Lehigh University
Lehigh Preserve

Theses and Dissertations

1986

Improvement of software reliability prediction :

Chongman Park
Lehigh University

Follow this and additional works at: https://preserve.lehigh.edu/etd

Part of the Industrial Engineering Commons

Recommended Citation

Park, Chongman, "Improvement of software reliability prediction :" (1986). *Theses and Dissertations*. 4721.
https://preserve.lehigh.edu/etd/4721

This Thesis is brought to you for free and open access by Lehigh Preserve. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of Lehigh Preserve. For more information, please contact preserve@lehigh.edu.

IMPROVEMENT OF SOFTWARE RELIABILITY PREDICTION : PIECEWISE WEIBULL FAILURE RATE MODEL AND S-SHAPED RELIABILITY GROWTH MODEL

BY

Chongman Park

A Thesis

Presented to the Graduate Committee

of Lehigh University

in Candidacy for the degree of

Master of Science

in

Industrial Engineering

Lehigh University

1986

CERTIFICATE OF APPROVAL

This thesis is accepted and approved in potential fulfilment of the requirement for the degree of Master of Science in Industrial Engineering.

December 17, 1986
(date)

John C. Wiginton
Professor in Charge

George E. Kane
Chairman of Department

ACKNOWLEDGEMENTS

I would like to thank Professor John C. Wiginton for his guidance and advice during the whole phase of this study.

I would also like to acknowledge department chairman George E.Kane and Professor John C. Adams for their useful suggestion.

Finally,I thank my dear wife , Kyunghee and my parents ,Namki and Taeim for their forbearance and understanding.

TABLE OF CONTENTS

ABSTRACT

This study is consentrated on the efforts to improve the quality of software reliability prediction. The quality of software reliabilty prediction depend on the selection of appropriate model and statistical procedure. Only good model is not sufficient for the good quality.

Piecewise Weibull failure rate model offers not only the judging base of model behavior prior to the application of a particular software reliability model in searching a good model but also PWF model itself might be a good model.

When the failure data with an unknown ditribution are given, PWF modle starts to judge the basic trend of data with the assumption which its distribution is Weibull, and then through the plotting, polynomial regression of 1st and 2nd order and ANOVA, has the objectivity of statistical procedure, and after that, find the variation point by partial F-test. In each region seperated by the found variation point,better fitted curve is searched repeatedly and finally selected according to the characteristic of the each seperated region.

After obtaining the software reliability performance from the previous best fitted curve, S-curve fitting on based on SRGMs is performed. S-curve fitting method regards the realization of the random data event as the order statistics ,and then cumulative hazard rate data arranged by the number of error can be regarded as the time series data. Software reliability is obtained directly from the exponent of estimated equation.

The developed program for the application procedure of PWF model and S-curve fitting method will be a easy-to-use tool if model assumptions are handled carefully.

In numerical examples, the application results of each model through the two data group are showed and discussed.

Coclusively, the application of the developed PWF model and S-curve fitting method makes the quality of software reliability prediction improved. Improving stems from the saving of the time and money to seek the appropriate software reliability model.

LIST OF TABLES / FIGURES

I. INTRODUCTION

1. Awareness of The Problem

Since the 1970's,people are beginning to realize that some of the largest costs in the development of a computer system,or in the modification of an existing system,are those associated with development of the system software.

In the American goverment fisical year 1980, approximately $51billion was spent a computer systems and $32 billion(56% of the total) was spent on computer software.How aweful costs are (if we note that annual sales of 9 million automobiles at an average cost $8000 each represents $72 billion)!.

Moreover trend of the estimated software growth can be obtainable from computer manufacturers shows the unbelivable amount in Table 1. *1

year	machine instruction	by Exponential fitting
1954	5000	5414
1956	20000	12919
1959	35000	47628
1961	100000	113657
1964	350000	418983
1966	1000000	999838
1967	2000000	1544532
1970	-	5693740
1980	-	440622966
1985	-	3268228800

Table 1. Exponetial curve fit to McClure's data for software growth

In the Table 2 also,about 40% of the effort on programming projects is devoted to testing to detect errors and correcting the software to eliminate those which are found.*2

When we overview the remarkable growth of software size and software effort distribution,problem area can be focused in the maintenance and testing cost. These high cost of software is largely due to reliability problem. Therefore software reliability and error contents measures should be viewed significantly as quantitative measures to sell whomever when enough testing has been done and product is ready for release in the trade-off of cost-effective.

---

*1 & *2 : Martin L. Shooman," Software Engineering ",Mcgraw-Hill, 1983,pp 10-14.

	analysis & design	coding & auditing	test and integration
Command-control (SAGE ,NTDS)	35	17	48
Command-control (TRW)	46	20	34
Spaceborne (Gemini,Saturn)	34	20	46
GP executive (OS/360)	33	17	50
Scientific (TRW)	44	26	30
Business (Raytheon)	44	28	28

Table 2. Software effort distribution by activity (%)

Modern programming techniques(structure programming : top down design) will produce significantly fewer errors. However,there are still some errors. Actually, critical software errors have beeen experienced in the most highly technical area. These set of classic errors might have been resulted in disaster or near-disasters. A software error in the onboard computer of appollo 8 spacecraft erased part of the computer's memory. Eighteen errors were detected during the 10 day flight of Appollo 14. The effort attracted some of the nation's best computer programmers and involved two competing teams. The Air Command's 465L command system, even after being operational for 12 years, still averaged one software failure per day. An error in a single FORTRAN statement resulted in the loss of the first AMERICAN probe to VENUS. Worst of all, errors in medical software have caused death and an error in an aircraft design program contributed to several serious air crash,although information on these error is, as one might expect,sketchy. *1

Awareness of the above problem area leads to the necessity which ways to develop the more reliable software should be suggested and methods to assure the more accurate or adequate software reliability should be developed.In fact,unless we are experienced with a low-error-content design technique, or unless we can measure the error content to judge the quality of the software,we may not be willing to trust the method and reduce the amount of program testing.

Up to now, it is true that a number of models and techniques concerned with software reliability have been proliferated and many of them have been used useful measures. However, even though various measuring techniques and models have

*1 : Glenford J.Myers," Software Reliability ", John Wiley & Son, 1976,p 25.

been developed, approaching as the purpose of general use of it is not easy because of the different assumption, limitted condition,difficulty of data acquisition.

2. Prelude and unfolding

Considerable research has been carried out to study software failure phenomenon and to develop and apply software reliability models to predict software performance. Various models have been proposed for characterizing software reliability in a numerical sence and describing its depedence on various related to the software product and the software development process. Most of the model designers have tried to validate their theory about the software reliability estimation, measurement, prediction, using the various data.

However,software engineers and manager have been left adrift with very little guidance as to which models may be best or may be best for their application. The resulting lack of credibility of the model due to the small number of experiments and the lack of consensus on what is the model utility,applicability,and validity dosen't facilitate their use. This dificiency is a barrier for the quality assurance and certification of computer sysytem. Generally there is no systemmatic approach by which an analyst could choose the best model for his use.

Intention of author devoted to improve the quality of software reliability prediction as adopting the concept of Piecewise Weibull failure rate which is the changed form of the existing Weibull failure rate model rather than making of relatively new model and then competing with other models, and also devoted to seek the improved methods of software reliability prediction as analyzing the applicability of Piecewise Weibull failure rate model by comparing with the S-shaped reliability growth model.

II. BACK GROUND INFORMATION

1. Meaning and Measurement of Software Reliability

A number of views to what software reliability is and how it should be quantified has been discussed.

Software reliability is a metric which is the probability of operational success of the software. Since this metric can be predicted,measured during program development, and demonstrated upon program completion, reliability analysis and testing serves as one of the most important means of measuring the quality of software and managing its development.

In practical,program proving and program testing are two approaches to judge whether program is reliable or not. However,due to the imperfectness of these approaches in assuring a correct program, a metric is needed which reflects the degree of program correctness and which can be used in planning and controlling additional resources needed for enhancing software quality. One such quantifiable metric of quality that is commonly used in software engineering practice is software reliability.

The common definition of software reliability is summerized as probability that a software performs successfully (software faults do not cause a failure) by the given specification for a specified exposure period of time without encountering an error.

The probabilistic nature of this measure is due to the uncertainty in the usage of the various software function. This means software reliability is a function of the impact that errors have on the system users ; it is not necessarily a function of the actual magnitude of the error within the software system.*1 It is not an inherent property of a program ; it is largely related to the manner in which the program is used. An assessed value of the software reliability measure is always relative to a given use environment. Two users exercising two different sets of paths in the same software are likely to have different values of software reliability.

The specific exposure period of time here may means a single run, a number of runs, or time expressed in operating or calendar or excution time units. We must carefully define time since there are many time variables during software development.

The choice of time as the random variable assumes that failures occur due to random traversing of paths in the program which contain bugs for some values of the input parameters.

---

*1 : Amrit L. Goel, " Software Reliability Models: Assumptions,limitations,and Applicability ",IEEE Tran.on Soft. Eng. vol.se-11.No.12,Dec.1985,pp.1412.

These bugs are residual because they have been undetected during development ,because the path has been tested for other parameter value and the program has worked well. The program size has not allowed exhaustive testing, and so these bugs have remained hidden. This means that as operating time increases,the probability of encountering at least one bug increases. If failures occured only when the data arrived and processing began and failed, then a different choice of random variable would be in order.

A careful definition of software errors will be needed for the measurement and demonstration phase of reliability. We can define software failures in the abstract. However, raw data are in terms of system failures practically. When a system failure occurs, all available records are recorded and analyzed and divided into hardware, software, operator, and unresolved errors.

A software " error "is presented when the software does not do what the software user reasonably expects it to do. The presence of an error is a function of both the software and the expectations of its users.

A software " failure " is an occurence of software error. It is said to occur when an error results because the program did not compute or perform a function correctly.

A software error occurs when a system failure is experienced which is traceable to an underlying " software fault ". In colloquial speech, either errors or faults are called " bugs ".

We may think of faults as causes and errors as effects. If a single fault results in an associated single system failure,we call it a single error. If system failure exists and we are sure it is a software problem,then a software error exist regardless of whether or not we can find the corresponding faults. *1

The detection of errors can be effected by monitoring the system(or simulated system) performance or by reading the code and finding a fault which will cause an error.

Current approach for measuring software reliability basically parallel those used for hardware reliability assesment with appropriate modifications to account for the inherent differences between software and hardware. A commonly used approach is via an analytical model whose parameters are generally estimated from available data on software failures. Reliability model and other relevant measure are then computed from the fitted model.

---

* 1 : Martin L. Shooman, " Software Engineering ", McGraw-Hill ,1983 ,pp.304-314.

2. Review of Software Reliability Models

The models are shown in literature concerning with software reliability are categorized mainly as software reliability models, software release time models, hardware/software reliability models. The interest in thesis is focused on one of the software reliability models.

The purpose of any software reliability model is to support practical estimation of reliability of large-scale software to assist management in deciding when enough testing has taken place.

Software reliability models can be categorized according to various classifying scheme.

Goel [2] classified the models as four categories according to the nature of the failure process; time between failure models, failure count models, faults seeding models, input domain based models.

Time between failure models assume that time between failures follows a distribution by faults remaining in the program.Model parameters are estimated from the observed value of times between failure,and software reliability and mean time to next failure are estimated by fitting the model. Another approach is which regard failure times as the realization of stochastic process and which describe the failure process as time series. The classified models are Jelinski and Moranda(JM), De-Eutrophication model, Shick and Wolverton (SW) model,Goel and Okumoto Imperfect Debugging model, Littlewood-Verall Bayesian model.

Failure count models assume that failure cannot follows a known stochastic process with a time dependent discrete or continuous failure rate. Model parameters are estimated from the observed values of failure counts. They include Musa excution time model, Shooman exponential model, Goel-Okumoto nonhomogeneous poisson process model, Generalized poisson model, Musa-Okumoto Logarithmic excution time model.

Fault seeding model approach is that the number of original indigenous faults are estimated by using Hypergeometric from known number of fault seeded in program. After testing the program, fault contents of the program prior to seeding is estimated. The including model is Mill's Hypergeometric model.

