. To validate these

subprograms, a data set was created with one component and one failure. It was made this simple so that correct results could be found with a hand calculator. For testing calculations with the Weibull model, two variations of the data set were used, one in which the component was observed from its installation, the other in which the component was observed starting at a non-zero initial age; these cases correspond to two different sets of formulas in EHF. All calculated results were correct to 5 significant digits, except that under the exponential model when β was near $-\infty$, the derivative of the conditional log-likelihood was accurate to fewer places.

Many of the expressions must be evaluated using asymptotic approximations for certain values of β (β near 0 under the exponential or linear hazard model, or β near -1 under the Weibull hazard model.) The subprograms therefore use the standard formula or the asymptotic approximation depending on whether β is greater than or less than some critical value. Many of the validation calculations focused on β near these critical values, to make sure that the calculated expression varied continuously as β moved from one side of the critical value to the other. In one case (calculating the information for β under the linear hazard model), it was found that the second order asymptotic approximation was inadequate; the fourth order approximation was therefore put into the Fortran coding, and into Table 1 of EHF.

Class (2)

Some of the subroutines in Class (2) require more than one component, for example, subroutines that compare components to see if they have a common β. For these, a second data set was used, having two components and four failures, and given in Figure 3 above. It was found that sometimes the covariance matrix for $(\hat{\beta}, \log\hat{\lambda}_0)$ was accurate only to two significant digits. This was the case when $\hat{\beta}$ and $\log\hat{\lambda}_0$ were strongly correlated, and when the information matrix (the inverse of the covariance matrix) was only accurate to four places. Two things were done to mitigate this problem. First, the MLE $\hat{\beta}$ is now required to be found to approximately 5-place accuracy (relative error 0.00001) rather than 4-place accuracy. This means that the information matrix is now accurate to about five significant digits. Second, a warning message is now printed when the correlation of $\hat{\beta}$ and $\log\hat{\lambda}_0$ is strong enough to greatly magnify the inaccuracy of the inverse matrix.

Class (3)

To validate the members of Class (3), the subprograms were individually called with various

inputs, and the outputs were compared to values in published statistical tables, or values produced by the computer package Minitab (1985). The tests showed that several internal checks were needed to catch illegal inputs; these checks were added to the coding. The answers produced were correct, to within the accuracy printed by the tables or by Minitab, with one exception: The calculated significance levels for the Kolmogorov-Smirnov (K-S) test may be off by more than ± 0.01. This inaccuracy seems unavoidable. For sample sizes in the range 20 to 35, published values only give the K-S statistic to two decimal places, meaning that interpolated significance levels could be off by 0.01 or more. The asymptotic formula for the significance level, when the sample size is in this range, also seems to have only this degree of accuracy.

Class (4)

The subprograms in Class (4) were validated indirectly. It was verified that PHAZE correctly read data files, wrote requested output files, and interpreted the user's commands. Therefore the subprograms in Class (4) seem to work correctly.

Other Validation Checks

A static analysis of the code was performed using the INEL software CRAFT. Two undefined variables were identified—typographical errors that were corrected. Three unused variables were identified, but two were left in the coding to facilitate possible future program enhancement. As a final check, the program was compiled with three compilers: IBM Professional Fortran/2 Version 1.01, Lahey Fortran 77 Version 4.00, and Microsoft Fortran Version 4.1. To compile the program under Microsoft Fortran, it was necessary to reduce the length of the main driver program, making subroutines out of major sections. After this restructuring, the program was successfully compiled with all three compilers. One useful option in the file-handling subroutines allows the user to append output to an earlier file. This option works with Professional Fortran and Lahey Fortran, but not with Microsoft Fortran. Therefore, two source-code versions of the file-handling subroutine GETNAM are kept, one with the append-option and one without it.

The validation tests described above were carried out in March through May, 1990. The verification and validation history of each subprogram is documented in comments at the beginning of the subprogram's source code.

5. TECHNICAL DETAILS

5.1 Language and Portability

PHAZE is written in ANSI standard Fortran 77, with the following two exceptions: (1) Lower case letters are used freely in comments, and in Fortran statements used for debugging. (2) One version of the subroutine GETNAM includes the line

```
OPEN (UNIT=IUN,FILE=FILNAM,STATUS='OLD',ACCESS='APPEND')
```

even though ACCESS='APPEND' is not ANSI standard. This line works with IBM Professional Fortran/2 Version 1.01 and with Lahey Fortran 77 Version 4.00, but not with Microsoft Fortran Version 4.1. Therefore, two versions of the source code for that subroutine are kept, one with the non-standard line and one without it.

Several machine-dependent constants are initialized in the subroutine BLOCK DATA. The currently set values are acceptable for the three compilers on an IBM PC or clone, but may need to be reinitialized if PHAZE is used on a different kind of computer. The current values of the three fundamental constants are

TINY = 1.2E-38, set to be slightly larger than the minimum positive real number,

SMALL = 2.4E-7, such that 1. + SMALL approximately equals 1. in single precision,

HUGE = 3.E38, a number almost as large as the maximum possible number.

A few other constants, simple functions of these three, are also initialized to the corresponding values.

No special libraries are required; all the necessary calculations are performed by ANSI Fortran intrinsic functions or by subprograms of PHAZE.

5.2 Calculation Methods

This section is a supplement to the explanations of the commands in Section 3.3. The material here is organized by the commands, just as Section 3.3 is. This section does not repeat material presented in Section 3.3 or in EHF, however, but only gives additional technical detail where required. Most of the equations from EHF involve a notation that is interpreted for each model (exponential,

linear, and Weibull) in Table 1 of EHF.

CC B

Part of the comparison of components for β is to determine the attained significance level corresponding to component j. The significance level is defined as the probability that a test rejects the hypothesis $H_0{:}\beta_j = \beta_{-j}$ in favor of the alternative $H_0{:}\beta_j \neq \beta_{-j}$. Two methods are used to test the hypothesis and to approximate the significance level.

