CHAPTER 2
SOFTWARE RELIABILITY-INTER FAILURE TIME DATA

2.1 Introduction

Software reliability is the probability of failure free operation of a computer program in a specified environment for a specified time. A failure is a departure of a program operation from program requirements. A software reliability model provides a general form in terms of a random process describing failures for characterizing software reliability or a related quantity as a function of experienced failures over time.

In this backdrop Musa and Okumoto (1984) suggested a new model for software reliability measurement based on execution time and named it as logarithmic Poisson execution time model (LPETM). Their results are applicable for calendar time as well. Motivated by their approach we develop a new software reliability model in this chapter and present its performance in measuring software reliability. Because our proposed model runs parallel to LPETM as a ready reference, we describe LPETM briefly in this section. The development of our model and the reliability measurement characteristics associated with it are given in Sections 2.2 through 2.5.

As narrated in Section 1.3 of Chapter 1, a software reliability growth model (SRGM) can be described by its failure process as an NHPP. LPETM is based on the notion that failure rate decreases as a result of repair action following earlier failures, that frequently occur.

Let \([N(t), t \geq 0], m(t), \lambda(t)\) be the counting process, mean value function and intensity function of a software failure phenomenon. LPETM was derived with the following assumptions:
Assumption (i): There is no failure observed at time $t = 0$. That is, $N(0) = 0$ with probability one.

Assumption (ii): The failure intensity $\lambda(t)$ will decrease exponentially with the expected number of failures experienced up to time $t$ so that the relation between $m(t)$ and $\lambda(t)$ is

$$\lambda(t) = \lambda_0 e^{-\theta m(t)}$$

(2.1.1)

where $\lambda_0$ and $\theta$ are the initial failure intensity and the rate of reduction in the normalized failure intensity per failure respectively.

Assumption (iii): For a small interval $\Delta t$ the probabilities of one, more than one failure during $[t, t+\Delta t]$ are $\lambda(t). \Delta t + O(\Delta t)$ and $O(\Delta t)$, respectively, where $\frac{O(\Delta t)}{\Delta t} \to 0$ as $\Delta t \to 0$. Note that the probability of no failure during $[t, t+\Delta t]$ is given by $1 - \lambda(t) \Delta t + O(\Delta t)$. Using assumptions (i) and (ii) the mean value function and intensity function are derived to be

$$m(t) = \frac{1}{\theta} \log(\lambda_0 \theta t + 1)$$

(2.1.2)

$$\lambda(t) = \frac{\lambda_0}{(\lambda_0 \theta t + 1)}$$

(2.1.3)

It may be noted that $m(t) \to \infty$ as $t \to \infty$. Hence this model is also called 'infinite failures' model. Using assumptions (i) and (iii) the probability distribution of the stochastic process $N(t)$ is given by

$$P[N(t) = y] = \left[\frac{m(t)}{y}\right]^y e^{-m(t)}$$

(2.1.4)

where $m(t)$ is given by equation (2.1.2). With these basic functions Musa and Okumoto (1984) gave expressions for conditional distribution of $N(t)$ given the experienced number of failures up to time $t$, the cumulative distribution function of time to the $i$'th failure, which is also the distribution of time to remove the first $i$ failures, conditional distribution function of the time to the $i$'th failure given the experienced failures before this time, reliability function, hazard function, maximum
likelihood estimation of parameters - \( \lambda, \theta \) based on failure intervals, number of failures per interval. Evaluation of the model in relation to exponential class, Weibull class, Pareto class, geometric family, inverse linear family, inverse second degree family and power family was also discussed. The findings are made applicable to calendar time component also by making the following three additional assumptions:

**Assumption (iv):** The place of testing at any time is constrained by one of three limiting resources: failure-identification (test team) personnel (I), failure-correction (original designer) personnel (F), or computer time (C).