Input domain model is that test case are guaranted in input distribution with input domain which is associated with program path. An estimate of program reliability is obtained from the failures observed during physical or symbolic excution of the test cases sampled from the input domain. They include Nelson model, Ramamoorthy and Bastani model.

J. G. Shanthikumar [36] classified the software reliability as analytical model and empirical model. The difference between their nature is that the former uses some data gathered from software failure and the latter uses some software metric such as a program complexity measure to predict software reliability. Above analytical models have the two types,i.e, dynamic nature which software failures behave dependently and static nature which dosen't show the time dependent behavior of software failures.

3. Mathmatics of failure density and reliability

A. Failure density and hazard rate

Many failure data are a sequence of time to failure, but the failure density function and hazard rate are continuous variables. It can be shown these discrete functions approach the continuous functions in the limit as the number of data becomes large and the interval between failure time approaches zero by piecewise continuous failure density and hazard rate function.

When assume that there is a set of N items placed in operation at time t = 0 , if items fail according to the progress of time and the number of survivor at any time t_i is expressed as function of time, the number of survivor is n(t). Empirical density function defined over the time interval $t_i < t < t_i + t_i$ is given by the ratio of the number of failures occuring in the interval to the size of the original population, and divided by the length of the interval.

$$fd(t) = [\{ n(t_i) - n(t_i + \Delta t_i)\}/ N] /\Delta t_i$$

$$t_i < t < \Delta t_i$$

Similarly, the hazard rate is defined as the ratio of the number of failures occuring in the time interval to the number of survivors at the beginning of the time interval,devided by the length of the time interval.

$$Zd(t) = [\{ n(t_i) - n(t_i + \Delta t_i)\}/n(t_i)] /\Delta t_i$$

$$t_i < t < t_i + \Delta t_i$$

The failure density function fd(t) is a measure of the overall speed at which failure are occuring, whereas the hazard rate Zd(t) is a measure of the instantaneous speed of failure.

A data failure distribution function Fd(t) and data success distribution function Rd(t) can be defined by

$$Fd(t) = \int_0^t fd(x)\,dx$$

$$Rd(t) = 1 - Fd(t) = 1 - \int_0^t fd(x)\,dx$$

Since the fd(t) curve is a piecewise continuous function consisting of a sum of step functions, its integral is a piecewise continuous function made of a sum of ramp functions.*1

B. Reliability and hazard rate

The random variable t_i is defined as the failure time of the item. The probability of failure as a function of time is given as

$$P(t < t_i) = F(t_i)$$

which is simply the definition of the failure distribution function. The items fail indepedently with probability of failure given by F(t) = 1 - R(t) . The reliability function is a probability of success in terms of F(t), as

$$R(t_i) = P(t_i) = 1 - F(t_i) = P(t \geq t_i)$$

If the random variable N(t) represent the number of units surviving at time t ,then N(t) has a binomial distribution with P = R(t) .

$$P[N(t) = n] = B[n:N,R(t)]$$
$$= [N!/\{n!(N-n)][R(t)]^n[1-R(t)]^{[N-n]}$$
$$n = 0, 1, 2, \text{--------}, N$$

The number of units n(t) operating at any time t is a random variable and the expected value of random variable with binomial; distribution is given by

$$n(t) = E\,[N(t)] = NR(t)$$

Therefore

$$R(t) = n(t) / N \text{ ---------------------------(1)}$$
$$F(t) = 1 - n(t) / N = [N - n(t)] / N$$

From the above function

$$f(t) = dF(t)/dt = -(1/N)(dn(t)/dt) \text{ -----(2)}$$

$$f(t) = \lim_{t \to 0}[\{ n(t) - n(t+ t) \}/(N \ t)] \text{ ----(3)}$$

These relationship explain that the failure density function f(t)

---

*1 : Martin L.Shooman, Ibid. p.563

is nomalized in terms of the size of original population N.

Similarly, hazard rate is defined as

$$Z(t) = -\lim_{\Delta t \to 0}[\{ n(t) - n(t+\Delta t)\} /\{n(t) \Delta t\}]$$

From (3)

$$Z(t) = N f(t) \{ 1/n(t) \}$$

From (1) $Z(t) = f(t) / R(t)$ ----------------------(4)

From above induction, we can obtain the reliability function.

$$R(t) = 1 - F(t)$$
$$= 1 - \int_0^t f(x)dx$$

Substituting into (4) and (1)

$$Z(t) = - \{ 1/N \}\{ dn(t)/dt \}\{ N/n(t) \}$$
$$= - \{ d/dt \} \ln n(t)$$
$$\ln n(t) = - \int_0^t Z(x)dx + c$$

Taking the antilog of both sides of the equation gives

$$n(t) = \exp[c] \exp[- \int_0^t Z(x)dx]$$

When t = 0, initial condition $n(0) = \exp(c) = N$ gives

$$n(t) = N \exp[- \int_0^t Z(x)dx]$$
$$\exp[- \int_0^t Z(x)dx] = n(t)/N$$

Substituting of (1) completes the derivation

$$R(t) = \exp[- \int_0^t Z(x)dx]$$

4. Weibull distribution

The Weibull distribution is well known as one of the most flexible distributions. It is useful in a great variety of applications and empirically fits many kinds of data.

The Weibull probability density function with two parameters is

$$f(t) = (\beta/\alpha)\, t^{\beta-1} \exp[-(t/\alpha)^{\beta}] \quad \text{for } t \geq 0$$

β is called scale parameter
α is called shape parameter

The Weibull cumulative distribution function is

$$F(t) = 1 - \exp[-(t/\alpha)^{\beta}], \qquad t > 0$$

The distribution parameters are sometimes expressed differently

$$F(t) = 1 - \exp[-\lambda t^{\beta}], \qquad = 1/(\alpha^{\beta})$$

or

$$F(t) = 1 - \exp[-t^{\beta}/\theta], \qquad = 1/\lambda = \alpha^{\beta}$$

As the above substitution,

$$F(t) = 1 - \exp(\lambda t^{\beta})$$

The corresponding reliability function is

$$R(t) = \exp[-(t/\alpha)^{\beta}], \quad t > 0$$

or

$$R(t) = \exp[-(-\lambda t^{\beta})]$$

The Weibull hazard rate function is

$$Z(t) = (\beta/\alpha)(t/\alpha)^{\beta-1}, \quad t > 0$$

or

$$Z(t) = \lambda \beta t^{\beta-1}$$

The cumulative hazard function is

$$H(t) = \int_0^t (\beta/\alpha)(t/\alpha)^{\beta-1}\, dt = (t/\alpha)^{\beta}, \quad t > 0$$

or

$$H(t) = \lambda t^{\beta}$$

This form is a power function of time.
Then taking the antilog of time t as a function of H is

$$\log(t) = (1/\beta)\log(H) + \log(\alpha)$$

The Weibull mean is expressed by Gammma function

$$E(t) = \alpha \Gamma[1 + (1/\beta)]$$

The Weibull variance is

$$Var(t) = \alpha^2 \{\Gamma[H(2/\beta)] - \{\Gamma[H(1/\beta)]^2\}\}$$

When $\beta = 1$,Weibull distribution is the simple exponential distribution and we get a constant hazard rate reliability function.

When $\beta < 1$, we get decreasing hazard rate reliability function.

When $\beta > 1$, we get an increasing hazard rate reliability function.

When $\beta = 2$, Weibull disrtibution is the Rayleigh distribution.

When $3 < \beta \leq 4$, The shape of the Weibull distribution is close to that of the nomal distribution.

When $\beta \geq 10$, The shaped of the Weibull distribution is close to that of extreme value distribution.

III. PROPOSITION OF THE PIECEWISE WEIBULL MODEL

1. Weibull Models in Software Reliability

M. lloyd and M. Lipow suggested Weibull distribution which is different with general Weibull distribution. *1 Their probability density function of the distribution is given by

$$f(t) = b\beta t^{\beta-1} \exp(-\beta t)^{b}, \quad t > 0, \beta > 0, b > 0$$

where t is time
b is the shape parameter
β is the scale parameter

Estimations of model parameter can be obtained by using an iterative process through the maximum likelihood estimation technique.

John D. Musa and Kazuhira Okumoto suggested generic function as a theoretical failure intensity function (t) in searching the data for possible trends. This assumed function also represents the form of Weibull class function. *2

$$\lambda(t) = \alpha t^{r-1} \exp(^{r}-\beta t)$$

Resently Abdalla A. Abdul-Ghaly assumed the Weibull dirtibution as similar form to the model of M. Lloyd and M. Lipow in his Ph.D dissertation. *3

$$f(x) = \alpha x^{\beta-1} - \alpha \frac{x}{d}^{\beta}, \quad x > 0$$

---

*1 : M. Lloyd and M. Lipow," Reliability; Management,Method, and Mathmatics ",Prentice-Hall Englewood Cliffs, NJ ,1961.

General Weibull ditribution is ditribution of Waloddi Weibull.

*2: John D.Musa and Kazuhira Okumoto ," A Comparison of Time Domains for Software Reliability Models ", The Journal of Systems and Software ,1984,pp.277-287.

*3 : Abdalla A.Abdel-Ghaly, P.Y. Chan, and Bev Littlewood," Evalution of Competing Software Reliability Prediction", IEEE Tran.on soft. eng. vol.se-12, No.9, Sep. 1986. 950-967

Wagonor's model (*4) which the time to failure caused by each error is represented by general Weibull distribution. Type of model is continuous time-independent and identical probabilistic error behavior. The major assumption is that hazard rate function $\lambda(t)$ of the time to software failure caused by an error has

$$\lambda(t) = (\beta/\alpha)(t/\alpha)^{\beta-1}, \quad t > 0$$

where β is the shape parameter
α is the slope parameter

The estimation procedure of model parameter is performed by least square method and each estimates are obtained by estimation of a and b in

$$Y(i) = a + bx(i)$$

where $Y(i) = \ln\{ \ln[1/(1 - F(i))]\}$, $i = 1,2,----,m$

$X(i) = \ln\{ t_i \}$, $i = 1,2,------,m$

$F(i) = ni / n$
(nomalized cumulative errors in the i-th time interval with repect to the total number of errors)

t_i = cumulative time up to and including the i-th debugging interval

n_i = cumulative number of errors detected and removed up to time t

n = total number of errors detected during the total of m debugging interval

Then the estimates for β and α are

$$\beta = [\sum_{i=1}^{m} (Y(i) - \bar{Y})(X(i) - \bar{X})] / \sum_{i=1}^{n} (X(i) - \bar{X}^2)$$

$$\alpha = \exp\{ - (\bar{Y} - n\bar{X}) / n \}$$

where $\bar{X}$ = Geometric mean of X(i)

$\bar{Y}$ = Geometric mean of Y(i)

As the performance measure, mean time to failure is

$$MTTF = \beta\alpha\Gamma(1/\beta)$$

where $\Gamma(\cdot)$ is the Gamma function
and reliability of the software is

$$R(t) = \exp\{ - (t/\alpha)^{\beta} \}, \quad t \geq 0$$

---

*4 : J. G. Shanthikumar classified the Wagoner's model in his paper, " Software reliability models: A Review ", Micro. Relia. vol.23, No.5,1983, pp.914-916.

2. Estimation of Empirical cumulative hazard rate

The value of the population cumulative distribution function at a given time is the population fraction failing by that time. Similarly , the value of the sample cumulative distribution function at a time is the sample fraction failing by that time. If a sample has i of n observations during the particular time,then the sample cumulative distribution function at that time is i/n.

Similarly,for a sample plotted on hazard paper, the increase in the sample cumulative hazard function at a failure time is equal to its conditional failure probability 1/K ,where K is its reverse rank.Then the sample cumulative hazard function, based on the sum of the conditional probabilities of failure,approximates the theoretical cumulative hazard functions, which is the integral of the conditional probability of failure.*1

When we calculate the each failure times corresponding hazard value, a single failure time hazard value is given by 1/n ,where n denotes the number of items or units whose running or failure times are greater than or equal to that failure.

