Interval Enclosures for Reliability Metrics

M.A. CAMPOS* and A.F. MENDONÇA

Received on October 14, 2015 / Accepted on May 30, 2016

ABSTRACT. The computation of reliability metrics, that are reliability function, mean time to failure, hazard rate function, involves real numbers. Therefore, numerical problems are generated due to the limitation of representing and operating with real numbers in computers. This paper is focused on computing intervals that bound numeric errors introduced during computation process of reliability metrics in digital machines for Exponential, Weibull and Normal failure distributions. Interval functions were proposed for controlling numeric errors in the computation of reliability metrics values of complex systems, based on interval mathematics and high accuracy arithmetic. The interval functions calculate interval enclosures, using Intlab toolbox, for real values of reliability metrics and the SHARPE software was used to validate the results. Analysis of the numerical results obtained with the proposed functions showed that the intervals really enclose the real numbers calculated by the SHARPE software, indicating that these functions, in fact, are an alternative for auto-validating representation of these reliability values of complex systems.

Keywords: reliability, interval mathematics, high accuracy arithmetic, interval enclosures.

1 INTRODUCTION

Reliability is defined as the probability that a system (component) will function over some period of time [5]. Usually, it is used for model the system reliability the probability functions: reliability function and hazard rate function. In addition, a common parameter that is usefull in reliability analysis is the mean time to failure [5, 13, 23]. Throughout this paper, we call these three ways to quantify reliability aspects as reliability metrics.

As reliability metrics are real numbers, the computation of these values can generate numeric problems caused primarily by the limitation in handling real numbers in a digital machine [6]. The calculation process introduces round-off and truncation errors [8]. In order to control these types of numeric errors, this paper presents interval functions, which yield interval enclosures. These intervals encapsulate guarantee that the real values of reliability metrics, for sure, will be within the computed interval enclosures.

*Corresponding author: Marcilia Andrade Campos. Centro de Informática, UFPE – Universidade Federal de Pernambuco, 50740-560 Recife, PE, Brasil. E-mails: mac@cin.ufpe.br; afm4@cin.ufpe.br
Let $X$ be the set of all the real numbers $x$ that satisfy $\underline{X} \leq x \leq \overline{X}$. Then, $X$ is an interval [19, 20, 21, 26] with lower and upper bounds equal to $\underline{X}$ and $\overline{X}$, respectively. Thus, $X$ can be given by: $X = [\underline{X}, \overline{X}] = \{x \in [\underline{X}, \overline{X}]\}$. We denote the set of the closed real intervals by $\mathbb{IR}$. Interval functions use arithmetic operations [19, 20, 21] defined on $\mathbb{IR}$. These interval functions here defined receive as argument a real number $t \geq 0$, which represents the observation time of the system. Therefore, such functions are given by $F : \mathbb{R} \rightarrow \mathbb{IR}$.

The interval enclosures in this paper have high accuracy [12], i.e., they have the smallest possible width that can be represented in digital machines. High accuracy intervals have lower and upper bounds that belong to the floating-point system of the digital machine [3]. Let be the floating-point system $F \subseteq \mathbb{R}$. Since $X = [\underline{X}, \overline{X}]$ are a high accuracy interval and encloses the real number $x$, then (i) $\underline{X} \in F$, (ii) $\overline{X} \in F$, (iii) $\underline{X} \leq x \leq \overline{X}$. These conditions define the called high accuracy intervals [12]. To represent these type of intervals and the arithmetic operations between them in a floating-point system, it is mandatory the use of the directional rounding operators, $\triangle$ (upward direction) and $\nabla$ (downward direction). Each of these directional operators, when applied to a real number $x$, yield a number in the floating-point system closer to $x$, such that $\triangle x \in F$, $\triangle x \geq x$ and $\nabla x \in F$, $\nabla x \leq x$.

The probabilistic modeling of reliability is based on a nonnegative continuous random variable $T$, which represents the time to failure of a given system. In this paper, the computed interval encapsulates real values of reliability metrics real values of systems with Exponential, Weibull and Normal time to failure distribution. Analysis were performed using isolated components and complex systems [5] with components logically connected in series and in parallel [13, 23]. These procedures use the Simpson Interval Method [3] to yield interval enclosures for probability values of continuous random variables [1]. At the end of this work, intervals obtained by the proposed interval functions were compared with the numerical results of the Symbolic Hierarchical Automated Reliability and Performance Evaluator software (SHARPE) [23]. According to Hirel et al. [9] the SHARPE software is used in the reliability field and performance analysis, being used by universities and companies.