When the jth component has more than one nonreplacement failure, the hypothesis is tested using the difference of the MLEs $\hat{\beta}_j$ and $\hat{\beta}_{-j}$. The attained significance level is defined as twice the smaller of the probabilities

$P(\hat{\beta}_j - \hat{\beta}_{-j} \geq \text{observed difference})$ and $P(\hat{\beta}_j - \hat{\beta}_{-j} \leq \text{observed difference})$,

where $\hat{\beta}_j$ and $\hat{\beta}_{-j}$ are understood to be random and all the components are assumed to have a common value of β. The normal approximation is used for the distribution of both $\hat{\beta}_j$ and $\hat{\beta}_{-j}$. The calculation of $\hat{\beta}_j$ uses only the data from component j, while that for $\hat{\beta}_{-j}$ uses all the data except from component j. Each MLE is based on the conditional likelihood, and is found by solving Eq. (6) of EHF, while the corresponding Fisher information I is found by evaluating Eq. (11) of EHF. The variance of each $\hat{\beta}$ is the inverse of the Fisher information. Because the two MLEs are based on independent data sets, the variance of their difference is the sum of their variances. Therefore, if all the components really have the same β,

$$(\hat{\beta}_j - \hat{\beta}_{-j}) \,/\, [I^{1}(\hat{\beta}_j) + I^{1}(\hat{\beta}_{-j})]^{1/2}$$

is approximately normal(0,1). The approximate significance level can therefore be calculated. This approximation method is poorest if $\hat{\beta}_j$ is based on only two failures or if $\hat{\beta}_{-j}$ is based on few failures.

When the jth component has only one nonreplacement failure, the normal approximation is clearly inadequate. The approximation used instead is to treat $\hat{\beta}_{-j}$ as known. This is reasonable if $\hat{\beta}_{-j}$ is based on many more failures than $\hat{\beta}_j$ is, for in this case $\hat{\beta}_{-j}$ has much less variability than does $\hat{\beta}_j$. The single failure time t has a conditional density given below Eq. (5′) of EHF. The test rejects H_0 if t is far from the value expected assuming that β_j equals the known $\hat{\beta}_{-j}$. Therefore the attained significance level is twice the smaller of the probabilities

$P(t \geq \text{observed value})$ and $P(t \leq \text{observed value})$,

where t is understood to be random and governed by $\beta_j = \hat{\beta}_{-j}$. This approximation method is poorest if $\hat{\beta}_{-j}$ is based on very few failures.

CD

Centering the data means that all times are measured from a point t_{mid} in the middle of the observation periods. To define t_{mid} precisely, let r_j be the time range for the jth component, that is, the end time of the observation period minus the starting time of the observation period. Let $\bar{s}_j$ be the midpoint of the observation period for the jth component. Define the value $t_{mid} = \Sigma r_j \bar{s}_j \,/\, \Sigma r_j$. (Note, when there is just one component, t_{mid} reduces to the midpoint of the component's observation period.) The data are called centered if all times are measured from t_{mid}. Failures that occurred before t_{mid} are then counted as having occurred at negative times, which is allowed. This centering takes place within the calculations and is not seen by the user; times are still requested from and reported to the user in their original units.

Centering the data at some t_c changes the meaning of parameters, and can effect the allowed values of those parameters. For example, suppose that a single component is observed from times s_0 to s_1, with $0 \le s_0 \le s_1$, and consider first the case when the data are not centered. The linear hazard function is $\lambda_0(1+\beta t)$, and λ_0 is the value of $\lambda(0)$. The hazard function $\lambda(t)$ must be positive for $s_0 \le t \le s_1$; this means that β must satisfy $\beta > -1/s_1$. When instead the data are centered at t_c, the linear hazard function, with t written in the original units, is $\lambda_0[1+\beta(t-t_c)]$. In this case, λ_0 is the value of $\lambda(t_c)$. If t is between s_0 and s_1, the requirement that $\lambda(t)$ remain positive for $s_0 < t < s_1$ translates to $-1/(s_1 - t_c) < \beta < 1/(t_c - s_0)$. This is the reason why Figures 7 and 8, based on the same data set, are so different from each other.

CI

The procedure depends on the values of certain switches, as shown in Table 2. First consider a confidence interval for β when the full-likelihood and bivariate-asymptotic-normality switches are both off. As explained in Section 5.1 of EHF, define the standardized score SS by

$SS = L'(\beta_0) \,/\, [I(\beta_0)]^{1/2}$,

where L is the conditional log-likelihood, and I is the information for β based on the conditional likelihood. SS is approximately normal(0,1) for large samples. Therefore an approximate confidence interval for β is the set of all β_0 such that $-c < SS < c$, where c is the appropriate number from a normal table, For example, $c = 1.645$ yields an approximate 90% confidence interval.

A potential difficulty is that this may define a confidence set that is not an interval. This can happen if SS is not monotone in β_0. PHAZE finds a confidence interval by first acting as if SS is monotone. Beginning at $\beta = 0$, PHAZE searches through larger or smaller values until it finds a value β_{UP} where $SS = c$. Similarly it finds β_{LO} where $SS = -c$. This is reported as the confidence interval, but first PHAZE does some checking of monotonicity. It goes out beyond the upper limit, a distance 5 times $(\beta_{UP} - \hat{\beta})$ above $\hat{\beta}$, and similarly below the lower limit. Of course it never goes beyond any theoretical bounds of β. In this range, PHAZE looks at evenly spaced values of β, evaluates SS there, and decides if SS appears monotone. If not, PHAZE prints a warning, and offers to print the values of SS so that the user can examine them. Non-monotonicity is common under the linear model when the confidence intervals are infinite. Therefore the warning message has been suppressed with the linear model; it becomes a nuisance to see. Under the Weibull model, non-monotonicity may occur near the theoretical lower bound of $\beta = -1$. Non-monotonicity has not yet been observed with the exponential model.

For λ_0, when the bivariate-asymptotic-normality switch is off, a confidence interval is found, given the assumed value β_0 introduced in Section 3.2. The confidence limits use exact formulas given below, not normal approximations. In EHF a quantity v is defined, depending on the model and on β. Let n be the number of failures for time censored data (all non-replacement failures), and let m be the number of failures for failure censored data (including the replacement failures). The confidence limits on λ_0 for time censored data are then

$$\chi^2_{2n,\alpha/2}/(2v)$$
$$\chi^2_{2(n+1),(1-\alpha/2)}/(2v)$$

while for failure censored data the confidence limits are

$$\chi^2_{2m,\alpha/2}/(2v)$$
$$\chi^2_{2m,(1-\alpha/2)}/(2v) \ .$$

The confidence limits for the two cases agree except for the degrees of freedom. In PHAZE, it is possible for some components to be time censored while others are failure censored. PHAZE uses twice the total number of failures as the degrees of freedom for the lower limit. If there are any failure censored components, PHAZE uses this same number as the degrees of freedom for the upper limit; if instead all the data are time censored, PHAZE uses $2(n+1)$, where n is the number of failures. These assign-

ments agree with the correct values if all the data have one kind of censoring, and appear to be conservative for the other cases.