If suppose that n(t) is the number of unfailure(remaining error)that do not fail or are not detected prior to instant and t_i, t_{i+1} is (i)th , (i+1)th failure time, empirical failure rate can be obtained as below for sufficiently small and large n.

$$Z(t_i) = \frac{n(t_i) - n(t_{i+1})}{n(t_i)\,\Delta t_i}$$

here, if $\Delta t_i = t_{i+1} - t_i$,

$$Z(t_i) \simeq \frac{\Delta n}{t_i\, n(t_i)}$$

when n is the number of failures during the interval (t_i, t_{i+1}). TheΔn is 1 failure between i-th failure and i+1 th failure.Then $Z(t_i)$ is rewritten by

$$Z(t_i) = \frac{1}{n(t_i)\,\Delta t_i}$$

We remember that cumualtive hazard function $H(t_i)$ is

$$H(t_i) = \int_0^{t_i} Z(t_i)\,dt$$

If Δt_i is equal to $(t_i - 0)/n$, we can obtain the

*1 : Wayne Nelson," Applied Life Data Analysis ",John Wiley & Sons, 1982,p.155 .

cumulative hazard value from the fundamental definition *1.

$$\int_{0}^{t_i} Z(t_i)dt \simeq \sum_{i=1}^{n} Z(0 + a \quad t_i) \quad t_i$$
$$= Z(n \quad t_i) \quad t_i$$
$$= Z(t_i) \quad t_i$$
$$= 1 / n(t_i)$$

Thus cumulative hazard value can be estimated as the reverse number of the unfailure to that time.

---

*1 : Martin L. Shooman," Probabilistic Reliability : an engineering approach ",McGraw Hill , 1968,p.495.

3. Formulation of Piecewise Weibull Model

As defined in chapter 2 , the discrete function approaches the continuous functions in the limit as the number of data becomes large and interval between failure time approaches zero. We remember that fd(t) function is a piecewise continuous and also its integral i.e,cumulative distribution function is a piecewise continuous function.

Generally,determination of model parameters by estimation theory from failure analysis data results in computations which are made directly from the data itself rather than from fd(t) or Zd(t) . Study of these piecewise continuous function is followed by the choice a continuous model which fits the data satisfactorily.

Graphical estimation method about these study can be greatly useful to determine a distribution which fits a set of failure data and to derive interval estimates of the distribution parameters. Probability plotting techniques have been developed for such purpose.A significant stimulus to the use of the Weibull distribution in reliability engineering was the publication of papers by the Kao[11],Nelson[37] , where extremely simple graphical procedures were presented whereby the distribution parameters could be estimated.Along with graphical procedures ,formal analytical procedures have been developed by statisticians.*1 These are based upon the cumulative distribution function of the distribution concerned.

However, instead of plotting the cumulative proportion of failure,we can plot the cumulative hazard function by using hazard paper.This technique has particular advantage when dealing with censored data. *2 One of the advantage is able to sketchy quickly and roughly with less labor prior to fit adequate theoretical distribution.

Thus hazard plotting technique should be used in preference to cumulative probability plotting when dealing with censored data, or when the data include multiple failure modes and we wish to analyze the overall failure distribution, as well as individual failure models.

Careful consideration also should be used in interpreting data that do not plot as a straight line since the cause of the non-linearity may be due to the existence of mixed distributions,or because the data do not fit the Weibull

---

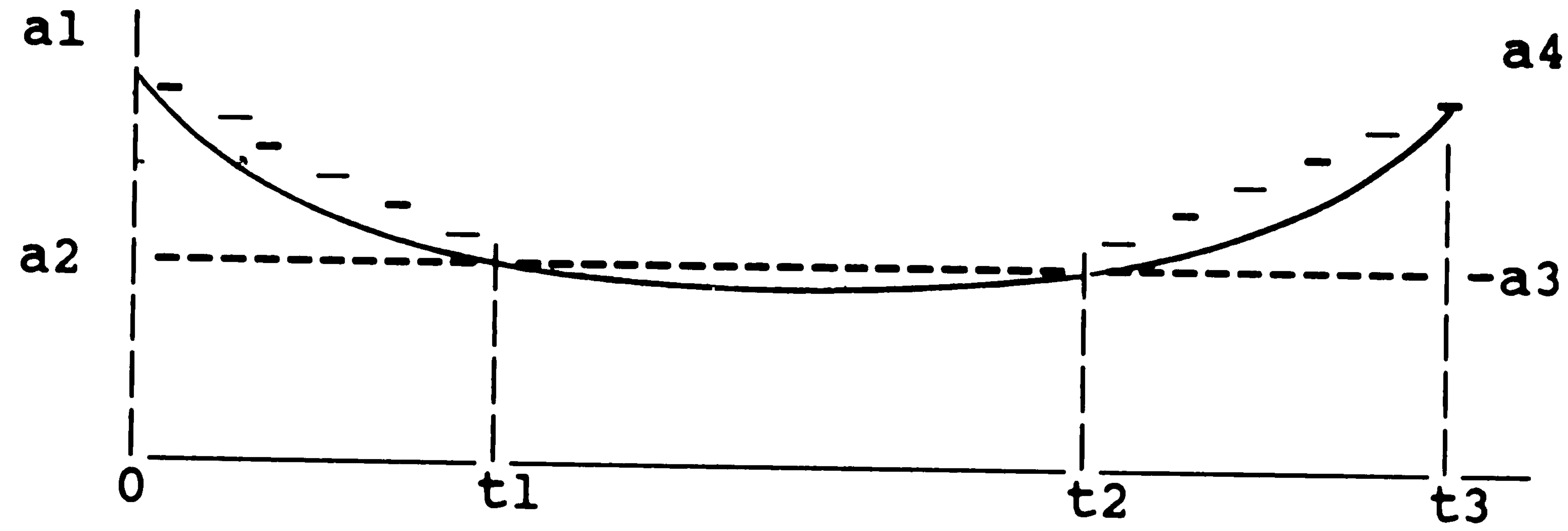
*1 : Karen Fung and A.K.S. Jardine," Weibull Parameter Estimation ", Microelectron. Reliab., vol.22, No.4,pp.681-684, 1982

*2: Patrick D.T. and O'connor, " Practical Reliability Engineering ",John Wiley & Sons,Mar.1984,PP 75 - 77.

distribution. For such situation,various technique have been tried to better fit in shifted models, piecewise linear model(*1) , power series, wide range of the general failure curves.

Piecewise linear approach is to subdivide the curve into a number of regions and fit each region with a simple model Fig. 1 . The truncated nature can be treated and time-shifted function may be thought of as a shifted Weibulll function.

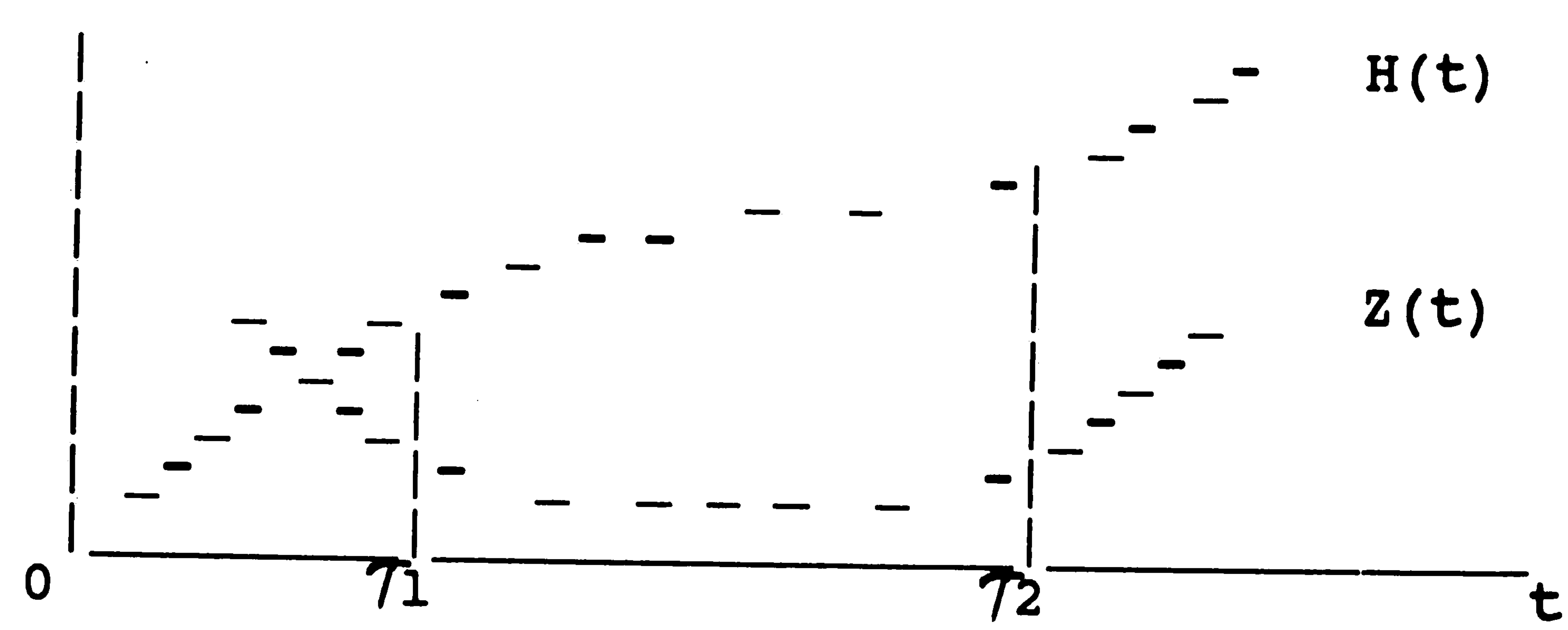
Fig.1 Subdivision of curve



When the distribution of given failure data is unknown theoretically and empirically,theoretical disrtibution of failure data can be obtain by the estimation of empirical cumulative hazard rate, the assumption of distribution by plotting of data, the finding of variation points,the fitting of distribution in each region between variation points.

Therfore,if failure data or cumulative failure data is given,we can sketchy the cumulative hazard curve corresponding to failure rate curve as below.

Fig.2 Inflection of cumulative hazard



If we assume that failure rate data has Weibull distribution, cumulative hazard curve can be devided according to the each regions which has the pattern of different failure. In Fig. 2 ,if 1, 2, 3 are nomalized time, cumulative hazard function can be expressed differently in each regions of (0 - τ_1),(τ_1- τ_2), (τ_2-∞).

---

*1 : Martin L. Shooman , Ibid. Ch.4 ,1968

We remember that cumulative hazard function H(t) is given from Ch.II.

$$H(t) = \lambda t^{\beta}$$

α = shape parameter
β = scale parameter
t = time

Then
$$Z(t) = dH / dt$$
$$F(t) = 1 - Exp[H(t)]$$
$$R(t) = Exp[-H(t)]$$

If the results by plotting of unknown failure data is judged as the simple power function which is affected by shaped parameter β_i , then

$$\begin{aligned} H(t) &= \lambda_1 t^{\beta_1} && ,0 < t \leq \tau_1 \\ &= \lambda_1 t^{\beta_1} + \lambda_2 (t - \tau_1)^{\beta_2} && ,\tau_1 < t \leq \tau_2 \\ &= \lambda_1 t^{\beta_1} + \lambda_2 (t - \tau_1)^{\beta_2} + \lambda_3 (t - \tau_2)^{\beta_3} && ,\tau_2 < t \end{aligned}$$

$$\lambda_1 , \lambda_2, \lambda_3, \beta_1, \beta_2, \beta_3 > 0$$

Hazard rate function Z(t) is obtained by differential of H(t) against the time t.

$$\begin{aligned} Z(t) &= \lambda_1 \beta_1 t^{(\beta_1 - 1)} && ,0 < t \leq \tau_1 \\ &= \lambda_2 \beta_2 (t - \tau_1)^{(\beta_2 - 1)} && ,\tau_1 < t \leq \tau_2 \\ &= \lambda_3 \beta_3 (t - \tau_2)^{(\beta_3 - 1)} && ,\tau_2 < t \end{aligned}$$

This form of hazard rate function is same as Weibull distribution which is explained in Ch.2. The accuracy of the piecewise Weibull approximation can be improved by taking more segments.

Here,we can define Z(t) the as piecewise Weibull hazard rate function.

From the relationship in Section 3 of Ch.II, R(t),F(t) is obtained easily.