**Assumption (v):** The rate of resource expenditures with respect to execution time can be approximated by:

\[
\frac{dX_k}{dt} = \theta k + m(k) \frac{dm(t)}{dt}, \quad k = I, F, C.
\]

**Assumption (vi):** The quantities of the available resources are constant for the remainder of the test period.

### 2.2 The Proposed Model

On lines of Goel and Okumoto (1979), let us specify that the mean value function \( m(t) \) is finite valued, non decreasing, non negative and bounded with the boundary conditions:

\[
m(t) = \begin{cases} 
0, & t = 0 \\
\alpha, & t \to \infty 
\end{cases}
\]

Here 'a' represents the expected number of software failures eventually detected. If \( \lambda(t) \) is the corresponding intensity function, the basic assumption of LPETM says that \( \lambda(t) \) is a decreasing function of \( m(t) \) as a result of repair action following early failures. We model this assumption as a relation between \( m(t) \) and \( \lambda(t) \) so that one can be expressed in terms of the other. Our proposed relation is

\[
\lambda(t) = \frac{b}{2a} \left[ a^2 - m^2(t) \right]
\]
where ‘b’ is a positive constant, serving the purpose of constant of proportional fall in \( \lambda(t) \). This relation indicates a decreasing trend for \( \lambda(t) \) with increase in \( m(t) \) - a characteristic similar to that of LPETM which requires exponential decrease of \( \lambda(t) \) with increase in \( m(t) \), whereas our proposed model requires a quadratic decrease of \( \lambda(t) \) with \( m(t) \).

From the fact that \( \lambda(t) \) is the derivative of \( m(t) \) we get the following differential equation

\[
\frac{dm(t)}{dt} = \frac{b}{2a} \left[ a^2 - m^2(t) \right]
\]

whose solution is

\[
m(t) = \frac{a\left(1-e^{-bt}\right)}{1+e^{-bt}}
\]

We propose an NHPP with its mean value function given in equation (2.2.1). Its intensity function is

\[
\lambda(t) = \frac{2abe^{-bt}}{(1+e^{-bt})^2}
\]

Our proposed model is derived on lines of Goel and Okumoto (1979) specifying a relation between \( m(t) \) and \( \lambda(t) \) as motivated from LPETM. Our model turns out to be the probability model of half logistic distribution of Balakrishnan (1985). This model is paid a considerable attention as a reliability model recently by many authors and some of those published works are Kantam et al. (1994), Kantam and Dharmarao (1994), Kantam and Rosaiah (1998), Kantam et al. (2000), Kantam and Srinivasarao (2004) and the references there in. In our present investigation we study and evaluate the performance of SRGM based on NHPP with mean value function given by (2.2.1) - a constant multiplier of half logistic cumulative distribution function. In short we abbreviate this as HLSRGM.
2.3 Reliability Performance Measures

Using the mean value function $m(t)$ derived in Section 2.2 for HLSRGM, the probability distribution of the counting process $N(t)$ is given by

$$P[N(t) = y] = \frac{[m(t)]^y}{y!} e^{-m(t)}, y = 0, 1, 2, \ldots$$  \hspace{1cm} (2.3.1)$$

where $m(t)$ is given by Equation (2.2.1)

Here $N(t)$ is random number of failures experienced by time ‘t’. Suppose that $y_e$ failures have been observed during $[0, t_e]$. Then the conditional distribution of $N(t)$ given $N(t_e) = y_e$ for $t > t_e$ is the distribution of number of failures during $[t_e, t]$. That is,

$$P[N(t) = y / N(t_e) = y_e] = \frac{[m(t) - m(t_e)]^{y - y_e}}{(y - y_e)!} x e^{-[m(t) - m(t_e)]}$$ \hspace{1cm} (2.3.2)$$

Let $X_k$ be the time between $K-1$ and $K^{th}$ failures of the software product. And $S_k$ be the time up to the $K^{th}$ failure. The probability that $X_k$ exceeds a real number ‘x’ given that the total time up to the $(K-1)^{th}$ failure is equal to ‘s’ is

$$R_{X_k} / S_{K-1}(x / s) = e^{-[m(s+x) - m(s)]}$$ \hspace{1cm} (2.3.3)$$

Equation (2.3.3) is called the software reliability.