For (β, $\log\lambda_0$) when the full-likelihood and bivariate-asymptotic-normality switches are both on, a confidence ellipse is based on the bivariate asymptotic normality of the pair ($\hat{\beta}$, $\log\hat{\lambda}_0$). The asymptotic covariance matrix is the inverse of the Fisher information matrix, which is estimated by the sample information matrix given by Equation (15) of EHF.

TE

This test uses the approximate normality of the standardized score *SS*, introduced above for the command CI. The null hypothesis is H_0: $\beta = \beta_0$, and the alternative is H_1: $\beta \geq \beta_0$. The test rejects H_0 at significance level α if the two-sided $100(1-2\alpha)\%$ confidence interval for β is entirely to the right of β_0, or equivalently, if the one-sided 100(1-α)% confidence interval (β_{LO}, ∞) is entirely to the right of β_0. See the discussion below Expression (12) of EHF, and the examples in Section 5.3 of EHF.

5.3 Some Details of the Coding

This section is for the benefit of anyone responsible for maintaining or modifying the source code. Most of the notation follows that of EHF. For example the two-dimensional array T stores the failure times t_{ij}; R0 stores the observation ranges r_{0j}, and so forth. A difference is that the times in T for any component are measured from the beginning of the component's observation period, not from the component's installation. This has historical rather than rational reasons, and is a minor quirk that is not worth the trouble of changing. The rest of this section mentions a few ways in which the coding does not have the straightforward, obvious meaning.

In writing PHAZE, much of the coding effort was concerned with preventing numerical overflows or underflows. In typical examples with the exponential or linear hazard models, the values of time are on the order of 10^4 hours or more, and β is on the order of 10^{-5} hours^{-1} or less. Therefore, the numbers are scaled internally, in order to keep the magnitudes nearer to 1.0. After a driver program obtains a new data set by calling the subroutine GETDAT, the driver calls the subroutine PREP, to prepare the data for analysis. PREP looks at the observation range, i.e., the length of the observation period, for each component. It then defines TNORM as the maximum of these ranges, and divides

every time (failure time, observation beginning time, and observation end time) by TNORM. This is effectively a change in the units of time. These renormalized times are the basis of the values stored internally and used for calculations. This means that for the exponential and linear hazard models we have

BETA = TNORM $\times \beta$,

where β is in the original units and BETA is in the units for internal calculations. The function of TNORM is invisible to the user: all times and parameters are reported to the user in the original units.

For the Weibull model, t_0 has the value of TNORM. The command SN can be used with the Weibull model to reset TNORM to any user-specified value. It works by calling PREP, which sets TNORM to the user-specified value and then divides all times (restored to their original input values) by the new TNORM.

Under the Weibull model, when β is near -1 expressions must sometimes be evaluated using asymptotic approximations in terms of $(\beta + 1)$, as given by Table 1 of EHF. To avoid round-off error in these cases, the value of $\beta + 1$ is stored internally, rather than the value of β. That is, we have

BETA $= \beta + 1$,

where β is in the original units and BETA is the value stored internally, when the Weibull model is in effect. This change was added late in the development of PHAZE, which is why the variable is called BETA; some subroutines use the less confusing name BARG (for "β-argument").

No problem is caused by giving BETA different meanings depending on the model in effect, because whenever a new model is selected, the default value β_0 is reset to 0. That is, the internal variable BETA0 is set to 0.0 if the exponential or linear model is in effect, and to 1.0 if the Weibull model is in effect. Statistical inference then proceeds using the internal units, and all output is translated into the original units.

6. REFERENCES

F. B. Alt, "Bonferroni Inequalities and Intervals," *Encyclopedia of Statistical Sciences*, Vol. 1, S. Kotz and N. L. Johnson, eds., New York: John Wiley & Sons, 1982.

Corwin L. Atwood, *Estimating Hazard Functions for Repairable Components*, EGG-SSRE-8972, May 1990. (Referred to as EHF in this user's guide.)

IEEE, *IEEE Standard for Software Quality Assurance Plans*, ANSI/IEEE Std 730-1984, 1984.

Minitab Reference Manual, Release 5.1, State College, PA: Minitab, Inc., 1985.

Ronald D. Snee and Charles G. Pfeifer, "Graphical Representation of Data," *Encyclopedia of Statistical Sciences*, Vol. 3, S. Kotz and N. L. Johnson, eds., New York: John Wiley & Sons, 1983.

Andrew J. Wolford, Corwin L. Atwood, and W. Scott Roesener, *Aging Risk Assessment Methodology: Development and Demonstration Study*, NUREG/CR-5378, EGG-2567, DRAFT, Rev. 1, June 1990.

Stephen Wolfram, *Mathematica: A System for Doing Mathematics by Computer*, Redwood City, CA: Addison-Wesley Publishing Company, 1988.

APPENDIX
SAMPLE PHAZE SESSION

APPENDIX

SAMPLE PHAZE SESSION

The following printout is the echo of a PHAZE session. The very first command of the session was

```
ou exampl.out
```

This caused all the remaining lines to be written to the disk output file EXAMPL.OUT, which was later captured electronically and inserted into this report. The very first lines of the session, before the command OU could be entered, appeared on the terminal screen but were not echoed to a disk file. Those few lines were copied manually to the beginning of the file, so that the reader of this report can see how the entire session proceeded.

Anything that was entered by the user is printed here in **bold face**. This highlighting is added in this report, to make it easier for the reader to distinguish between commands by the user and responses by the program.