Reliability function R(t) is given by

$$\begin{aligned} R(t) &= \exp[- (\lambda t^{\beta_1})] && , 0 < t \leq \tau_1 \\ &= \exp(-K1) \exp[- \lambda_2 (t - \tau_1)^{\beta_2}] && ,\tau_1 < t \leq \tau_2 \\ &= \exp(-K2) [- \exp\{ - \lambda_3 (t - \tau_2)] && , \tau_2 < t \end{aligned}$$

$$K1 = \lambda_1 \tau_1^{\beta_1}$$

$$K2 = \lambda_1 \tau_1^{\beta_1} + \lambda_2 (\tau_2 - \tau_1)^{\beta_2}$$

Distribution function F(t) is given by

$$F(t) = 1 - \exp[-(\lambda_1 t^{\beta_1})] \quad , 0 < t \leq \tau_1$$

$$= 1 - [\exp(-k1) \exp[-\lambda_2 (t - \tau_1)]] \quad , \tau_1 < t \leq \tau_2$$

$$= 1 - [\exp(-K2) \exp[-\lambda_3 (t - \tau_2)]] \quad , \tau_2 < t$$

4. Estimation of Model Parameters

A. Estimation of Cumulative Hazard Rate Function

We now turn to statistical techniques which can be used to efficiently process data and obtain best values for model parameters.

From the Ch.II, Cumulative hazard rate function is given as nonlinear regression form.

$$H(t) = \lambda t^{\beta}$$

Firstly, if we take the logarithm of both sides of the equation,then logH(t) is expressed as linear function of log t.

$$\log H(t) = \log \lambda + \beta \log t$$

β = slope
$\log \alpha$ = intersect of Y axis

Therefore,when failure time is given as t1,t2,....... tn, cumulative failure rate H_i is estimated by the method in Ch.III-2..

If $X_i = \log t_i$ and $Y_i = \log H_i$,simple linear regression model is set up.

$$Y = a + bx + \epsilon$$

From model, estimates a , b are obtained.

$$a = X - bY$$

$$b = \frac{\sum_{i=1}^{n} X_i Y_i - (\sum_{i=1}^{n} X \sum_{i=1}^{n} Y_i)/n}{\sum_{i=1}^{n} (X_i)^2 - (X_i)^2/n}$$

In the same manner, shape parameter $\hat{\beta}$ and scale parameter $\hat{\lambda}$ are obtained.

$$\hat{\beta} = \frac{\sum_{i=1}^{m}(\log t_i)(\log \hat{H}_i) - (\sum_{i=1}^{m}\log t_i \log \hat{H}_i)/n}{\sum_{i=1}^{m}(\log t_i)^2 - (\sum_{i=1}^{m}\log t_i)^2/n}$$

$$\hat{\lambda} = \exp\{\sum_{i=1}^{m}\log \hat{H}_i/n - \hat{\beta}(\sum_{i=1}^{m}\log t_i/n)\}$$

Graphical technique which is able to judge the rough distribution by log graph is also possible,but it requires considerable computation.

B. Estimation of Variation Point

When the given data approaches Weibull distribution,if plots the log $\hat{H}_i$ against log t_i,log $\hat{H}_i$ will show the straight line. When the given failure data can be fitted as straight line in the entire range of the observed time, a cumulative Weibull hazard rate function in entire region is obtained.

However, if a point begins to deviate the fitted straight line at the particular time point, this point will be an variation point which may be fitted better in another ditribution.These points will be another starting points which need to be fitted for another straight line.

The focus of this method is that the point losing the tendency of straight line will be a point which begins to fit better the given data in quadratic regression than simple regression.That is,this point is starting point which begins to need the addition of control variable X^2 in below quadratic polynomial regression model.

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \epsilon$$

Whether or not X^2 is needed depends on the test of below hypothesis

$$H_0 : \beta_2 = 0$$
$$H_1 : \beta_2 \neq 0$$

The important is to find the time point which is able toreject the hypothesis.

In summary,the methodof finding the variation point by partial F-test is : performs the first and second regression with excluding the last data point from the whole data,and then obtains the error sum of squares and performs the partial F-test, secondly,repeats the first procedure until the first point not to be rejected at the significant level $\alpha = 0.05$ will be found and then obtains the maximum region to be able to fit as a straight line.

C. Parameter Estimation of PWF Model

From Ch.III - 3.,

$$H(t) - \lambda_1 t^{\beta_1} = \lambda_2 (t - \tau_1)^{\beta_2}$$

$$\log (H(t) - \lambda_1 t^{\beta_1}) = \log \lambda_2 + \beta_2 \log(t- \tau_1)$$

here, $\lambda_1 t$ and τ_1 is the known value from the estimation of variation point. Therefore cumulative hazard rate function can be rewritten as the integral of hazard rate function.

$$\lambda_1 t^{\beta_1} = \int_0^{\tau_1} Z(x)dx$$

$$H(t) - \lambda_1 t^{\beta_1} = \lambda_2 (t - \tau_1) = \int_{\tau_1}^{t} Z(x)dx$$

This relatinship means that cumulative hazard rate after should be calculated newly without considering of data before . Newly calculated hazard rate is given by

$$\hat{H}_t = \hat{H}_t - \hat{H}_{\tau_1}$$

τ_2 also is estimated by estimation method of variation point from III-4.-B. When τ_1, τ_2 is estimated,model parameter λ_i, β_i are obtained automatically.

5. Computerized Estimation Procedure

A step by step procedure for software reliability modeling has the below steps generally.

step 1 : Collect and study software failure data
step 2 : Plot the data
step 3 : Choose a reliability model
step 4 : Obtain estimates of model parameters
step 5 : Obtain the fitted model
step 6 : Perform goodness-of-fit test
step 7 : Obtain estimates of performance measures
step 8 : Decision making

According to the above steps,computerized procedure for modeling is estabished. Graphical plotting for step 1,2 is an extremely useful technique for data screening.It is of assistance in deciding whether or not the observed data are likely to come from a Weibull distribution, and further it has the advantage of assisting in the detection of outliers. It is also a fast and easy way of getting a rough estimate of the parameter values.

However, the major source of errors comes from the subjectivity inherent in fitting the line to the plotted points. Therefore, computerization of estimation procedure will support the objectivity in fitting.

Detail procedure for modeling are described below.

1) Tabulate the time to failure in ascending order of time
2) Count the number of remaining after previous failure (censoring)
3) For each failure, calculate hazard interval

$$H = 1 / (\text{ number of items remaining after failure occuring })$$

4) Calculate the cumulative hazard rate

$$H = \Delta H1 + \Delta H2 + \Delta H3 + \text{-------} + \Delta Hn$$

5) Plot the ranked data on the appropriate hazard paper (Weibull paper)
6) Performs the first regresson analysis against the whole failure data with the below model.

$$Y = \beta_0 + \beta_1 X + \epsilon$$

t_i = each failure time
$\hat{H}_i$ = cumulative failure rate
X = log t
Y = log H

7) After the $\hat{\beta}_0$, $\hat{\beta}_1$ and confidence limit are obtained, and variance and residual are analyzed, performs the second regression and partial F-test with the confidence limit $\alpha = 0.05$. If hypothesis is accepted, stop and obtain the parameter of model
8) If hypothesis is rejected, exclude the last failure data and performs the first and second regression and partial F-test.
9) During the repeat of (8) procedure, if the point is not rejected is founded firstly, obtain the model parameter by the first regression until that point. If remained data is less than two, stop.
10) About the remaining data, repeat the procedure to 9 from 5.
11) Obtain the fitted distribution in each region or entire region
12) performs the sensitivity analysis about the specific cases.

The program to above computerized procedure can be offered .

IV. S-SHAPED SOFTWARE RELIABILITY GROWTH

1. S-shaped Growth Curve

During the debugging/test phase,the software is tested to detect software errors remaining in the system and correct them. Assuming that no errors are introduced,the probability that no failure occurs for a fixed time interval,i.e,the reliability increases with the progress of software testing. Thus phenomenon is called software reliability growth.

A software reliability growth curve representing a relation between the time span of software testing and the cumulative number of detected errors is observed in a software error detection process during the software debugging/testing phase. The curve of the number of detected software errors for the observed historical data is S-shaped.

There are many reason why observed software reliability growth curve often become S-shaped.

The S-shaped software reliability growth curve is typically caused by the definition of errors(i.e,failures or faults): under what conditions test personnel decide that they have detected an error.The growth is also caused by the continuous test efforts increase in which the test effort has been incrementally increased through the test period.

If we assume the mutual independency of fault,all faults in a system(program) are randomly captured(failure occurs randomly).Actually,faults are mutually depedent because of logical or functional dependencies that exist within a program. This mutual dependency of faults makes the observed software reliability curve S-shaped,the number of faults increases as the number of detectable faults increases.During the early phase of a test,the growth is slow.The more faults are removed,the more dependent faults become detectable.Then the growth gradually goes up while the number of undetected faults which are detectable increases.The growth becomes slow again beyond this point,because the number of detectable faults gradually decreases.Thus,the growth of this failure detection process becomes S-shaped[9].

In different explanation,S-shaped growth curve can be regarded as a learning process in which test-team members become familiar with test environment,i.e,test skills gradually improve[20].

2. S-shaped software Reliability Growth Model

A software reliability growth curve is already defined in previous section. A software reliability model describing an error detection phenomenon which the reliability increases with the progress of software testing is called a software reliability growth model(SRGM) [24].

Applying the SRGM's to the observed software error data, the number of errors remaining in the system and software reliability function can be estimated. Then,using the software reliability data analyses based on the SRGM's,software reliability can be evaluated.

Several SRGMs have been developed for analyzing the software error detection process in S-shaped growth curves of detected errors.The delayed S-shaped SRGM,inflection S-shaped SRGM,exponential and modified exponential SRGM have been developed as stochastic SRGMs based on NHPP(nonhomogeneous poisson process) [9, 20, 21, 31]. The logistic and the Gompertz SRGM have been widely used to various project as deterministic SRGMs based on the regression analysis through the curve fitting[21].

In stochastic SRGMs,the software reliability growth are described by the error detection rate per error at an arbitrary testing or debugging time point.

The mean value functions of the each stochastic SRGMs are as below.

The delayed SRGM : increasing error detection rate

$$H(t) = M(t) = a\,[1 - (1 + bt)\exp(-bt)]$$

a = statitically expected cumulative number of errors to be detected eventually,i.e,expected initial error content of a software

b = the failure detection rate (the error isolation rate).

The inflection SRGM : increasing error detection rate

$$H(t) = I(t) = a\,[1 - (\exp(-bt)]/[1 + c\exp(-bt)]$$

b = the failure detection rate

c = the inflection factor

The exponential SRGM : constant error detection rate

$$H(t) = M(t) = a[1 - \exp(-bt)]$$

b = the error detection rate

The modified exponetial SRGM : decreasing error detection rate

$$H(t) = M(t) = a \sum_{i=1}^{2} p_i [1 - exp(-b_i t)]$$

p = 1, $0 < p < 1$ (i = 1,2)
content proportion of Type i errors.
a is the expected initial error content of Type i errors

b = $0 < b2 < b1 < 1$
error detection rate per Type i error(i =1,2)

In deterministic SRGMS, Gompertz model and logistic curve are used represent S-shaped software reliability growth.

The expected cumulative number of errors detected up to testing time t is given as below.

The logistic curve model :

$$nl(t) = K / [1 + m\ exp(-pt)]$$

$$m > 0,\ p > 0,\ K > 0$$

The Gompertz curve model :

$$ng(t) = K\ a^{b^t}$$

$$0 < a < 1,\ 0 < b < 1,\ K > 0$$

K,p,m,a,b = constant parameter to be estimated by regression analysis

K = the expected initial error content of a software system

3. Computerized procedure for curve fitting

It does not seem possible to analyze the particular context in which reliability measurement is to take place so as to decide a priori which model is likely to be trustworthy. However,if a user knows that past predictions emanating from a model have been in close accord with actual behavior for a particular data set then user might have confidence in future predictions for the same data.

However,only good model is not sufficient to produce the good prediction.To get the truthworthy prediction ,the more objective procedure should be supported.