If we take the negative of derivative of Equation (2.3.3) with respect to x, we get the conditional density as

$$f(X_k / S_{K-1}) = \lambda(s+x)e^{-[m(s+x) - m(s)]}$$ \hspace{1cm} (2.3.4)$$

The hazard rate is given by the ratio of density to reliability which becomes

$$h(X_k / S_{K-1}) = \lambda(s + x)$$ \hspace{1cm} (2.3.5)$$

In the equations (2.3.3), (2.3.4) and (2.3.5) the expressions for $m(t)$ and $\lambda(t)$ are given by the respective quantities of HLSRGM.
2.4 Estimation Based on Inter Failure Times

We may recall that the mean value function and intensity function of HLSRGM are given by

\[ m(t) = \frac{a(1-e^{-bt})}{(1+e^{-bt})}, \quad a > 0, b > 0, t \geq 0 \]  \hspace{1cm} (2.4.1)

\[ \lambda(t) = \frac{2abe^{-bt}}{(1+e^{-bt})^2} \]  \hspace{1cm} (2.4.2)

The constants 'a', 'b' which appear in the mean value function and hence in NHPP, in intensity function (error detection rate) and various other expressions are called parameters of the model. In order to have an assessment of the software reliability 'a', 'b' are to be known or they are to be estimated from a software failure data.

Suppose we have 'n' time instants at which the first, second, third..., nth failures of a software are experienced. In other words if \( S_k \) is the total time to the kth failure, \( s_k \) is an observation of random variable \( S_k \) and 'n' such failures are successively recorded. The joint probability of such failure time realizations \( s_1, s_2, s_3, ..., s_n \) is

\[ L = e^{-m(s_n)} \prod_{k=1}^{n} \lambda(s_k) \]  \hspace{1cm} (2.4.3)

The function given in equation (2.4.3) is called the likelihood function of the given failure data. Values of 'a', 'b' that would maximize \( L \) are called maximum likelihood estimators (MLEs) and the method is called maximum likelihood (ML) method of estimation. Accordingly 'a', 'b' would be solutions of the equations

\[ \frac{\partial \log L}{\partial a} = 0, \quad \frac{\partial \log L}{\partial b} = 0. \]

Substituting the expressions for \( m(t) \), \( \lambda(t) \) given by equations (2.4.1) and (2.4.2) in equation (2.4.3), taking logarithms, differentiating with respect to 'a', 'b' and equating to zero, after some joint simplification we get
\[
a = n \left[ \frac{1 + e^{-bs_n}}{1 - e^{-bs_n}} \right] \quad (2.4.4)
\]

\[
g(b) = \sum_{k=1}^{n} s_k - \frac{n}{b} - 2 \sum_{k=1}^{n-1} s_k e^{-bs_k} \frac{2s_e e^{-bs}}{1 + e^{-bs}} \left[ 1 - \frac{n}{1 - e^{-bs}} \right] = 0 \quad (2.4.5)
\]

MLE of 'b' is an iterative solution of equation (2.4.5) which when substituted in equation (2.4.4) gives MLE of 'a'. In order to get the asymptotic variances and covariance of the MLEs of 'a', 'b' we needed the elements of the information matrix obtained through the following second order partial derivatives.

\[
\frac{\partial^2 \log L}{\partial a^2} = -\frac{n}{a^2} \quad (2.4.6)
\]

\[
\frac{\partial^2 \log L}{\partial a \partial b} = \frac{2s_a e^{-bs_n}}{(1 + e^{-bs_n})^2} \quad (2.4.7)
\]

\[
\frac{\partial^2 \log L}{\partial b^2} = -\frac{n}{b^2} - 2as_e e^{-bs} \left[ \frac{2e^{-bs} - (1 + e^{-bs})}{(1 + e^{-bs})^3} \right] + 2 \sum_{k=1}^{n} s_k^2 e^{-bs_k} \left[ \frac{s_k e^{-bs} - (1 + e^{-bs})}{(1 + e^{-bs})^2} \right] \quad (2.4.8)
\]

Expected values of negatives of the above three derivatives would be the following information matrix

\[
E \begin{bmatrix}
\frac{\partial^3 \log L}{\partial a^3} & -\frac{\partial^3 \log L}{\partial a \partial b} \\
-\frac{\partial^3 \log L}{\partial a \partial b} & \frac{\partial^3 \log L}{\partial b^3}
\end{bmatrix}
\]

Inverse of the above matrix is the asymptotic variance covariance matrix of the MLEs of 'a', 'b'. Generally the above partial derivatives evaluated at the MLEs of 'a', 'b' are used to get consistent estimator of the asymptotic variance covariance.
matrix. However, in order to overcome the numerical iterative way of solving the log
likelihood equations and to get analytical estimators rather than iterative, some
approximations in estimating the equations can be adopted from Kantam and
Dharmarao (1994), Kantam and Sriram (2001) and the references therein. We use
two such approximations here to get modified MLEs of ‘a’ and ‘b’.