The spacing of the program output has sometimes been changed slightly for reproduction here. This is because the output of PHAZE can be up to 80 characters wide, while the word processor being used for this report allows fixed-spacing lines only of 76 characters. Also, the output from the program was continuous, scrolling up the screen. In this appendix, however, page-breaks have been inserted at logical places rather than at mechanical places after a fixed number of lines.

```
C:\FORTRAN>phaze

Program PHAZE, for Parametric HAZard Estimation
Version 1.00
Enter command when prompted by '->'
For help, enter H
-> ou exampl.out
-> ################
-> # This is a sample PHAZE session, with comments sprinkled
-> # liberally through the session.
-> # We will follow the steps (1) through (10) of Section 1.3.
-> # First, however, we demonstrate the help commands
-> ################
-> h
Help messages are given for groups of commands

Enter     For commands that
-----    ----------------------------
 H D     Control Data or files
 H I     Perform statistical Inference
 H O     Set or show Options
-------------------------------------
To quit the program, enter Q
-> hd
 Commands that control data or files are
 Command    Meaning
 -------    --------------------
 DD         Describe Data (summary statistics)
 GD x       Get Data
 OU x       define disk file for OUtput
 PF         toggle option to allow creation of Plot Files.
               Default is OFF
 SD         See Data in concise display
 #          comment only (echoed to output file)
x is an (optional) DOS file name.
If none is given, you will be prompted.
-> h i
In the commands below, x identifies the parameter(s) of interest.
Replace x by B for inference on beta
          by L for inference on lambda0
          by + for inference on both
       or by H for inference on the hazard function
 Commands that perform statistical inference are
 Command    Meaning
 -------    --------------------
 CC x       Compare Components for x values
 CI x k     find k% Confidence Interval for x, where
              for example k = 90, 95, or another value
 HF t       find Hazard Function at time t (MLE and 90% int.)
 HF R t1 t2 find Hazard Function at times in Range t1-t2
 KS         Kolmogorov-Smirnov test of model
 ML x       find Maximum Likelihood estimator for x
```

```
 QQ          calculate info for Q-Q plot
 SB y        Set Beta0 to y, assumed value for tests or QQ plot or lambda
               calculations.  Default is 0; becomes MLE after ML command.
 TE y        TEst whether beta = y.  If no value is given for y,
               beta0 is used.
x retains its most recent value.
For example, following the command ML B, the command CC is interpreted as
CC B.

-> h o
 Commands that set options are
 Command     Meaning
 -------     --------------------
 AN          Toggle option to treat MLE of (beta, log-lambda0)
               as jointly Asymptotically Normal.
               Default is asymptotic normality only for beta-hat
 CD          Toggle option to Center Data (with linear hazard function
               only).
               Default is no centering.
 FL          Toggle option to use Full Likelihood.
               Default is to use conditional likelihood for inference about
               beta
 LE          Toggle option to assume Lambda0s Equal for all components.
               Default is not to assume equality.
 SM x        Select Model x, where x is E, L, or W for exponential,
               linear, or Weibull hazard function.  There is no default.
 SN x        Set Normalizing time to x, for Weibull model.
               If x is missing, default time is used.
 ST          See current STatus of options.
-> #############
-> # Step 1
-> #############
-> gd movdemo.dat

Data found for  16 components.
     0 warnings
     0 errors encountered in data
-> # This data set is shown in Figure 1 of the text
-> dd
```

Component	Nonrepl. Fails.	Observed Hrs.	Mean Failure Time (Normed)	Replaced on Fail.?	Initial Age
MOV-1A	1	8.8584E+04	0.378		4.1448E+04
MOV-1B	1	8.8584E+04	0.086		4.1448E+04
MOV-1C	2	8.8584E+04	0.752		4.1448E+04
MOV-1D	7	8.8584E+04	0.743		4.1448E+04
MOV-1E	0	2.1840E+04		Y	4.1448E+04
MOV-1E(R)	3	6.6744E+04	0.498		0.0000E-01
MOV-1F	3	4.3608E+04	0.568	Y	4.1448E+04
MOV-1F(R)	1	4.4976E+04	0.487		0.0000E-01
MOV-2A	4	8.8584E+04	0.619		3.7824E+04
MOV-2B	5	8.8584E+04	0.567		3.7824E+04

```
MOV-2C .            1     4.9728E+04     0.756              Y    3.7824E-04
MOV-2C(R)           1     3.8856E+04     0.866                   0.0000E-01
MOV-2D              6     8.8584E+04     0.464                   3.7824E-04
MOV-2E              0     2.2608E+04                             3.7824E-04
MOV-2E(R)           2     6.5976E+04     0.698                   0.0000E-01
MOV-2F              7     8.8584E+04     0.593                   3.7824E+04
-> sd

The data set contains the following  16 components

MOV-1A
Between component ages  4.1448E+04 and  1.3003E+05, there was 1 failure :
    7.4916E+04
Mean (non-replacement) failure age =  7.4916E+04

MOV-1B
Between component ages  4.1448E+04 and  1.3003E+05, there was 1 failure :
    4.9044E+04
Mean (non-replacement) failure age =  4.9044E+04

MOV-1C
Between component ages  4.1448E+04 and  1.3003E+05, there were  2 failures:
    9.1092E+04    1.2496E+05
Mean (non-replacement) failure age =  1.0802E+05

MOV-1D
Between component ages  4.1448E+04 and  1.3003E+05, there were  7 failures:
    9.0804E+04    9.1740E+04    1.0127E+05    1.1135E+05    1.1536E+05
    1.1543E+05    1.2496E+05
Mean (non-replacement) failure age =  1.0727E+05

MOV-1E
Between component ages  4.1448E+04 and  6.3288E+04, there were 0 failures
One more failure, at age  6.3288E+04, resulted in component's replacement

MOV-1E(R)
Between component ages  0.0000E-01 and  6.6744E+04, there were  3 failures:
    1.1460E+04    2.6820E+04    6.1356E+04
Mean (non-replacement) failure age =  3.3212E+04

MOV-1F
Between component ages  4.1448E+04 and  8.5056E+04, there were  3 failures:
    4.8300E+04    7.2876E+04    7.7436E+04
Mean (non-replacement) failure age =  6.6204E+04
One more failure, at age  8.5056E+04, resulted in component's replacement

MOV-1F(R)
Between component ages  0.0000E-01 and  4.4976E+04, there was 1 failure :
    2.1924E+04
Mean (non-replacement) failure age =  2.1924E+04

MOV-2A
```

```
Between component ages  3.7824E+04 and  1.2641E+05, there were  4 failures:
    4.7844E+04    1.0655E+05    1.0660E+05    1.0955E+05
Mean (non-replacement) failure age =  9.2634E+04

MOV-2B
Between component ages  3.7824E+04 and  1.2641E+05, there were  5 failures:
    6.4188E+04    6.5868E+04    8.4444E+04    1.0640E+05    1.1917E+05
Mean (non-replacement) failure age =  8.8015E+04

MOV-2C
Between component ages  3.7824E+04 and  8.7552E+04, there was 1 failure :
    7.5420E+04
Mean (non-replacement) failure age =  7.5420E+04
One more failure, at age  8.7552E+04, resulted in component's replacement

MOV-2C(R)
Between component ages  0.0000E-01 and  3.8856E+04, there was 1 failure :
    3.3636E+04
Mean (non-replacement) failure age =  3.3636E+04

MOV-2D
Between component ages  3.7824E+04 and  1.2641E+05, there were  6 failures:
    4.3260E+04    6.1668E+04    6.2148E+04    8.4444E+04    1.0640E+05
    1.1576E+05
Mean (non-replacement) failure age =  7.8948E+04

MOV-2E
Between component ages  3.7824E+04 and  6.0432E+04, there were 0 failures

MOV-2E(R)
Between component ages  0.0000E-01 and  6.5976E+04, there were  2 failures:
    4.5972E+04    4.6068E+04
Mean (non-replacement) failure age =  4.6020E+04

MOV-2F
Between component ages  3.7824E+04 and  1.2641E+05, there were  7 failures:
    6.1572E+04    8.4444E+04    8.7492E+04    9.0300E+04    9.5988E+04
    1.0640E+05    1.0655E+05
Mean (non-replacement) failure age =  9.0393E+04
-> # Often, DD is enough, with no need for the long output of SD
-> #############
-> # Step 2, Construction of a cumulative failure plot
-> #############
-> qq
Enter model type - Exponential failure rate
                   Linear failure rate
                   Weibull failure rate

-> e
Now re-enter command
-> # I forgot to enter the model, so PHAZE prompted me
-> # Now we do the cumulative failure plot
```