Therefore, design of computerized prediction system which are combined with the specific model and statistical procedure will be a leading way to improve the accuracy of prediction of software reliability, and practically will be a useful tool for judging the applicability of model.

A computerized procedure used in this paper for S-shaped curve fitting has the below steps.

step1 : obtain the general trend and averages
step2 : select the curves
step3 : Estimate parameter
step4 : Test by Chi-square statistics
step5 : choose the best fitted curve

The curves used to be selected include linear, quadratic, exponential, modified exponential, logistic, Gompertz curves.

This program can be offered by author.

V. NUMERICAL EXAMPLE AND ANALYSIS

For illustration of software reliability analysis based on the PWF model and the SRGMs ,application examples are presented.

Data set D1,D2 used in making the model application in this paper comes from the investigated sources [38 , 32]. Data set D1 is the continuous time reliability growth type which are measured as excution time in hundreth of second between successive failures. Data set D2 are originally from the U.S Navy Fleet Computer Programming Center,and consist of the errors in the development of software for the real time,multicomputer complex which forms the core of the Naval Tactical Data System(NTDS).

Typically data set D1,D2 are available from software tests as a sequence t1,t2,.... tj of successive times between failures,or as samples x(t1),x(t2),....,x(tk) of failure counting process x(t). The both of data set are regarded as the completely censored data in this paper.

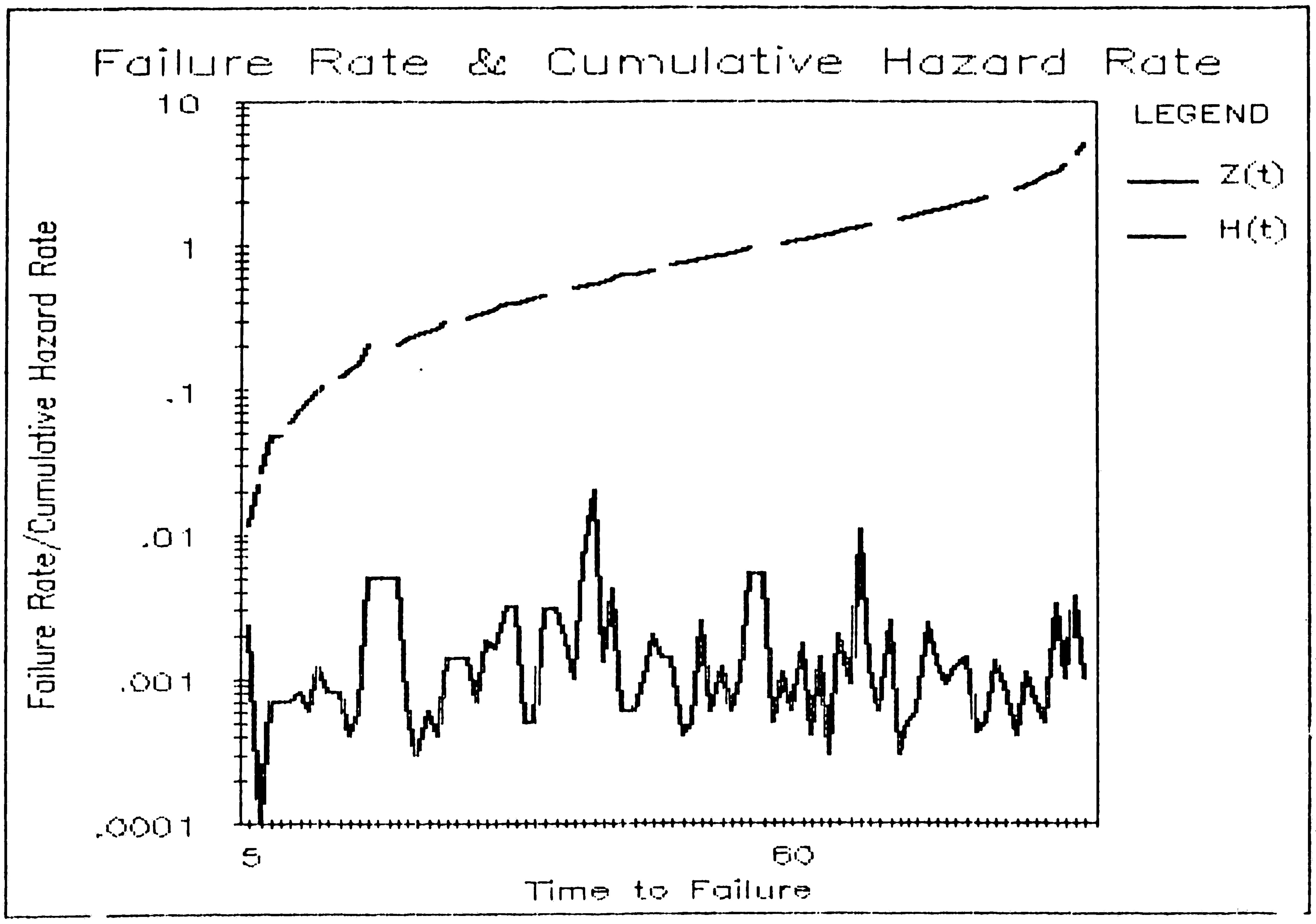
General assumptions underlying the models described in previous chapter are following.

1) The hazard rate to the time to software failure caused by error is an class of Weibull distribution.
2) Time between errors are independent
3) Initial error content is a random variable.
4) Detected error is immediately and completely corrected.
5) No new error are introduced during the fault removal process

1. Application of PWF model

As shown previous chapter, PWF is flexible and can be simple to work. For the data set D1, time between failures/errors,cumulative hazard rate, and failure rate are calculatd and their relationship are plotted in Fig. 3. Data analysis from Fig. 3 indicate that it is not easy to judge the error behavior.

Fig. 3



But plotting of cumulative hazard rate in Fig.3 give us the useful information as the base of judgement in choosing the model.It is useful to analyze the trends in finding the estimates of the unknown distribution. To analyze the detail characteristic of the obtained data,first regression is performed. As the results of regression,regression equation are given by

$$Y = -7.6039695 + 1.0675012 X$$

$Y = \ln H(i)$
$X = \ln t(i)$
$H(i)$ = Cumulative hazard rate
$t(i)$ = time to the detected errors

Standard deviation is 0.0235925 and standard error of estimate is 0.2393280. Coefficient of multiple determination is 0.9763907. These results will be compared with the results of the second regression later.For the quick judgement of the closeness in the given data, the plotting of the estimated Y value is performed.

polynomial regression of order 2 is performed to get the base of the previous guessing which curve is going to yield the better fitting. The results of regression are given by,

$$Y = - 7.694227 + 1.0762097X + 0.000722X^2$$

Standard deviation for regression coeficient β_1, β_2 are 0.0131027 and 0.0010676 and standard error estimate is 0.1862724. Coefficient of multiple determination is 0.9763107.

Comparing the coefficient of multiple determination,the results of 1st order regression gives the better fit than those of 2nd regression becasue coeffcient of multiple determination measures the percent of the total variation about the mean accounted for by the fitted curve. The next step is to investigate the variation point.In order to investigate whether or not a significant trend exist in the estimated cumulative failure rate, F-test are used as explained previously.

Partial F-test at significant level α = 0.05 are performed repeatedly until the Hyphothesis to find the variation point, Ho : $\beta_2 = 0$ and Ha : $\beta_2 \neq 0$ is accepted. The rejection of Hypothesis means that coefficient of 2nd power is needed in the estimated equation.

Through the above process of patial F-test,variation point is found at 14th failure data point because the F value of the 14th failure data,4.62087 begins to less at that point than actual F vaue 4.8443. This means there is significant trend in data point after 14th failure data and the distribution in the back and forth of the 14th failure data is diifferent.

Polynomial regression of 1st order to the observation of 14 failure data is performed and the equation is given by,

$$Y_{14} = - 6.0583256 + .711189 X$$

Stanadard deviation is 0.1071571 and standard eror of estimate is 0.3941247. Coefficient of mutiple determination is 0.7858900. Weibull parameters are obtained as α = 0.0023831 and β = 0.7111899.

By the same method,polynomial regression of the 1st and 2nd order to the observation of 72 failure data are performed and each equation are given by.

1st : $Y_{72} = -5.0890399 + .670886\ X$

2nd : $Y_{72} = -5.3072448 + 0.6911735\ X + 0.0020895\ X^2$

The 2nd order regression equation is prefer to the 1st order regression. So, the next step is to find another variation point at data without 14 failure data. But specific variation point which can affect the fit of the 72 failure data is not found.

Consequently,the estimated equations are expressed in differently in 2 regions. The each results are summerized in Table 3.

< Data Set D1 >

Region	1	2
Ending Time	277.0	5490.
Weibull Parameter	= 0.00233831 = 0.7111899	= 0.00616393 = 0.6708860
Failure Rate Z(t)	0.0016298 * t ** (- 0.2888101)	0.0041353 * (t - 277) ** (- 0.3291140)
Cumulative hazard rate H(t)	0.0023383 * t ** (0.711899)	0.460647 + 0.0061639 * (t -277)** 0.6708860
Reliability	Exp{ - 0.0023383 * t ** (0.711899) }	0.6308765 * Exp { - 0.0061639 * (t-277) ** 0.6708860 }

Table. 3 PWF model performance by data set D1

Similary, PWF model to data set D2 is performed and data set D2 is fitted on the polynomial regression of 1st order through the same process. However, specific variation point is not found in data set D2.That is,there is no seperate region. The results are summerized in Table. 4.

	< Data Set D2 >
Region	1
Ending Time	250.0
Weibull Parameter	= 0.00148537 = 1.2926184
Failure Rate Z(t)	0.00192992 * t ** 0.2926184
Cumulative hazard Rate H(t)	0.00148537 * t ** 1.2926184
Reliability R(t)	Exp { - .0.148537 * t ** 1.2916184 }

Table.4 PWF model performance by data set D2

2. Application of S-curve based on SRGM

For the data set D1,D2 ,S-curve based on SRGMs are applied. S- curves include linear, quadratic, exponential, modified exponential, logistic, Gompertz curves. The using of S-curve has the advantage that the cumulative hazard rate from PWF model is used easily without corection in S-curve fitting. The result of application in S-curve give the chance to judge whether the result from PWF model is valid. The result from PWF might be biased due to unknown factor because it is experimented on only specific distribution . S-curve also compare again linear and quadratic for the above purpose. The model performance from S-curve can be obtained because the source data for S-curve method are consist of the cumulative hazard rate.

The curve fitting of cumulative hazard rate to dataset D1,D2 yields the Table.5 and Table. 6.

< Data Set D1 >

Curve	Function(H(t))	Chi-Square / Degree
Linear	-0.48635 + 0.034083 * t	40.4167 / 27
Quadratic	0.59765 + 0.017041 * t + 0.0001671 * t ** 2	19.3182 / 27
Exponential	0.080147 * 1.0468 ** t	15.8234 /27
Modified Exponential	- 0.18622 + 0.24828 * 1.0303 ** t	9.733 /27
Gompertz	25,256 + 0.0023887 ** (0.98771 ** t)	13.5650 /27
Logistic	0.34308 /(1 + 10.** (.91194 - 0.097599 * t)	381.7619 /27

Table. 5 The fitted function of S-curve

*. t = Number of Error

H(t) = Cumulative hazard rate

< Data Set D2 >

Curve	Function(H(t))	Chi-Square / Degree
Linear	- 0.13287 + 0.05200 *t	6.5333 / 7
Quadratic	0.49211 + 0.026 * t + 0.00034228 * t ** 2	61.3077 / 7
Exponential	0.074171 * 1.1326 ** t	6.6762 / 7
Modified Exponential	- 0.52064 + 0.55679 * 1.0495 ** t	0.5333 / 7
Gompertz	2.9625 * 0.017782 ** (0.9358 ** t)	1.833 / 7
Logistic	1.3694 / {1 + 10.** (1.166 - -0.08875 * t }	6.395 / 7

Table.6 The fitted function of S-curve

The chosen curve for data set D2 is Modified Exponential curve and for data set D1 is also Modified Exponential. The software reliabilty performance to the chosen models are given by,

data set D1 : R(t) = Exp { 0.18622 -0.24828 * 1.0303 ** t }

data set D2 : R(t) = Exp { 0.52064 - 0.55679 * 1.0495 ** t}

3. Comparison of Performance

Comparison of software relaibility performance in each model gives the information for selecting the appropriate software reliability model against the specific data. Intention of model comparison in this paper is not to show which model is superior to another but to suggest the possibility of practical use .