Equation (2.4.5) can be written as

\[ b \sum_{k=1}^{n} s_k - n - 2 \sum_{k=1}^{n} \frac{z_k e^{-z_k}}{1 + e^{-z_k}} + \frac{2nz_n e^{-z_n}}{1 - e^{-2z_n}} = 0 \]  

(2.4.9)

Let us approximate the following expressions in the L.H.S of equation (2.4.9) by
linear functions in the neighborhoods of the corresponding variables.

\[ \frac{z_k e^{-z_k}}{1 + e^{-z_k}} = \alpha_k + \beta_k z_k, k = 1, 2, \ldots, n. \]  

(2.4.10)

\[ \frac{z_n e^{-z_n}}{1 - e^{-2z_n}} = \gamma_n + \delta_n z_n \]  

(2.4.11)

where the slopes \( \alpha_k, \gamma_n \) and intercepts \( \beta_k, \delta_n \) in equations (2.4.10) and (2.4.11) are to
be suitably found. With such values equations (2.4.10) and (2.4.11) when used in
equation (2.4.9) would give an approximate MLE for ‘b’ as

\[ \hat{b} = \frac{n + 2 \sum_{k=1}^{n} \alpha_k - 2n \gamma_n}{\sum_{k=1}^{n} s_k - 2 \sum_{k=1}^{n} \beta_k s_k + 2n \delta_n s_n} \]  

(2.4.12)

We suggest two methods to get the slopes and intercepts in the R.H.S of equations
(2.4.10) and (2.4.11)

Method I.

Let \( F(z) = \frac{1 - e^{-z}}{1 + e^{-z}} \)

\[ P_i = \frac{i}{n+1}, i = 1, 2, \ldots, n \]
Given a natural number \( n \) we can get the values of \( u'_i \) and \( u''_i \) by inverting the above equations through the function \( F(z) \). If \( G(.) \), \( H(.) \) are the symbols for the L.H.S of equations (2.4.10) and (2.4.11) we get

\[
\beta_k = \frac{G(u'_{k}) - G(u'_{k})}{u''_{k} - u''_{k}} \quad , k = 1, 2, ..., n
\]

\[
\alpha_k = G(u'_{k}) - \beta_k u'_{k} \quad , k = 1, 2, ..., n
\]

\[
\delta_n = \frac{H(u''_{n}) - H(u''_{n})}{u''_{n} - u''_{n}}
\]

\[
\gamma_n = H(u'_{n}) - \delta_n u'_{n}. \]

It can be seen that the evaluation of \( \alpha_k, \beta_k, \delta_n, \gamma_n \) are based on only a specified natural number \( n \) and can be computed free from any data. Given the data observations and sample size using these values along with the sample data in equation (2.4.12) we get an approximate MLE of \( b \). Equation (2.4.4) gives approximate MLE of \( a \).

**Method II.**

In this method we consider the Taylor series expansion up to the first derivative of the functions \( G(.) \), \( H(.) \) around \( u_i \), \( (i = 1, 2, ..., n) \) and \( u_n \) respectively where \( u_i \) is the solution of the following equations.
Accordingly the expressions for the slopes and intercepts of equation (2.4.10) and (2.4.11) are given by

\[ \beta_k = G'(u_i), \alpha_k = G(u_i) - \beta_i u_i, i = 1, 2, \ldots, n. \]

\[ \delta_n = H'(u_n), \gamma_n = H(u_n) - \delta_n u_n. \]

where \( G'(\cdot), H'(\cdot) \) are the derivatives of \( G(\cdot), H(\cdot) \) respectively. These are

\[ G'(z) = \frac{e^{-z} - z - e^{-z} - z - e^{-z} - e^{-z}}{(1 + e^{-z})^2} \]

\[ H'(z) = \frac{e^{-z} - z - e^{-z} - z - e^{-z} - e^{-z}}{(1 - e^{-z})^2} \]

As mentioned in method I, here also the slopes and intercepts can be calculated free from magnitudes of data observations. These values when used in equation (2.4.9) would give another approximate MLE of \( b \) and subsequently another approximate MLE of the parameter \( a \) from equation (2.4.4). Thus, given a software failure data in the form of inter failure times of a finite number of failures in terms of \( X_k \) or \( S_k \) we can get point estimates of the parameters \( a, b \) of HLSRGM by exact ML method as iterative solution of equation (2.4.5) and an approximate MLE by two methods of approximation as described above.