-> qq
QQ information when beta = 0.000E-01
based on conditional distribution of non-replacement failures

Calculated t	Observed t
7.93E+03	1.15E+04
1.59E+04	2.19E+04
2.38E+04	2.68E+04
3.17E+04	3.36E+04
3.84E+04	4.33E+04
4.11E+04	4.60E+04
4.29E+04	4.61E+04
4.47E+04	4.78E+04
4.65E+04	4.83E+04
4.83E+04	4.90E+04
5.01E+04	6.14E+04
5.19E+04	6.16E+04
5.38E+04	6.17E+04
5.56E+04	6.21E+04
5.74E+04	6.42E+04
5.92E+04	6.59E+04
6.10E+04	7.29E+04
6.29E+04	7.49E+04
6.47E+04	7.54E+04
6.66E+04	7.74E+04
6.87E+04	8.44E+04
7.08E+04	8.44E+04
7.29E+04	8.44E+04
7.51E+04	8.75E+04
7.72E+04	9.03E+04
7.93E+04	9.08E+04
8.14E+04	9.11E+04
8.35E+04	9.17E+04
8.58E+04	9.60E+04
8.82E+04	1.01E+05
9.08E+04	1.06E+05
9.35E+04	1.06E+05
9.61E+04	1.06E+05
9.87E+04	1.07E+05
1.01E+05	1.07E+05
1.04E+05	1.07E+05
1.07E+05	1.10E+05
1.09E+05	1.11E+05
1.12E+05	1.15E+05
1.14E+05	1.15E+05
1.17E+05	1.16E+05
1.20E+05	1.19E+05
1.22E+05	1.25E+05
1.25E+05	1.25E+05

QQ information when beta = 0 and all lambda0 values assumed equal
based on observation times rather than component ages
This is relevant if the number of components was the same at all times