To analyze the model performance , data set D2 is preferred to D1 because the software reliability performance to D1 is not suggested from original source. The performance of data set d2 are known.

Software reliability performance by using the data set d2 are below.

PWF : $R(t) = \exp(- 0.0014853 * t ** 1.2926184)$

S-curve: $R(t) = \exp(0.52064 - 0.55679 * 1.0495 ** t)$

NHPP : $R(t)= \exp(-33.99(e ** -0.00579(250) - e ** - 0.00579(250+t)))$

" t " in S-curve means the number of error.

Time	250	540	849
R pwf(t)	0.154379644	0.00637215	0.0001146
R scurve(t)	0.238173235	0.139587879	0.094667503
NHPP	0.23511574	0.00150048	0.00043316

Table.7 Comparison of performance

Table.7 indicates that s-curve shows higher performance than PWF and NHPP.

VI. CONCLUSION / SUGGESTION

The use of Weibull distribution in software reliability has not been so much. Applications of Weibull distribution in software reliability might have been avoided because software has not wareout failure. However, practical application of Weibull distribution and S-curve in predicting of software reliability should be studied with tool easy-to-use.

Application of PWF model prior to application of specific software reliability model(even though appropriate model exist) will be a profitable method for software manager because PWF model can present the various behavior of errors according to the characteristic of the failure rate and save the time and cost to find a appropriate model through the program.

S-curve fitting method based on SRGMs will have the complementary relationship with the PWF model for the reliable pediction of software reliability.When the software growths are observed ,the nature and extent of the growth will be investigated again in S-curve fitting for the good prediction. It will be a way to compare the predictive quality for obtaining of better prediction than those obtained directly from the original prediction system.It will supplement that only good model is not sufficient to produce good predictions and will offer the base to measure the depth of the too optimistic or pessimistic prediction.

The reason that there is no suggestion in comparison of models is my thinking that the applicability of model and the appropriateness of assumption should be made by the only user of model because of the various environment of model application.

Although the validation of model application is not suffcient because of the lackness of obtainable data and difficulty of data acquisition,it will not be a weakness. However,the developed modelin this paper might bein better explainability if it is used in the large project. Actually, some industrial people who is met for the acquisition of data suggested that modle could be better if it is applied in large project like NASA or Goverment projects.

The reasons they say like that is said because not only private companies haven't made the historical software reliability data but also actuallly haven,t released data willingly if it is not the case of contraction ,even though data are existed and no matter whatever intention of nonrelease is.

The actual case of large project should be studied in the near future to avoid the avoidable disaster and unexpectable loss.

Finally , two computer programs which are cosisted of the PWF program and S-curve fitting program would be a usefull tool in studying various fields of hardware/software reliability.

Working Bibliography

1.
C.V.Ramamoorthy and Farokh B.Bastani," Software reliability - status and perspectives ",IEEE Trans. on software relia. eng. vol.se-8,No.4,July 1982, pp 354-371.

2.
Armit L. Goel, " Software Reliability Models : Assumptions Limitations,and Applicability ",IEEE Trans.on Software Relia. eng.,vol.se-11,No.12,Dec 1985, pp 1411 -1423.

3.
Richard E.Balow and Frank Proschan, " Statistical theory of Reliability and Life Testing ",Mcardle Press,Inc, 1981.

4.
K.K.Govil, " Philosophy of a new structure of software reliability modeling ",Microele. Relia.,vol.24,no 3,pp 407 -409, 1984

5.
Z,jelinski and P.Moranda, " Software reliability research, " in statistical computer performance evaluation, W.Freiberger,ED. New York: Academic,1972,pp.465 -484

6.
J.D.Musa, " A theory of software reliability and its applications," IEEE Trans.Software Eng.,vol.se-1,pp.312-327., Sept. 1975

7.
M.L.Shooman, " Probability models for software reliability prediction, "in Statistical Computer Performance Evaluation,W.Freiberger,ED. New York:Academic,1972,pp. 485 -502.

8.
___," An analysis of competing software reliability models," IEEE Trana.Software Eng.,volse-4,pp.104-120.,1978

9.
Mitsuru Ohba, " Software Reliability Analysis Models," IBM J. Res.Develop. Vol.28 NO.4 July 1984.

10.
B.Littlewood," Theories of software reliability:How good are they and How can they be improved ?," IEEE Trans. Software Eng.,vol.se-6,pp 489-500,Sept.1980

11.
Kao,J.K, " A graphical estimation of mixed Weibull parameter in life-testing of electron tubes," Technometrics,vol.1,4,pp.389-407. 1959.

12.
Shooman,M.L, "Probabilistic Reliability: an engineering approach," Macgraw-Hill, New York 1968.

13.
J.T.Duane," Learning curve approach to reliability monitoring," IEEE Trans.on aerospace,pp.563-566,1964

14.
Shigeru Yamada ,Mitsuru Ohba, Shunji Osaki, " S-shaped reliability growth modeling for software error detection ",IEEE Trans. on reliability,vol.r-32,nNo.5,Dec.1983,pp. 475-484

15.
Shigeru Yamada and Shunji Osaki, " Software reliability growth modeling:Models and applications ",IEEE Trans on Software eng.vol.se-11,No.12,Dec 1985,pp.1431-1437.

16. omit

17.
K.K.Govil, " Fitting cost-reliabiltiy data in a standard curve " Microele. Relia.,vol.25,No.5,pp.913-915.,1985

18.
Shigeru Yamada and Shunji Osaki, " Cost-reliability optimal release policies for software systems," IEEE Trans. on Relia.,vol.R-34,No.5,Dec.1985, pp. 422-424.

19.
Harvey S.Koch and Peter Kubat , " Optimal release time of computer software ",IEEE Trans on software eng.,vol.se-9.No.3,May 1983,PP 323-327.

20.
Shigeru Yamada, and Mitsuru Ohba, and Shunji Osaki, " S-Shaped Reliabilty Growth Modeling For Software Error Detection ", IEEE Tran. on Reliability,vol.r-32,no.5, Dec. 1983. pp.475-478.

21.
Shigeru Yamada, and Mitsuru Ohba, and Shunji Osaki, " S-Shaped Software Reliability Growth Models and Their Application ", IEEE Tran. on Relia.,vol.r-33,no.4,Oct. 1984. pp 289-292.

22.
Sheldon M.Loss, " Statistical Estimation Of Software Reliabilty " ,IEEE Tran.on Software Eng.,vol.se-11,no.5,May 1985, pp 479-483.

23.
Robert Troy, " Assessment of Software Reliabilty Models " IEEE Tran. on Software Eng.,vol.se-11,no.9,Sept. 1985. pp 839-849.

24.
Shigeru Yamada , and Shunji Osaki " Software Reliabilty Growth Modeling: Models and Applications ",IEEE Tran. on Software Eng., vol.se-11,no.12,Dec 1985. pp.1431-1437.

25.
Shigeru Yamada, and Shunji Osaki " Cost-Reliability Optimal Release Policies For Software Systems ",IEEE Tran. on Relia.,vol. r-34,no.5,Dec.1985. pp. 422-424.

26.
Y.S.Sherif, and N.A. Kheir " Reliabilty and Failure Analyses of Computer Systems ",Comput. & Elect. Eng. vol.11,no.2/3, pp. 151-157, 1984.

27
Shigeru Yamada, Hiroyuki Narihisa and Hiroshi Ohitera ," Software Reliability Analysis Based on A Nonhomogeneous Error Detection Rate Model ", Microelect.Relia.,vol.24,no.5, pp. 915-920. 1984.

28.
Armit l. Goel, and Jopie Soenjoto, " Models for Hardware-Software System Operatiional-Performance Evaluation ",IEEE Tran. on Relia., vol.r-30,no.3,Aug 1981. pp.232-239.

29.
Sheldon M.Ross, "Software Reliability: The Stopping Rule Problem ", IEEE Tran. on Software Eng.vol. se-11,no. 12,Dec 1985. pp. 1472-1476.

30.
Janet R.Dunham," Experiments in Software Reliability: Life-Critical Applications ",IEEE Tran. on Software Eng,vol.se-12,no.1, Jan. 1986. pp.110-123.

31.
A.L. Goel,K. Okumoto, " Time Defendent Error Detection Rate Model for Software Reliability," IEEE Trans.Relia.,vol r-28,1979 Aug, pp. 206-211.

32.
W.Nelson,Applied Life Data Analysis,John Wiley,new York, 1982.
H. S. Koch, and P. Kubat, " Optimal Release Time Of Computer Software, "IEEE Trans.Software Eng,vol se-9, 1983 May, pp. 500-503.

33.
E. H. Forman and N. D. Singpurwalla, " Optimal Time Intervals for Testing Hypotheses on Computer Software Errors," IEEE Trans.Relia.,vol. r-28, pp.250-253, Aug. 1979.

34
Pei Hsia, " Software Reliability - Theory and Practice ", Comput. & Elect.Eng ,vol.11,no.2/3,pp.145-149. 1984.

35.
Martin L. Shooman, " Software Reliability : A Historical Perspective ",IEEE Trans.Relia.,vol.r-33,no. 1,April 1984. pp.48-55

36.
J.G. Shanthikumar," Software reliability models: A review " Micro. Reliab. vol.23,no.5 ,pp 903-943,1983.

37.
Nelson,W.," Hazard Plotting for incomplete failure rate data ", J. of Qualit. Tech. vol 1,1,pp.27-52,1969

38.
Abdalla A.Abdel- Ghally P.Y.Chan, and Bev Littlewood ," Evaluation of completing software reliability predictions ",IEEE. Software Eng.,vol. se-12,no.9 Sept. 1986.