In the case of approximations, the basic principle is, some expressions of the estimating equations are approximated by linear functions. The larger the size of the sample the closer the approximation. Hence the exactness of the approximation becomes finer for large values of \( n \). Therefore, approximate MLEs and exact MLEs differ little, when the number of experienced software failures is large. As supported by Bhattacharya (1985), we can consider approximated expressions of log L for the
elements of the information matrix in order to get the asymptotic variances and covariance of the estimators of a, b. Hence

$$\frac{\partial^2 \log L}{\partial a^2} = \frac{-n}{a^2}$$

(2.4.13)

$$\frac{\partial^2 \log L}{\partial a \partial b} = \frac{2be^{-bs}}{(1 + e^{-bs})^2}$$

(2.4.14)

$$\frac{\partial^2 \log L}{\partial b^2} = \frac{-n}{b^2} \left[ -2 \sum_{k=1}^{n} s_k b e^{-bs} + 2s_k ab e^{-bs} \right] \frac{(1 - e^{-bs})}{(1 + e^{-bs})^3}$$

(2.4.15)

Estimated variances and covariance of $\hat{a}$ and $\hat{b}$ are obtained from the elements of the inverse of the matrix evaluated at $\hat{a}, \hat{b}$.

\[
\begin{bmatrix}
-\frac{\partial^2 \log L}{\partial a^2} & -\frac{\partial^2 \log L}{\partial a \partial b} \\
-\frac{\partial^2 \log L}{\partial a \partial b} & -\frac{\partial^2 \log L}{\partial b^2}
\end{bmatrix}
\]

(2.4.16)

which can be used to get the estimated asymptotic variance of estimate of any reliability characteristic like mean value function, intensity function, reliability function etc., In all these cases a software reliability characteristic is a parametric function say $g(a, b)$. Estimated asymptotic variance of estimate of $g(a, b)$ is given by

\[
\text{Est. Asy. var.} g(\hat{a}, \hat{b}) = \left( \frac{\partial g}{\partial a} \right)^2 \text{Est. Asy. Var}(\hat{a}) + \left( \frac{\partial g}{\partial b} \right)^2 \text{Est. Asy. Var}(\hat{b}) + 2 \frac{\partial g}{\partial a} \frac{\partial g}{\partial b} \text{cov}(\hat{a}, \hat{b})
\]

(2.4.17)

With the help of asymptotic optimum properties of MLEs, we can get the $(1 - \alpha)\%$ equitailed confidence intervals for ‘a’, ‘b’ or any parametric function of ‘a’, ‘b’ using
the elements of inverse of the matrix (2.4.16) or the R.H.S of equation (2.4.17) evaluated at \( \hat{a}, \hat{b} \). That is,

\[
\hat{a} \pm z_{1 - \alpha / 2} \text{Est. Asy. Var}(\hat{a})
\]

\[
\hat{b} \pm z_{1 - \alpha / 2} \text{Est. Asy. Var}(\hat{b})
\]

\[
g(\hat{a}, \hat{b}) \pm z_{1 - \alpha / 2} \text{Est. Asy. Var}[g(\hat{a}, \hat{b})]
\]

where \( z_{1 - \alpha / 2} \) is the \((1 - \alpha / 2)^{th}\) percentile of standard normal distribution.

Considering the likelihood function say \( L(a, b) \) we can get confidence interval for \( L(a, b) \) using the result namely \( 2 \log \left[ \frac{L(\hat{a}, \hat{b})}{L(a, b)} \right] \) follows a chi-square distribution with 2 degrees of freedom. Accordingly, 100\((1 - \alpha)\)% equitailed confidence interval for the parametric function \( L(a, b) \) is

\[
L(\hat{a}, \hat{b}) e^{-\frac{U^*}{2}} < L(a, b) < L(\hat{a}, \hat{b}) e^{-\frac{L^*}{2}}
\]

(2.4.18)

where \( L^*, U^* \) are respectively \( \frac{\alpha}{2}, 1 - \frac{\alpha}{2} \) percentiles of a chi-square distribution with 2 degrees of freedom. The procedure of interval estimation described above can be adopted by using the exact MLEs of ‘a’, ‘b’ and the corresponding estimated variances, covariance of the MLEs with the help of equations (2.4.4) through (2.4.8).