Calculated t	Observed t
1.85E+03	5.44E+03
3.69E+03	6.85E+03
5.54E+03	7.60E+03
7.38E+03	1.00E+04
9.23E+03	2.18E+04
1.11E+04	2.37E+04
1.29E+04	2.38E+04
1.48E+04	2.43E+04
1.66E+04	2.64E+04
1.85E+04	2.80E+04
2.03E+04	3.14E+04
2.21E+04	3.33E+04
2.40E+04	3.35E+04
2.58E+04	3.60E+04
2.77E+04	3.76E+04
2.95E+04	4.36E+04
3.14E+04	4.66E+04
3.32E+04	4.66E+04
3.51E+04	4.66E+04
3.69E+04	4.87E+04
3.88E+04	4.94E+04
4.06E+04	4.96E+04
4.24E+04	4.97E+04
4.43E+04	4.97E+04
4.61E+04	5.03E+04
4.80E+04	5.25E+04
4.98E+04	5.82E+04
5.17E+04	5.98E+04
5.35E+04	6.55E+04
5.54E+04	6.86E+04
5.72E+04	6.86E+04
5.91E+04	6.86E+04
6.09E+04	6.86E+04
6.27E+04	6.87E+04
6.46E+04	6.87E+04
6.64E+04	6.87E+04
6.83E+04	6.88E+04
7.01E+04	6.99E+04
7.20E+04	7.17E+04
7.38E+04	7.39E+04
7.57E+04	7.40E+04
7.75E+04	7.79E+04
7.94E+04	8.13E+04
8.12E+04	8.32E+04
8.30E+04	8.34E+04
8.49E+04	8.35E+04
8.67E+04	8.35E+04

```
-> # The second set of output is what is needed for a cumulative failure
-> #   plot.  It is only generated when beta = 0.
-> # If the PF switch had been on, PHAZE would have offered to write
-> #   the output to a disk file, for us to use with a graphics package.
-> ###############
-> # Step 3, compare components for common beta
-> ###############
-> ccb
 95.0% conf. intervals with MLEs for beta, by component and MLE without
comp.
MOV-1A          (-1.10E-04,-1.72E-05, 6.48E-05)          1.42E-05
MOV-1B          (-4.85E-04,-1.32E-04, 2.55E-05)          1.54E-05
MOV-1C          (-1.77E-05, 4.09E-05, 1.08E-04)          1.21E-05
MOV-1D          ( 4.06E-06, 3.90E-05, 7.52E-05)          8.57E-06
MOV-1E            not estimable
MOV-1E(R)       (-5.41E-05,-4.31E-07, 5.32E-05)          1.41E-05
MOV-1F          (-6.39E-05, 1.88E-05, 1.03E-04)          1.33E-05
MOV-1F(R)       (-1.67E-04,-3.35E-06, 1.59E-04)          1.35E-05
MOV-2A          (-2.04E-05, 1.67E-05, 5.46E-05)          1.30E-05
MOV-2B          (-2.35E-05, 9.12E-06, 4.20E-05)          1.40E-05
MOV-2C          (-8.71E-05, 7.48E-05, 3.04E-04)          1.29E-05
MOV-2C(R)       (-7.88E-05, 1.91E-04, 6.82E-04)          1.28E-05
MOV-2D          (-3.47E-05,-4.86E-06, 2.49E-05)          1.71E-05
MOV-2E            not estimable
MOV-2E(R)       (-3.18E-05, 3.99E-05, 1.19E-04)          1.26E-05
MOV-2F          (-1.55E-05, 1.29E-05, 4.16E-05)          1.35E-05
OVERALL         ( 5.32E-07, 1.34E-05, 2.62E-05)

Comparison of beta values for components
Component     Signif    95.0% Confidence Interval
MOV-1A        0.48                    (----*I-)
MOV-1B        0.09  (---------------*------I)
MOV-1C        0.44                        (I*--)
MOV-1D        0.14                         I*-)
MOV-1E
MOV-1E(R)     0.64                      (-*I-)
MOV-1F        0.91                     (---*---)
MOV-1F(R)     0.82                 (------*I------)
MOV-2A        0.87                       (*-)
MOV-2B        0.80                      (-*)
MOV-2C        0.61                   (----I--*----------)
MOV-2C(R)     0.33                    (---I-------*----------------------)
MOV-2D        0.21                      (*I)
MOV-2E
MOV-2E(R)     0.54                      (-I*---)
MOV-2F        0.97                       (*)

OVERALL                                 (*)
Overall significance level for testing equality of betas = 1.00

Signif for jth component means estimated probability that
beta-hat(j) - beta-hat(others) is at least as extreme as observed
```

```
Overall significance level is P[at least one signif(j) <= observed min].
and is computed with Bonferroni inequality
-> # Overall significance is 1.00 -- no evidence of different beta's
-> ###################
-> # Step 4, test for increasing failure rate
-> ###################
-> te 0
For testing beta =  0.000E-01, test statistic =   2.04
Approx. significance level, for testing against beta >  0.000E-01, is 0.021
Significance level is based on normal approximation
and distribution conditional on observed counts
-> # Small significance, i.e. strong evidence of increasing failure rate
-> # Construct confidence interval to supplement the test
-> ml b
Maximum Likelihood Estimate beta-hat is  1.336E-05
-> ci b 90
90.0% confidence interval for beta is
         ( 2.56E-06, 2.42E-05)
This is based on the normal approximation,
and the likelihood conditional on the number of nonreplacement failures.
-> ##################
-> # Step 5, check goodness of fit of model to data
-> ##################
-> ks
For testing model with beta =  1.336E-05
Kolmogorov-Smirnov test statistic is 0.090
Value of test statistic is attained at observation   41
Significance level equals 0.869, based on  44 observed failures
-> # For the following Q-Q plot info, I will turn on the plot-file switch.
-> # The resulting disk file is identical to the one used to produce
-> #    Figure 9 in the text, using a commercial graphics package.
-> pf
Option to allow creation of plot-files turned ON
-> qq
Plot information can be written to a file
Do you want a separate file for this?

-> y
Please enter DOS filename
-> qqdemo.out
Please enter identifier to copy to qqdemo.out

-> Q-Q plot info, exponential model, data from movdemo.dat
QQ information when beta =  1.336E-05
based on conditional distribution of non-replacement failures

 Calculated t      Observed t
--------------  --------------
     1.10E+04        1.15E+04
     2.06E+04        2.19E+04
     2.91E+04        2.68E+04
```

```
     3.67E+04          3.36E+04
     4.10E+04          4.33E+04
     4.37E+04          4.60E+04
     4.62E+04          4.61E+04
     4.88E+04          4.78E+04
     5.13E+04          4.83E+04
     5.37E+04          4.90E+04
     5.61E+04          6.14E+04
     5.84E+04          6.16E+04
     6.06E+04          6.17E+04
     6.27E+04          6.21E+04
     6.49E+04          6.42E+04
     6.70E+04          6.59E+04
     6.95E+04          7.29E+04
     7.20E+04          7.49E+04
     7.45E+04          7.54E+04
     7.68E+04          7.74E+04
     7.90E+04          8.44E+04
     8.13E+04          8.44E+04
     8.33E+04          8.44E+04
     8.55E+04          8.75E+04
     8.80E+04          9.03E+04
     9.05E+04          9.08E+04
     9.31E+04          9.11E+04
     9.55E+04          9.17E+04
     9.78E+04          9.60E+04
     1.00E+05          1.01E+05
     1.02E+05          1.06E+05
     1.04E+05          1.06E+05
     1.06E+05          1.06E+05
     1.08E+05          1.07E+05
     1.10E+05          1.07E+05
     1.12E+05          1.07E+05
     1.14E+05          1.10E+05
     1.16E+05          1.11E+05
     1.18E+05          1.15E+05
     1.19E+05          1.15E+05
     1.21E+05          1.16E+05
     1.23E+05          1.19E+05
     1.25E+05          1.25E+05
     1.26E+05          1.25E+05
-> # The identifier is written as the first line of qqdemo.out
-> # Now, because this is only a demonstration, we turn off the
-> #   plot-file option
-> pf
Option to allow creation of plot-files turned OFF
-> ###############
-> # Step 6, compare the components for a common lambda0
-> ###############
```