APPENDIX

(ONLY MAIN PROGRAM)

```
PWF.
USER(*)
      FTN.
LGO.
C     MAIN PROGRAM BY CHONGMAN PARK
C
C
C
      PROGRAM PWF(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
      DIMENSION B(5),ANS(10),F(200),CH1(200),TTF1(200)
      DIMENSION DELT(200),RAMDA(5),SHAPE(5),X(200),Y(200)
      DIMENSION TD(200),MDE(200),TTF(200),CH(200),Z(200),PMAT(600)
      DIMENSION HZ(200)
      DIMENSION CON1(5),CON2(5),INTT(5),OBS(200),FITT(200),AAMU(200)
      DIMENSION GROUP(100),SFAIL(100)
      COMMON L,M
C
C
      DATA F/161.40,18.51,10.13,7.71,6.61,5.99,5.59,5.32,5.12,
     + 4.96,4.84,4.75,4.67,4.60,4.54,4.49,4.45,4.41,4.38,4.35,4.32,
     +4.30,4.28,4.26,4.24,4.22,4.21,4.20,4.18,4.17/
C
C
      L=5
      M=6
      CASE=0.
C
C
      READ(L,411) KSTEP,KPROC,MODET,MRESI,MDATA
  411 FORMAT(5I1)
      WRITE(M,411)KSTEP,KPROC,MODET,MRESI,MDATA
      IF(MDATA.EQ.1) GO TO 432
      READ(L,1) NT,NTE,MS,IS
    1 FORMAT(4I10)
      WRITE(M,1)NT,NTE,MS,IS
      READ(L,2) (TD(I),I=1,NTE)
    2 FORMAT(8F5.0)
C
      DO 92 I=31,40
   92 F(I)=-0.09*(I-30)/10.+4.17
      DO 93 I=41,60
   93 F(I)=-0.08*(I-40)/20.+4.08
      DO 94 I=61,120
   94 F(I)=-0.08*(I-60)/60.+4.00
      DO 311 I=121,200
  311 F(I)=(3.89-3.92)*(I-120)/80.+3.92
C
C
      IF(MS-1) 15,6,6
   15 WRITE(M,501)
  501 FORMAT(///,30X,"HAZARD CALCULATION FOR COMPLETE DATA"//)
      DO 170 I=1,NTE
  170 MDE(I)=MS
      GO TO 21
    6 READ(L,3)(MDE(I),I=1,NTE)
```

```
    3 FORMAT(40I2)
      IF(MS.NE.1)GO TO 17
      WRITE(M,502)
  502 FORMAT(///,30X,"HAZARD CALCULATION FOR INCOMPLETE DATA"//)
      DO 160 I=1,NTE
      IF(MDE(I).EQ.0)GO TO 160
      MDE(I)=MS
  160 CONTINUE
      GO TO 21
   17 WRITE(M,503) MS
  503 FORMAT(///,30X,"HAZARD CALCULATION FOR FAILURE MODE",I2//)
   21 WRITE(M,504)
  504 FORMAT(/,10X,"FAILURE NO.",5X,"TIME",8X,"MODE",3X,"HAZARD VALUE",
     + 3X,"CUM. HAZARD ",2X,"FAILURE RATE"//)
C
C
      IF(IS.EQ.0)GO TO 4
      CALL SORTT(TD,MDE,NTE)
C
C
    4 CUH=0.
      K=0.
      DO 10 I=1,NTE
      REV=NT+1.-I
      HV=1./REV
      HZ(I)=HV
      IF(MS.EQ.MDE(I).OR.MS.EQ.0) GO TO 11
      GO TO 10
   11 K=K+1
      TTF(K)=TD(I)
      CUH=CUH+HV
      CH(K)=CUH
      IF(K.GT.1)GO TO 29
      DELT(K)=TTF(1)
      GO TO 27
   29 DELT(K)=TTF(K)-TTF(K-1)
      IF(DELT(K).NE.0)GO TO 27
      DELT(K)=DELT(K-1)
   27 Z(K)=HV/DELT(K)
   10 CONTINUE
C
C
      KFAIL=K
      K=0
      K2=0
      DO 220 I=1,NTE
      IF(MS.EQ.MDE(I).OR.MS.EQ.0)GO TO 221
      WRITE(M,500) I,TD(I),MDE(I),HZ(I)
  500 FORMAT(10X,4X,I3,3X,F10.1,8X,I3,5X,F10.7)
      GO TO 220
  221 K=K+1
      IF(K.LE.K2)GO TO 233
      IS=0
      KK=K+1
      DO 230 J=KK,KFAIL
      IF(IFIX(TTF(K)) .NE. IFIX(TTF(J)))GO TO 250
```

```
      IS=IS+1
  230 CONTINUE
  250 CONTINUE
      IF(IS.EQ.0)GO TO 233
      K2=K+IS
      ZS=0.
      DO 231  I1=K,K2
  231 ZS=ZS+Z(I1)
      DO 232  I1=K,K2
      CH(I1)=CH(K2)
  232 Z(I1)=ZS
  233 CONTINUE
      WRITE(M,505)I,TTF(K),MDE(I),HZ(I),CH(K),Z(K)
  505 FORMAT(10X,4X,I3,3X,F10.1,8X,I3,5X,F10.7,F14.7,5X,E11.5)
  220 CONTINUE
      IF(MS.EQ.0)  GO TO 441
      WRITE(M,251)
  251 FORMAT(////,20X,"MODE 0 REPRESENTS CENSORING"//)
  441 CONTINUE
      WRITE(M,254)NT,KFAIL
  254 FORMAT(//,20X,"   SAMPLE SIZE =",I7,9X,"   NUMBER OF FAILURES =",
     + I5/)
      GO TO 433
C
C
C
C  GROUPED DATA
  432 CONTINUE
      READ(L,434)NPER
  434 FORMAT(I10)
      KFAIL=NPER
      READ(L,435)(TTF(I),I=1,NPER)
  435 FORMAT(10F8.1)
      READ(L,436)(GROUP(I),I=1,NPER)
  436 FORMAT(10F8.0)
      READ(L,436)(SFAIL(I),I=1,NPER)
      WRITE(M,439)
  439 FORMAT(//,30X,"HAZARD CALCULATION OF GROUPED DATA"//,10X,"NUMBER",
     + 3X,"TIME",4X,"GROUP SIZE",3X,"FAILURES",3X,"FAILURE RATES",3X,
     + "CUM. HAZARD")
      GCUH=0.

      DO 437  I=1,NPER
      Z(I)=SFAIL(I)/GROUP(I)
      GCUH=GCUH+Z(I)
      CH(I)=GCUH
      WRITE(M,438)  I,TTF(I),GROUP(I),SFAIL(I),Z(I),CH(I)
  438 FORMAT(10X,I4,2X,F8.1,2X,F9.1,3X,F9.1,4X,E12.5,3X,E12.5)
  437 CONTINUE
  433 CONTINUE
C
C
C PLOTTING Z(T)  AND H(T)
      KK1=KFAIL
      DO 20 K=1,KFAIL
      KG=KK1+K
```

```
      PMAT(K)=TTF(K)
      PMAT(KG)=Z(K)
   20 CONTINUE
      NCHART=100
      WRITE(M,252)NCHART
  252 FORMAT(1H1,//,20X," CHART",I5,"   FAILURE RATE PLOTTING ",/)
      CALL PLOTT(NCHART,PMAT,KFAIL,2,0,0)
C
C
      KK2=KFAIL
      DO 22 K=1,KFAIL
      KR=KK2+K
      PMAT(K)=TTF(K)
      PMAT(KR)=CH(K)
   22 CONTINUE
      NCHART=NCHART+1
      WRITE(M,253)NCHART
  253 FORMAT(1H1,///,20X,"  CHART",I5,"   CUM. HAZARD PLOTTING",/)
      CALL PLOTT(NCHART,PMAT,KFAIL,2,0,0)
C
C HAZARD PLOTTING FOR PWF MODEL
C
      CON=0.
      CONN=0.
      ZCO=0.
      BTF=0.
      BTF=0.
      KCO=1
      INIT=0
C
C SAVE DATA
      DO 33 I=1,KFAIL
      TTF1(I)=TTF(I)
   33 CH1(I)=CH(I)
      KNUMB=KFAIL
C
C
  210 JQ=KNUMB
      DO 30 J=1,KNUMB
      IF(CH1(J).LE.0)GO TO 999
      Y(J)=ALOG(CH1(J))
      IF(TTF1(J) .NE. 0)GO TO 8888
      X(J)=-0.0
      GO TO 8887
 8888 X(J)=ALOG(TTF1(J))
 8887 JR=JQ+J
      PMAT(J)=X(J)
      PMAT(JR)=Y(J)
   30 CONTINUE
      WRITE(M,255)KCO
  255 FORMAT(1H1,//,15X,"LOG VALUES OF TIME TO FAILURES AND CUM.HAZARD
     +  ",I5,"   INTRVAL ",///,10X,"NUMBER",5X,"TIME TO FAILURE ",5X,"CU
     + M. HAZARD",5X,"LOG TTF",5X,"LOG CUM. HAZARD"//)
      DO 322 I=1,KNUMB
      WRITE(M,256)I,TTF1(I),CH1(I),X(I),Y(I)
  256 FORMAT(10X,I5,10X,F10.1,6X,F10.7,4X,F10.5,6X,F10.5)
  322 CONTINUE
C
C REGRESSION FOR PWF MODEL
```

```
C
      NAME=1000+KNUMB
      WRITE(M,257)NAME
  257 FORMAT(1H1,//,10X,"  REGRESSION FOR TOTAL INTERVAL  CHART NO.",I5
     + //)
      CALL POLREG(PMAT,NAME,KNUMB,1,1,B,ANS)
      BO=ANS(1)
      B1=B(1)
      IF(MODET.EQ.3)GO TO 155
      IF(KNUMB.EQ.3)GO TO 155
      SS1=ANS(4)
      WRITE(M,301)
  301 FORMAT(1H1)
      CALL POLREG(PMAT,NAME,KNUMB,2,0,B,ANS)
      BBO=ANS(1)
      BB1=B(1)
      BB2=B(2)
C
      SS2=ANS(4)
      ESQ=ANS(9)
      IDOF=ANS(8)
C
C
C
C PARTIAL F TEST
C
      CALL PARF(SS1,SS2,ESQ,F(IDOF),IDOF,ISIG,NAME)
      IF(ISIG.EQ.0)GO TO 155
C
C


C VARIATION POINT
      WRITE(M,302)
  302 FORMAT(1H1/5X,"   DETECTION OF PROCESS VARIATION POINT   ")
      KKK=KNUMB-1
   82 CONTINUE
      IF(KKK.LE.3)GO TO 83
      DO 70 J=1,KKK
      JQ=KKK+J
      PMAT(J)=X(J)
      PMAT(JQ)=Y(J)
   70 CONTINUE
      NAME=2000+KKK
C
      CALL POLREG(PMAT,NAME,KKK,1,KPROC,B,ANS)
```

```
      SS1=ANS(4)
C
      CALL POLREG(PMAT,NAME,KKK,2,KPROC,B,ANS)
      SS2=ANS(4)
      ESQ=ANS(9)
      IDOF=ANS(8)
C
      CALL PARF(SS1,SS2,ESQ,F(IDOF),IDOF,ISIG,NAME)
      WRITE(M,303)
  303 FORMAT(2X,128(1H1))
      IF(KSTEP.EQ.0)GO TO 404
C
      IF (ISIG.EQ.1) GO TO 80
C
C
      GO TO 83
  404 CONTINUE
   80 KKK=KKK-1
      IF(KKK.LT.4)GO TO 83
      GO TO 82
   83 CONTINUE
      IF(KSTEP.EQ.0)GO TO 405
      KP=KKK
      NSAM=0
      KP1=KP+1
      DO 180 I=KP1,KNUMB
      IF(TTF1(KP) .GE. TTF1(KP1)) GO TO 181
      NSAM=NSAM+1
  180 CONTINUE
  181 CONTINUE
      KP=KP+NSAM
C
   81 NAME=3000+KP
      DO 73 J=1,KP
      JQ=KP+J
      PMAT(J)=X(J)
      PMAT(JQ)=Y(J)
   73 CONTINUE
C
      WRITE(M,301)
      CALL POLREG(PMAT,NAME,KP,1,1,B,ANS)
C
      BO=ANS(1)
      B1=B(1)
      GO TO 151
  155 KP=KNUMB
  151 CONTINUE
      RAMDA(KCO)=EXP(BO)
      SHAPE(KCO)=B1
      INIT=KP+INIT
      WRITE(M,301)
      WRITE(M,422)MODET
  422 FORMAT(//,5X,"     MODEL(",I2,")        ")
      WRITE(M,158)KCO,INIT,TTF(INIT)
  158 FORMAT(///,5X,I5,"TM PERIOD",//,7X,"ENDING NUMBER",I5,5X,
     + "ENDING TIME   ",F10.1)
```

```
      IF(MODET.NE.2)GO TO 421
      CON=CON+SHAPE(KCO)*RAMDA(KCO)*((TTF(INIT)-BTF)**(SHAPE(KCO)-1.))
  421 CONTINUE
      INTT(KCO)=INIT
      CON1(KCO)=CON
      ADT=TTF(INIT)-BTF
      CK=RAMDA(KCO)*ADT*SHAPE(KCO)+ZCO*ADT
      CONN=CONN+CK
      CON2(KCO)=CONN
      WRITE(M,52)RAMDA(KCO),SHAPE(KCO)