**Illustration:** The joint and interval methods of estimation are explained by applying the results to the software failure data of Navel Tactical Data system (NTDS) borrowed from Jelinski and Moranda (1972), which is presented in the following table.
<table>
<thead>
<tr>
<th>Error No (check out)</th>
<th>Time between errors $x_k$ days</th>
<th>Cumulative time $S_n=\sum x_k$ days</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Production phase</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>2</td>
<td>12</td>
<td>21</td>
</tr>
<tr>
<td>3</td>
<td>11</td>
<td>32</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>36</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>43</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>45</td>
</tr>
<tr>
<td>7</td>
<td>5</td>
<td>50</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>58</td>
</tr>
<tr>
<td>9</td>
<td>5</td>
<td>63</td>
</tr>
<tr>
<td>10</td>
<td>7</td>
<td>70</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>71</td>
</tr>
<tr>
<td>12</td>
<td>6</td>
<td>77</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>78</td>
</tr>
<tr>
<td>14</td>
<td>9</td>
<td>87</td>
</tr>
<tr>
<td>15</td>
<td>4</td>
<td>91</td>
</tr>
<tr>
<td>16</td>
<td>1</td>
<td>92</td>
</tr>
<tr>
<td>17</td>
<td>3</td>
<td>95</td>
</tr>
<tr>
<td>18</td>
<td>3</td>
<td>98</td>
</tr>
<tr>
<td>19</td>
<td>6</td>
<td>104</td>
</tr>
<tr>
<td>20</td>
<td>1</td>
<td>105</td>
</tr>
<tr>
<td>21</td>
<td>11</td>
<td>116</td>
</tr>
<tr>
<td>22</td>
<td>33</td>
<td>149</td>
</tr>
<tr>
<td>23</td>
<td>7</td>
<td>156</td>
</tr>
<tr>
<td>24</td>
<td>91</td>
<td>247</td>
</tr>
<tr>
<td>25</td>
<td>2</td>
<td>249</td>
</tr>
<tr>
<td>26</td>
<td>1</td>
<td>250</td>
</tr>
<tr>
<td><strong>Test phase</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>87</td>
<td>337</td>
</tr>
<tr>
<td>28</td>
<td>47</td>
<td>384</td>
</tr>
<tr>
<td>29</td>
<td>12</td>
<td>396</td>
</tr>
<tr>
<td>30</td>
<td>9</td>
<td>405</td>
</tr>
<tr>
<td>31</td>
<td>135</td>
<td>540</td>
</tr>
<tr>
<td><strong>User phase</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>258</td>
<td>798</td>
</tr>
<tr>
<td><strong>Test phase</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>16</td>
<td>814</td>
</tr>
<tr>
<td>34</td>
<td>35</td>
<td>849</td>
</tr>
</tbody>
</table>
We confirm the suitability of our model to the data by a test of goodness of fit known as Q-Q plot. We have taken the first 26 observations for the Q-Q plot. The correlation coefficient between sample and population quantiles is 0.97 for the Goel-Okumoto model (GOM) (1975) and for our HLSRGGM also indicating that both the models fit equally good for the data. Solving equations (2.4.4) and (2.4.5) by Newton-Raphson (N-R) method for the NTDS data, (Program listing for N-R method is given), the iterative solutions for MLEs of 'b', 'a' are \( \hat{b} = 0.011827, \hat{a} = 29 \). Using equations (2.4.6), (2.4.7), (2.4.8) we get the estimated asymptotic variance covariance matrix of the MLEs of 'a', 'b' as

\[
\begin{bmatrix}
\hat{a} & \hat{b} \\
\end{bmatrix} =
\begin{bmatrix}
32.346894 & -0.000018747 \\
-0.000018747 & 0.00000475 \\
\end{bmatrix}
\]

Using equations of the approximate MLEs namely (2.4.10), (2.4.11), (2.4.12) we get the values of approximate MLEs of 'a', 'b' as follows:

<table>
<thead>
<tr>
<th>Method</th>
<th>a</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>27.59124139</td>
<td>0.014067485</td>
</tr>
<tr>
<td>II</td>
<td>28.31665782</td>
<td>0.012618822</td>
</tr>
</tbody>
</table>