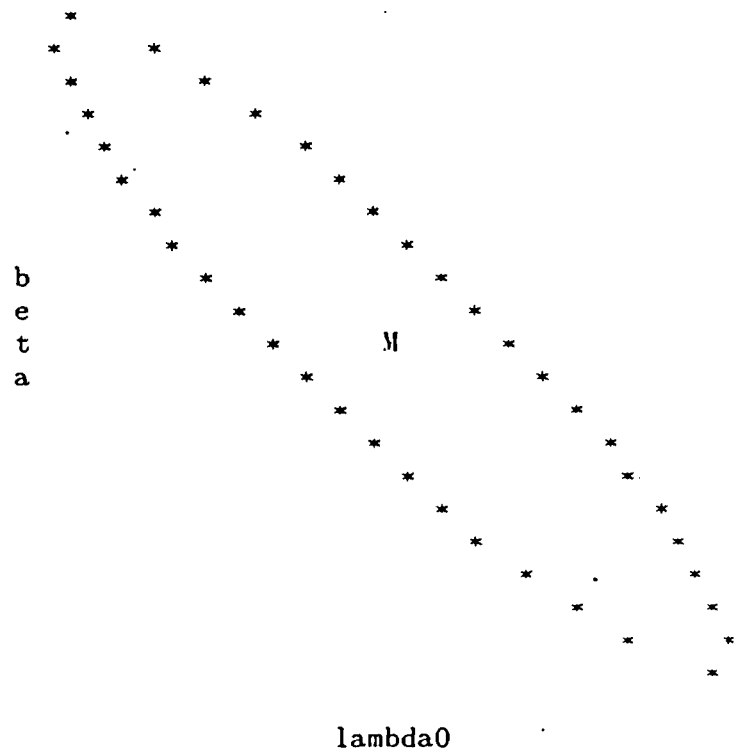
```
-> ccl
 95.0% conf. intervals with MLEs for lambda0, by component and MLE without
comp.
     assuming beta = 1.34E-05
MOV-1A          ( 8.58E-08, 3.39E-06, 1.89E-05)          1.74E-05
MOV-1B          ( 8.58E-08, 3.39E-06, 1.89E-05)          1.74E-05
MOV-1C          ( 8.20E-07, 6.78E-06, 2.45E-05)          1.70E-05
MOV-1D          ( 9.54E-06, 2.37E-05, 4.89E-05)          1.51E-05
MOV-1E          ( 5.74E-07, 2.27E-05, 8.36E-05)          1.59E-05
MOV-1E(R)       ( 5.75E-06, 2.78E-05, 8.14E-05)          1.55E-05
MOV-1F          ( 1.06E-05, 3.88E-05, 8.51E-05)          1.52E-05
MOV-1F(R)       ( 4.11E-07, 1.62E-05, 9.03E-05)          1.60E-05
MOV-2A          ( 3.88E-06, 1.42E-05, 3.64E-05)          1.62E-05
MOV-2B          ( 5.78E-06, 1.78E-05, 4.15E-05)          1.58E-05
MOV-2C          ( 2.07E-06, 1.71E-05, 4.76E-05)          1.59E-05
MOV-2C(R)       ( 4.97E-07, 1.96E-05, 1.09E-04)          1.59E-05
MOV-2D          ( 7.83E-06, 2.13E-05, 4.64E-05)          1.54E-05
MOV-2E          ( 0.00    , 0.00    , 8.43E-05)          1.62E-05
MOV-2E(R)       ( 2.29E-06, 1.89E-05, 6.82E-05)          1.59E-05
MOV-2F          ( 1.00E-05, 2.49E-05, 5.13E-05)          1.50E-05
OVERALL         ( 1.17E-05, 1.60E-05, 2.13E-05)

Comparison of lambda0 values for components, assuming beta = 1.34E-05

Component     Signif    95.0% Confidence Interval
MOV-1A        0.09  (-*------I
MOV-1B        0.09  (-*------I
MOV-1C        0.23  (--*----I---)
MOV-1D        0.32       (-I----*-----------)
MOV-1E        1.00  (-------I--*------------------------------)
MOV-1E(R)     0.42     (----I-----*-----------------------)
MOV-1F        0.15       (-I-----------*----------------------)
MOV-1F(R)     1.00  (-------*-------------------------------------)
MOV-2A        1.00    (----*I---------)
MOV-2B        0.80     (----I*-----------)
MOV-2C        0.90   (------*---------------)
MOV-2C(R)     1.00  (-------I-*-------------------------------------------)
MOV-2D        0.45      (---I--*-----------)
MOV-2E        0.65  *-------I--------------------------------)
MOV-2E(R)     1.00   (------I*------------------------)
MOV-2F        0.21       (-I----*------------)

OVERALL                    (-*--)
Overall significance level for testing equality of lambda0s = 1.00

Signif for jth component means estimated probability that
lambda0-hat(j) - lambda0-hat(others) is at least as extreme as observed

Overall significance level is P[at least one signif(j) <= observed min],
and is computed with Bonferroni inequality
-> # No evidence of different values of lambda0
-> # Before I do the next step, I check what options are in effect
```

```
-> st
Model type:  Exponential failure rate
Components are not assumed to have a common value of lambda0.
Conditional likelihood is used for inference about beta.
Inference for lambda0 and beta does not use joint asymptotic normality.
The data are from the file movdemo.dat
The session is being echoed to file exampl.out
-> ################
-> # Step 7, Generate 2-dimensional confidence region.
-> # If PF switch were on, the resulting disk file could be used to
-> #   generate one of the regions plotted in Figure 5 of the text.
-> ################
-> ci + 90
Setting switch to assume lambda0s equal for all components

Joint 90.0% confidence region for (lambda0,beta)
with lambda0 plotted on log scale

 beta          lambda0
2.62E-05  (3.78E-06, 6.84E-06)    **********
2.49E-05  (4.26E-06, 7.70E-06)      *        *
2.36E-05  (4.79E-06, 8.67E-06)        *       *
2.24E-05  (5.38E-06, 9.75E-06)         *        *
2.11E-05  (6.04E-06, 1.09E-05)           *        *
1.98E-05  (6.78E-06, 1.23E-05)             *       *
1.85E-05  (7.59E-06, 1.37E-05)              *        *
1.72E-05  (8.48E-06, 1.53E-05)                *        *
1.59E-05  (9.46E-06, 1.71E-05)  b               *       *
1.47E-05  (1.05E-05, 1.91E-05)  e                *        *
1.34E-05  (1.17E-05, 2.12E-05)  t                  *   M    *
1.21E-05  (1.30E-05, 2.36E-05)  a                   *        *
1.08E-05  (1.45E-05, 2.62E-05)                        *        *
9.52E-06  (1.60E-05, 2.90E-05)                         *        *
8.24E-06  (1.77E-05, 3.21E-05)                           *        *
6.95E-06  (1.96E-05, 3.54E-05)                            *        *
5.67E-06  (2.16E-05, 3.90E-05)                             *       *
4.38E-06  (2.37E-05, 4.29E-05)                              *        *
3.10E-06  (2.61E-05, 4.72E-05)                               *       *
1.82E-06  (2.86E-05, 5.17E-05)                                *        *
5.32E-07  (3.13E-05, 5.67E-05)                                 **********

                                                  lambda0

-> # Note, the above command caused PHAZE to turn on the
-> #   lambdas-equal switch.
-> # Note also, a 90% conf. interval for lambda0 is
-> #   (3.78E-06, 5.67E-05)
-> # By contrast, the CI command gives a confidence interval, assuming
-> #   that the value of beta is the MLE beta-hat, as follows
-> ci 1 90
90.0% confidence interval for lambda0, assuming beta=1.336E-05, is
        ( 1.24E-05, 2.04E-05)
```

```
-> # This interval is much shorter than the range of lambda0 in the
-> #  .2-dimensional region, because it assumes a value of beta.
-> ###################
-> # From now on, we use bivariate asymptotic normality of
-> #   (beta-hat, log lambda-hat)
-> ###################
-> an
Using asymptotic normality of MLE of (beta, log-lambda0)
Setting switch to do joint inference with full likelihood
-> # Note, turning on the AN switch also turned on the FL switch
-> st
Model type:  Exponential failure rate
Components assumed to have a common value of lambda0.
Full likelihood is used for inference about beta and lambda0(s).
Inference for lambda0 and beta uses joint asymptotic normality
   of MLEs of beta and log lambda0.
The data are from the file movdemo.dat
The session is being echoed to file exampl.out
-> #################
-> # Step 8, Find MLE
-> #################
-> ml +
Maximum Likelihood Estimate beta-hat is  1.071E-05
Maximum Likelihood Estimate lambda0-hat is  1.984E-05
assuming that beta =  1.071E-05
Estimated Fisher information matrix of (beta, log-lambda0)
  3.469E+11   3.765E+06
  3.765E+06    47.0
Estimated covariance matrix of MLE of (beta, log-lambda0)
  2.208E-11  -1.769E-06
 -1.769E-06   0.163
-> ###################
-> # Step 9, Investigate adequacy of bivariate normal approximation
-> # If the PF switch were on for the next command, the resulting
-> #   disk file could be used to generate the ellipse in Figure 5
-> #   of the text.
-> ###################
-> ci + 90
```