The works of [27, 28, 29, 30] address the same thematic of the current one. These studies present reliability interval analysis aimed to determine failure probabilities with interval parameters. In [4] it is outlined a focus in providing intervals to reliability based on Bayesian analysis. The work of [16] presents interval enclosures for reliability function values of systems with Exponential failure distribution. This paper has a new approach that is focused on computing intervals that bound numeric errors introduced during computation process of reliability metrics in digital machines for Exponential, Weibull and Normal failure distributions.

All the computation procedures here presented were performed in the following computational platform:

- Processor: Celeron(R) Dual-Core CPU T3000 1.80 GHz;
- Main Memory: 2.00 GB;
- Operational System: Windows 7 Starter.

In this paper, the interval computation is performed over the set of IEEE 754 binary64 floating-point numbers [10], which is suitable with the Matlab [15] toolbox Intlab [22]. There are others approaches to implement interval computing, as it is outlined in [7, 11]. We chose IntLab since this toolbox is widely used for users from more than 50 countries. Moreover, Intlab produces reliable results which is proved to be true under any circumstances, in particular covering rounding errors and all error terms [22].

2 RELIABILITY MODELING AND INTERVAL ENCLOSURES

In this section, we define interval functions and their implementations using the IntLab. These interval functions result in interval enclosures for reliability metrics of a system (reliability function, mean time to failure and hazard rate function). Throughout this section, it is considered that the continuous random variable $T$ takes only non-negative values. In this work, calculations are performed for $T$ with Exponential, Weibull and Normal distributions. It is also assumed that this variable can be described by its probability density function $f_T$ and cumulative distribution function $F_T$. Also, it is assumed that the systems addressed in this paper are not repairable [13]. So, it is not considered repairs and maintenance of any analyzed system.

2.1 Reliability Function

Consider a single component system in operation for the period $[0, t]$ and $T$ its time to failure. The real-valued reliability function of these system, at time $t$, $R(t)$, is given by:

$$R(t) = P(T \geq t), \quad (2.1)$$

where $P(T \geq t)$ is the probability of the system will not fail during the operation period $[0, t]$.

If $f_T$ is the probability density function of the variable $T$, then

$$R(t) = \int_{-\infty}^{t} f_T(t)dt = 1 - \int_{0}^{t} f_T(t)dt. \quad (2.2)$$

In this paper, the interval enclosures for real-valued reliability function are obtained using the Simpson Interval Method [3]. This method is used to calculate intervals that encompass defined integrals over an interval $[a, b]$ of a given real function $f$. In this method, we divide the integration interval $[a, b]$ in $p$ partitions whose limits are:

$$a = a_0 < a_1 < a_2 < \cdots < a_p = b.$$  

This process of partitioning results in $p$ intervals $A_i = [a_{i-1}, a_i]$, where $i = 1, 2, \ldots, p$. For each partition, one interval enclosure is obtained by

$$S_i = \int_{a_{i-1}}^{a_i} f(t)dt = \frac{w(A_i)}{6}(F(a_{i-1}) + 4f(m(A_i)) + F(a_i)) - \frac{w(A_i)^3}{2880}G(\xi_i), \quad (2.3)$$

where $i = 1, 2, \ldots, p$, $\xi_i \in A_i$, $w(A_i)$ and $m(A_i)$ are the width and the midpoint of $A_i$, respectively, and $F$ and $G$ are interval extensions [20, 21] for $f(t)$ and its 4-order derivative $f^{(4)}(t)$. The resulting interval of the method is given by the sum of $p$ intervals $S_i$ evaluated by (2.3).
This method is employed for calculating the interval $P_v([0, t])$ such that $\int_0^t f_T(t)\,dt = P(0 < T < t) \in P_v([0, t])$.

As we can see in the Lemma 2.3 of [2], considering $A$ be a specific probabilistic event, since $P(A) = c$, then $P_v(A) = X = [x_1, x_2]$ and $c \in X$. In $P_v([0, t])$, we consider that, in this case, the probabilistic event $A$ is $t \geq T \geq 0$.