   52 FORMAT(///,5X,"WEIBULL DISTRIBUTION",/,7X,"SCALE PARAMETER"
     + ,E15.6,5X,"SHAPE PARAMETER",F15.7)
      WRITE(M,261)KCO,CON,KCO,CONN
  261 FORMAT(//,10X,"SMALL K(",I2,") =",E15.6,7X,"LARGE K(",I2," )=",
     + F15.9)
      NAMU=0
C
      IF(KP.EQ.KNUMB) GO TO 156
C
C NEXT PERIOD
C
      DO 90 I=1,KNUMB
      IR=INIT+I
      TTF1(I)=TTF(IR)-TTF(INIT)
      IF(MRESI.NE.1)GO TO 431
      CH1(I)=CH(IR)-CONN-CON*TTF1(I)
      GO TO 90
  431 CONTINUE
      CH(I)=CH(IR)-CH(INIT)-CON*TTF1(I)
   90 CONTINUE
      ZCO=CON1(KCO)
      KNUMB=KFAIL-INIT
      BTF=TTF(INIT)
      IF(KNUMB.GE.3)GO TO 173
      NAMU=KNUMB
      GO TO 171
  173 CONTINUE
      KCO=KCO+1
      GO TO 210
  171 CONTINUE
  156 WRITE(M,157)KCO
  157 FORMAT(//,5X,I5,"TH PERIOD   FINAL")
      WRITE(M,172)NAMU
  172 FORMAT(/,10X,"***** REMAINING DATA =",I5)
C
C
C  SUMMARY OF RESULT
C
      WRITE(M,271)
  271 FORMAT(1H1,///,20X,"   SUMMARY OF RESULT   ",/)
      WRITE(M,422)MODET
      WRITE(M,272)KCO
  272 FORMAT(///,5X," TOTAL PERIOD NO. =",I3)
      DO 273 I=1,KCO
      IN=I-1
```

```
      WRITE(M,275) I,TTF(INTT(I))
275 FORMAT(///,5X," PERIOD ",I3," :  ENDING TIME =",F10.1)
      WRITE(M,276)RAMDA(I),SHAPE(I)
276 FORMAT(///,10X,"WEIBULL PARAMETER",//,13X,"SCALE PARAMETER =",
   + E15.6,5X,"SHAPE PARAMETER =",F15.7)
      SPQ=RAMDA(I)*SHAPE(I)
      SPP=SHAPE(I)-1.
      IF(I.GT.1)GO TO 274
      WRITE(M,277)SPQ,SPP,RAMDA(I),SHAPE(I),RAMDA(I),SHAPE(I)
277 FORMAT(///,10X,"FAILURE RATE FUNCTION : Z(T) =",E15.6,"* T **",
   +F10.7//,10X,"CUM. HAZARD FUNCTION : H ",E15.6,"* T ** ",
   + F10.7//,10X,"RELIABILITY FUNCTION : R(T) = EXP(-",E15.6,
   + " * T **",F10.7,")")
      GO TO 278
274 CONTINUE
      TI=TTF(INTT(IN))
      WRITE(M,279)CON1(IN),SPQ,TI,SPP
279 FORMAT(///,10X,"FAILURE RATE FUNCTION : Z(T) =",E15.6,
   +" + ",F10.7,"* (T - ",F10.1,") ** ",F10.7)
      WRITE(M,281)CON2(IN),CON1(IN),TI,RAMDA(I),TI,SHAPE(I)
281 FORMAT(//,10X,"CUM. HAZARD FUNCTION : H(T)=",E15.6,"+",F10.7,"*
   + * (T - ",F10.1,") + ",F10.7," *(T - ",F10.1," ) ** ",F10.7)
      ECC=EXP(-1.*CON2(IN))
      WRITE(M,282)ECC,CON1(IN),TI,RAMDA(I),TI,SHAPE(I)
282 FORMAT(//,10X,"RELIABILITY FUNCTION : R(T)= ",F12.9,"* EXP -",F1
   +0.7," *(T -",F10.1,") -",F10.7,"* (T -",F10.1,") **",F10.7,")")
278 CONTINUE
273 CONTINUE
      WRITE(M,284)
284 FORMAT(1H1,///,5X,"RESULTING LIFE DISTRIBUTION PLOTTING",//,9X,
   + "TIME",16X,"OBSERVED DISTRIBUTION",10X,"FITTED DISTRIBUTION",15X,
   + "DIFFERENCE"//)
      DO 285 K=1,KCO
      LAS=INTT(K)
      IF(K.GT.1) GO TO 287
      DO 286 I=1,LAS
      OBS(I)=1. * (-EXP(-1. * CH(I)))
      FX=-1.*RAMDA(K)* TTF(I)**SHAPE(K)
      FITT(I)=1.-EXP(FX)
      AAMU(I)=OBS(I)-FITT(I)
      WRITE(M,289) TTF(I),OBS(I),FITT(I),AAMU(I)
286 CONTINUE
      GO TO 285
287 CONTINUE
      KM=K-1
      IX=INTT(KM)+1
      IF(K.LT.KCO) GO TO 182
      LAS =INTT(KCO) + NAMU
182 CONTINUE
      DO 288 I=IX,LAS
      DOD=TTF(I)-TTF(INTT(KM))
      FCO=EXP(-1.*CON2(KM))
      OBS(I)=1.-EXP(-1.*CH(I))
      FXX=-1.*(RAMDA(K)*DOD**SHAPE(K)+CON1(KM)*DOD)
      FITT(I)=1.-EXP(FXX)*FCO
```

```
      AAMU(I)=OBS(I)-FITT(I)
      WRITE(M,289) TTF(I),OBS(I),FITT(I),AAMU(I)
  289 FORMAT(4X,F10.1,15X,F15.7,15X,F15.7,15X,F15.7)
  288 CONTINUE
  293 CONTINUE
  285 CONTINUE
      DO 291 I=1,KFAIL
      IP=I+KFAIL
      IPP=I+KFAIL*2
      PMAT(I)=TTF(I)
      PMAT(IP)=OBS(I)
      PMAT(IPP)=FITT(I)
  291 CONTINUE
      WRITE(M,292)
  292 FORMAT(1H1,//,5X,"DISTRIBUTION PLOTTING  (1) IS OBSERVED DITRIBUT
     + ION"/,29X,"(2) IS FITTED DITRIBUTION ")
      CALL PLOTT(NAME,PMAT,KFAIL,3,0,0)
      GO TO 161
  999 WRITE(M,998)MODET
  998 FORMAT(//,5X,"          MODEL",I2,"   FAIL")
  405 CONTINUE
  161 STOP
      END
```

```
      CURVES.
      USER(*)
      FTN5.
      LGO.
C     MAIN PROGRAM
C
C
C
      PROGRAM CURVES (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C
C
      COMMON X(200),ORIGIN,N,T(200),OP(9),MAX
      DIMENSION Y(200),SOM(10,200),ROI(200),XMN(6,200)
      INTEGER T,ORIGIN,OP,CASE
C
C
      CASE=0
    1 CASE=CASE+1
      IF (CASE .GE. 4.)GO TO 99
      READ(5,100)N,ORIGIN,(OP(I),I=1,7),ND1,OP(8),ND2,OP(9),NB,
     + IN,NP
  100 FORMAT(8I5)
      IF (N.LE.0)GO TO 99
      WRITE(6,110)CASE,N,ORIGIN
  110 FORMAT(1H1,5(/),T30,43(1H$)/T30,"$",T72,"$",/T30,
     + " S SHAPE CURVE ANALYSIS AND FORCASTING",
     + /T30,"$",T72,"$"
     + ,/T30,43(1H$)///,//T31,"CASE NO. ",I3,
     + " / NUMBER OF DATA ",I3," /STARTING PERIOD ",I5)
      READ(5,113) (X(I),I=1,N)
  113 FORMAT(7F7.4)
      WRITE(6,120)(J,OP(J),J=1,9)
  120 FORMAT(/10X,"OPTION NO INPUT VALUE"//(14X,I1,15X,I3))
      WRITE(6,121) ND1,ND2,NB,IN,NP
  121 FORMAT(//10X,"ND1 =",I4,/ 10X, "ND2 =",I4,/10X," NB =",I4,
     + /10X,"IN =" ,I4,/ 10X," NP =",I4, /////)
      WRITE(6,113)(X(I),I=1,N)
  125 FORMAT(1H1///// 10X,"TIME    OBSERVED VALUE(X)     3 PERIOD", "
     + MOVING AVERAGE       5 PERIOD MOVING AVERAGE        7PERIOD","
     + MOVING AVERAGE",//40X," SUM          AVERAGE          SUM"
     +,"          AVERAGE          SUM          AVERAGE"/)
  169 FORMAT(1H1///10X,"FORCAST VALUE BY ARITHMETIC AND GEOMETRIC "
     + ," AVERAGE INCREASE RATE ",////10X,"TIME",5X," ARITHMETIC",8X,
     + "GEOMETRIC "/)
      IF(OP(1).NE.1)GO TO 50
      WRITE(6,125)
```

```
      DO 30 M=2,6,2
      NM=N-M
      DO 10 L=1,N
      XMN(M,L)=0.
   10 SOM(M,L)=0.
      DO 30 I=1,NM
      IM=I+M
      DO 20 J=I,IM
      K=(I+IM)/2
      SOM(M,K)=SOM(M,K)+X(J)
   20 CONTINUE
      XMN(M,K)=SOM(M,K)/(M+1)
   30 CONTINUE
      DO 35 I=1,N
      T(I)=ORIGIN+I-1
   35 WRITE(6,130)T(I),X(I),(SOM(M,I),XMN(M,I),M=2,6,2)
  130 FORMAT(9X,I5,3X,E18.5,T39,E10.4,E13.5,3X,E10.4,
     + E13.5,3X,E10.4,E13.5)
      K=I+1
      IF(OP(2).NE. 1..OR. OP(2) .NE. 3) GO TO 51
      IF(OP(2) .NE. 1)GO TO 40
      ORIGIN=ORIGIN+1
      N=N+2
      DO 37 I=1,N
      T(I)=ORIGIN+I-1
   37 X(I)=XMN(2,1)
      GO TO 50
   40 IF(OP(2) .NE. 2)GO TO 41
      ORIGIN=ORIGIN+2
      N=N-4
      DO 38 I=1,N
      T(I)=ORIGIN+I-1
   38 X(I)=XMN(4,I)
      GO TO 50
   41 ORIGIN=ORIGIN+3
      N=N-6
      DO 39 I=1,N
      T(I)=ORIGIN+I-1
   39 X(I)=XMN(6,I)
   51 SUMR=0.
      PROD=1.
      MAX=MAX0(N,OP(3))
      DO 55 I=2,N
      ROI(I)=X(I)/X(I-1)
      SUMR=SUMR+ROI(I)
      PROD=PROD+ROI(I)
   55 CONTINUE
      AR=SUMR/(N-1)*100.
      GR=PROD**(1./(N-1))*100.
   50 WRITE(6,150)(T(I),X(I),I=1,N)
  150 FORMAT(1H1//10X,"DATA SMOOTHENED BY MOVING AVERAGE"
     + ////10X,"TIME            VALUE OF X",//,(9X,I5,5X,E17.5))
      WRITE(6,160)AR,GR
  160 FORMAT(///10X,"AVERAGE RATE OF INCREASE",//15X,
     +  "ARITHMETIC MEAN",E15.5, /15X, "GEOMETRIC MEAN "
     + ,E15.5, /)
      IF(OP(3) .LE. 0)GO TO 65
      MN=N+OP(3)
```

```
      WRITE(6,169)
      NI=N+1
      DO 62 I=NI,MN
      T(I)=ORIGIN+I-1
      X(I)=X(I-1)*AR/100.
      Y(I)=X(I-1)*GR/100.
      WRITE(6,170)T(I),X(I),Y(I)
 62   CONTINUE
170   FORMAT(9X,I5,E15.3,2X,E15.3)
 65   IF(OP(4) .NE. 1)GO TO 75
      CALL LINE
 70   IF(OP(5) .NE. 1) GO TO 75
      CALL QUAD
 75   IF(OP(6) .NE. 1) GO TO 80
      CALL EXPO
 80   IF(OP(7).NE.1)GO TO 85
      CALL MEXP(ND1)
 85   IF(OP(8).NE.1)GO TO 90
      CALL GOMP(ND2)
 90   IF(OP(9) .NE. 1) GO TO 1
      CALL LOGST(NB,IN,NP)
      GO TO 1
 99   STOP
```

VITA

Chongman Park

Born : October 24, 1955. Seoul(KOREA)

Education : B.S - Bachelor of Industrial Engineering, Inha University,Korea, Feb. 1978

M.A - Master of production/operation management ,Yonsei university,Korea, Sept. 1983

M.S - Candidate of Industrial Engineering, Lehigh university,USA , Dec. Present,1986

Professional Experience : During 5 years, research experience in system analyses and design fields for defense science at korea Institute of Defense Analyses

During 1 years, Industrial experience in Pharmaseutic Co.,Korea