The asymptotic dispersion matrix \( \hat{a}, \hat{b} \) by Method I is

\[
\begin{bmatrix}
\hat{a} & \hat{b} \\
\end{bmatrix} =
\begin{bmatrix}
29.27987 & 0.00000176 \\
0.000000176 & 0.000007611 \\
\end{bmatrix}
\]

The asymptotic dispersion matrix \( \hat{a}, \hat{b} \) by Method II is

\[
\begin{bmatrix}
\hat{a} & \hat{b} \\
\end{bmatrix} =
\begin{bmatrix}
30.83973515 & 0.000000187 \\
0.000000187 & 0.000006124 \\
\end{bmatrix}
\]

Various quantities of interest can be obtained by substituting the estimates of 'a', 'b' in the appropriate equations of this section.


2.5 Evaluation of the Model

In this section we make an attempt to evaluate the performance of our model in relation to the GOM by two criteria namely Mean square Error (MSE) and Akaike's Information criterion (AIC) (Akaike, 1974), defined as follows:

\[
\text{MSE} = \frac{1}{n-k} \sum_{i=1}^{n} [m(t_i) - \hat{m}(t_i)]^2
\]

\[
\text{AIC} = -2 \log L \left( \frac{\hat{a}, \hat{b}}{s_1, s_2, \ldots, s_n} \right) + 2k
\]

where \( \hat{m}(t_i) \) is the estimated value of \( m(t) \) at estimates of \( 'a', 'b' \); \( K \) is the number of parameters estimated (here \( K = 2 \)). For the NTDS data the values of MSE, AIC for our model by the three methods of estimation and for GOM by exact ML method of estimation are as follows:

<table>
<thead>
<tr>
<th>HLSRGM</th>
<th>GOM (ML)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ML</td>
<td>MML1</td>
</tr>
<tr>
<td>MSE:</td>
<td>3.9999</td>
</tr>
<tr>
<td>AIC:</td>
<td>168.60</td>
</tr>
</tbody>
</table>

We find that MSE, AIC of our model are smaller than those of GOM. Regarding methods of estimation within our model we find that MMLE based on method-II of our model gives a least value of MSE. Hence method-II is preferable.

As mentioned in Section 2.1 one of the basic assumptions common to LPETM and HLSRGM is that the failure intensity decreases non-linearly with the expected number of failures experienced. Accordingly the specific relations between mean value function and intensity function of LPETM, HLSRGM are
LPETM: $\lambda(t) = \lambda_0 e^{-\theta m(t)}$ \hspace{1cm} (2.5.1)

HLSRGM: $\lambda(t) = \frac{b}{2a} \left[ a^2 - m^2(t) \right]$ \hspace{1cm} (2.5.2)

The solutions of these two relations when solved for $m(t)$ (using the fact $m'(t) = \lambda(t)$) would be

LPETM: $m(t) = \frac{1}{\theta} \log(\lambda_0 \theta t + 1)$ \hspace{1cm} (2.5.3)

HLSRGM: $m(t) = \frac{a(1-e^{-bt})}{(1+e^{-bt})}$ \hspace{1cm} (2.5.4)

The limiting values of $m(t)$ as $t \to \infty$ in the above two equations are $\infty, a$ respectively. Though the theoretical basis of these two models is similar, the asymptotic behavior of the mean value functions is different.

We know that the mean value function of GOM is given by

GOM: $m(t) = a\left(1 - e^{-bt}\right)$ \hspace{1cm} (2.5.5)

and its limiting value is 'a'. Thus HLSRGM has a similarity with GOM with respect to the mean value function in the limit, and another similarity with LPETM in developing the mean value function as presented in equations (2.5.3), (2.5.4). We therefore, thought of comparing the relative suitability of these three models for a live data.

Reparameterizing and rewriting the mean value functions of the three models we can get the following equations.