Joint 90.0% confidence region for (lambda0,beta)
with lambda0 plotted on log scale

beta	lambda0
2.08E-05	(8.85E-06, 8.85E-06)
1.98E-05	(8.37E-06, 1.10E-05)
1.88E-05	(8.62E-06, 1.25E-05)
1.78E-05	(9.01E-06, 1.41E-05)
1.68E-05	(9.51E-06, 1.57E-05)
1.58E-05	(1.01E-05, 1.74E-05)
1.47E-05	(1.08E-05, 1.91E-05)
1.37E-05	(1.16E-05, 2.10E-05)
1.27E-05	(1.24E-05, 2.29E-05)
1.17E-05	(1.34E-05, 2.50E-05)
1.07E-05	(1.45E-05, 2.71E-05)
9.70E-06	(1.58E-05, 2.94E-05)
8.69E-06	(1.72E-05, 3.17E-05)
7.68E-06	(1.88E-05, 3.41E-05)
6.68E-06	(2.06E-05, 3.65E-05)
5.67E-06	(2.27E-05, 3.90E-05)
4.66E-06	(2.51E-05, 4.14E-05)
3.65E-06	(2.79E-05, 4.37E-05)
2.64E-06	(3.14E-05, 4.57E-05)
1.63E-06	(3.58E-05, 4.71E-05)
6.24E-07	(4.45E-05, 4.45E-05)




```
-> # Figure 5 shows that the two regions overlay well.
-> # Therefore, we proceed with the hazard function estimation
-> #   assuming normality of the MLE
-> ##############
-> # Step 10
-> ##############
-> hf
Enter time (in hours after installation) at which hazard function
should be evaluated, or 'R' and two times for range

-> r
Enter two times (in hours after installation) defining range
   over which hazard function should be evaluated
Default is the data window 0.0000E-01 to 1.3003E+05
Enter two times or RETURN for default

->
Time range is from 0.0000E-01 to 1.3003E+05 hours
How many increments should this be divided into?
Default is 20 increments of  6502.     hours each
Enter number of increments, or RETURN for default

-> 4
```

```
MLE and approximate 90.0% conf. int. for hazard function
   Time              Haz-lo    Haz-MLE    Haz-up          Mean          E.F.
0.0000E-01       ( 1.02E-05, 1.98E-05, 3.86E-05)      2.15E-05       1.94
3.2508E+04       ( 1.81E-05, 2.81E-05, 4.36E-05)      2.91E-05       1.55
6.5016E+04       ( 3.05E-05, 3.98E-05, 5.20E-05)      4.03E-05       1.31
9.7524E+04       ( 4.28E-05, 5.64E-05, 7.43E-05)      5.72E-05       1.32
1.3003E+05       ( 5.07E-05, 7.99E-05, 1.26E-04)      8.30E-05       1.58
-> # For a real plot, I would have turned on the PF switch,
-> #   and I would have used more increments.
-> # By contrast, here is the hazard function if the AN switch is off.
-> # We reset the options by the LE command
-> le
Components not assumed to have common value for lambda0
Setting switch to use conditional likelihood for inference for beta
Setting switch to not use asymptotic normality
-> hf r
Enter two times (in hours after installation) defining range
   over which hazard function should be evaluated
Default is the data window 0.0000E-01 to 1.3003E+05
Enter two times or RETURN for default

->
Setting switch to assume lambda0s equal for all components
Time range is from 0.0000E-01 to 1.3003E+05 hours
How many increments should this be divided into?
Default is 20 increments of  6502.     hours each
Enter number of increments, or RETURN for default

-> 4
MLE and conservative 90.0% conf. int. for hazard function
   Time              Haz-lo    Haz-MLE    Haz-up
0.0000E-01       ( 3.78E-06, 1.60E-05, 5.67E-05)
3.2508E+04       ( 8.86E-06, 2.47E-05, 5.76E-05)
6.5016E+04       ( 2.08E-05, 3.81E-05, 5.86E-05)
9.7524E+04       ( 3.30E-05, 5.89E-05, 8.81E-05)
1.3003E+05       ( 3.35E-05, 9.09E-05, 2.07E-04)
-> # Each interval is conservative, based on the largest and smallest
-> #   values attained by the hazard function in the non-elliptical
-> #   2-dimensional confidence region produced above.
-> ##############
-> # Now we demonstrate a few other commands
-> ##############
-> sm 1
-> cd
Failure times centered at   69404.5     hours
```

```
-> st
Model type:  Linear failure rate
Components assumed to have a common value of lambda0.
Conditional likelihood is used for inference about beta.
Inference for lambda0 and beta does not use joint asymptotic normality.
The failure times are centered at   69404.5
The data are from the file movdemo.dat
The session is being echoed to file exampl.out
-> # The options remain as before we changed models, but the parameter
-> #   estimates are no longer in effect, until we re-estimate.
-> # This can be demonstrated by the KS command, for example.
-> ks
For testing model with beta =  0.000E-01
Kolmogorov-Smirnov test statistic is 0.178
Value of test statistic is attained at observation   21
Significance level equals 0.125, based on  44 observed failures

For testing model with beta = 0 and all lambdas equal,
based on observed failure times rather than component ages
Kolmogorov-Smirnov test statistic is 0.186
Value of test statistic is attained at observation   17
This test is relevant if the number of components was the same at all times
Significance level equals 0.078, based on  47 observed failures
-> # See, PHAZE now considers beta to equal zero, not the value
-> #   calculated as the MLE under the exponential model.
-> sm w
-> st
Model type:  Weibull failure rate
Components assumed to have a common value of lambda0.
Conditional likelihood is used for inference about beta.
Inference for lambda0 and beta does not use joint asymptotic normality.
Lambda0 is the value of the failure rate at time   88584.0
The data are from the file movdemo.dat
The session is being echoed to file exampl.out
-> # We can reset the normalizing time to the t-mid calculated for
-> #   the linear model:
-> sn 69404.5
-> st
Model type:  Weibull failure rate
Components assumed to have a common value of lambda0.
Conditional likelihood is used for inference about beta.
Inference for lambda0 and beta does not use joint asymptotic normality.
Lambda0 is the value of the failure rate at time   69404.5
The data are from the file movdemo.dat
The session is being echoed to file exampl.out
-> # Enough.  Let's quit.
-> q
```