GOM: $-\log\left(1 - \frac{m(t)}{a}\right) = bt$ \hspace{1cm} (2.5.6)

LPETM: $e^{\frac{m(t)}{a}} - 1 = bt$ \hspace{1cm} (2.5.7)

HLSRGM: $-\log\left[ \frac{a - m(t)}{a + m(t)} \right] = bt$ \hspace{1cm} (2.5.8)
where in equation (2.5.7), 'a' is the notation for 1/\theta and 'b' is the notation for \lambda_0 \theta of equation (2.5.3). If the expressions on the LHS of the equations (2.5.6) through (2.5.8) are considered as dependent on 't', these indicate that each LHS is a linear function of 't' with slope 'b'. Hence if observations on t, m(t) in a live data are available and the value of 'a' for that data is known or assigned by a specific procedure the correlation between each [LHS, 't'] indicates the strength of closeness between the data and the respective model. Because, in a way 'a' is the limiting value of m(t), a reasonable substitute for 'a' can be the maximum number of experienced faults by a given software in the given maximum time. For the NTDS data we may take 'a' as 34 so that the values of the correlation coefficient between the following pairs of variables,

GOM: \[ t, -\log \left[ 1 - \frac{m(t)}{\hat{a}} \right] \]

LPETM: \[ \frac{m(t)}{t, e^{\hat{a}} - 1} \]

HLSRGGM: \[ t, -\log \left[ \frac{\hat{a} - m(t)}{\hat{a} + m(t)} \right] \]

are -0.3954, 0.88, 0.9537 respectively where \( \hat{a} = 34 \); t, m(t) are columns 3 and 1 of NTDS DATA table. This shows that our HLSRGGM is the best suitable model for the NTDS DATA with respect to explaining the reparameterized linear relations given in Equations (2.5.6), (2.5.7), and (2.5.8).
PROGRAM FOR MLEs OF 'a' AND 'b' IN HLSRGM USING NEWTON-RAPHSON (N-R) METHOD

/*N-R METHOD FOR M.L.E'S OF a & b of Inter Failure time DATA*/
#include "stdio.h"
#include "conio.h"
#include "math.h"
define N 26
double g(double b, int s[], int sn,int sk);
double gdash(double b, int s[], int sn);
main()
{
    int i,sk;
    int s[N]={9,21,32,36,43,45,50,58,63,70,71,77,78,87,91,92,95,98,104,105,116,149,156,247,249,250};
    double s_avg,g1,g2,a;
    double b[25];
    clrscr();
    sk=0, s_avg=0.0;
    printf("n**********NEWTON-RAPHSON METHOD**************");
    printf("n Input 26 observations :n");
    for(i=0;i<N;i++)
    {
        sk=sk+s[i];
    }
    s_avg=sk/N;
    b[0]=log(4)/s_avg;
    //clrscr();
    printf("n b =%fn",b[0]);
    i=1;
    do
    {
        i=i+1;
        g1=g(b[i],s,s[N-1],sk);
        g2=gdash(b[i],s,s[N-1]);
        b[i+1]=b[i]-(g1/g2);
        printf("nb[%d]= %fb[%d]= %f",i,b[i],i+1,b[i+1]);
        printf("n\t\t\t|b[%d]-b'%d]| = %f",i+1 ,i,fabs(b[i+1]-b[i]));
        while((fabs(b[i+1]-b[i])>=0.00001));
        a=N*((1+exp(-b[i+1]*s[N-1]))/(1-exp(-b[i+1]*s[N-1])));
        printf("n\n%b[%d] = %f is the M.L.E of b = %f a=%f",i+1,b[i+1],b[i+1],a);
        printf("n****************************************************************");
        getch();
    }
/*Function for calclating g(b)*/
double g(double b, int s[N], int sn, int sk)
{
    int k;
    double v3=0.0, c3, c4, g_val;
    for(k=0; k<N-1; k++)
    {
        v3=v3+((s[k]*exp(-b*s[k]))/(1+exp(-b*s[k])));
    }
    c3=2*v3;
    c4=(2*sn*exp(-(b)*sn)/(1+exp(-(b)*sn)))*((1-N/(1-exp(-(b)*sn))));
    g_val=sk-(N/b)-c3-c4;
    return g_val;
}
/*function for calculating g'(b)*/
double gdash(double b, int s[N], int sn)
{
    int k;
    double gdash_val, v2=0.0, c2, c3;
    for(k=0; k<N-1; k++)
    {
        v2=v2+((s[k]*s[k]*exp(-b*s[k]))/(1+exp(-b*s[k])))/((1+exp(-b*s[k]))*(1+exp(-b*s[k])));
    }
    c2=2*v2;
    c3=(2*sn*sn*exp(-(b)*sn))/*((1/(1+exp(-(b)*sn)))/
        (1-exp(-2*b*sn))*/(1-exp(-2*b*sn));
    gdash_val=(N/(b*b))+c2+c3;
    return gdash_val;
}