This work presents implementations intended to compute intervals that encapsulate probabilities of random variables with Exponential, Weibull and Normal distributions [2]. All the developed implementations use Matlab [15] and IntLab toolbox [22].

For Exponential distribution, $exp1(a, b, p, \alpha)$ is the method developed for calculating $P_v([a, b])$.

The value $\alpha$ is the parameter of the distribution and the values of $a$ and $b$ are the lower and higher integration bounds, respectively, and thus,

$$\int_a^b f_T(t)\,dt = \int_a^b \alpha e^{-\alpha t}\,dt \in exp1(a, b, p, \alpha).$$

We have that $p$ is a positive integer representing the number of divisions of the interval $[a, b]$. Therefore, $p$ defines the resolution in which the Simpson Interval Method executes. As the value of $p$ increases, the width of the resulting intervals decreases. In [3], it is pointed out that the width of the calculated interval is proportional to $\frac{1}{p^5}$, what implies that doubling the value of $p$, the width of the result decreases by a factor of 32. Moreover, high accuracy, provided by the use of the IntLab toolbox, ensures that the intervals obtained have the least possible width that the Simpson Interval Method allows, considering a given $p$.

Besides Simpson Interval Method, there are others integral numeric integration interval methods based on Riemann formula, such as Moore and Yang’s Method [18]. The use of Simpson Interval Method has advantages over these methods, since, as described above, the Simpson’s approach produces width intervals that decrease 32 times when the precision parameter $p$ is doubled. In Moore and Yang’s Method, doubling the respective precision parameter also results the same decreasing rate in the width interval produced. Moreover, the Simpson Interval Method requires that the integrand is four times continuously differentiable [3], what is not a problem with Exponential, Weibull and Normal distributions.

In Appendix A, this work presents implementations of functions aimed at the calculation of $P_v([a, b])$ for Exponential, Weibull and Normal failure process distributions. The two latter mentioned implementations have signatures given by $weibull1(a, b, p, k, \lambda)$ and $normal1(a, b, p, \mu, \sigma)$. The values of $k$ and $\lambda$ are the shape and scale parameters of the Weibull distribution, respectively, while $\mu$ and $\sigma$ are the mean and standard deviation parameters of the Normal distribution.

**Definition 2.1.** Let $R_v$ be the interval function, called reliability enclosure, that encapsulates the real-valued reliability function $R$ of a system after the observation time $t$. Then,

$$R_v(t) = [1, 1] - P_v([0, t]).$$
We know that \( X + Y = \{ x + y | x \in X, y \in Y \}, \star \in \{+, -, \cdot, /\} \), where \( 0 \not\in Y \), when \( \star \) is /.
Since \( x \in X \) and \( y \in Y \), it follows that the real value \( x \star y \) also belong to the interval \( X + Y \). Therefore,

\[
x \star y \in X \star Y.
\] (2.4)

where \( 0 \not\in Y \), when \( \star \) is /.

Based on (2.4), we can prove that \( R_e(t) \) encompasses \( R(t) \), or \( R(t) \in R_e(t) \).

**Proposition 2.1.** Let \( T \) be the continuous random variable that represents the time to failure of a given system, and its real-valued reliability function given by \( R(t) = 1 - P(0 < T < t) \). Then, \( R(t) \in R_e(t) \).

**Proof.** \( 1 \in [1, 1] \) and \( P(0 < T < t) \in P_x([0, t]) \) (Equation 8.7 of [3]). Then \( 1 - P(0 < T < t) \in [1, 1] - P_x([0, t]) \Rightarrow R(t) \in R_e([0, t]). \)

Using IntLab toolbox, this article presents reliability enclosure implementations of single component systems. The Table 1 shows the signatures of these interval functions and its definitions for Exponential, Weibull and Normal distributions.

<table>
<thead>
<tr>
<th>Distribution</th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exponential</td>
<td>(\text{conf exp}(t, p, \mu) = [1, 1] - \exp(0, t, p, \mu))</td>
</tr>
<tr>
<td>Weibull</td>
<td>(\text{conf weibull}(t, p, k, \lambda) = [1, 1] - \text{weibull}(0, t, p, k, \lambda))</td>
</tr>
<tr>
<td>Normal</td>
<td>(\text{conf normal}(t, p, \mu, \sigma) = [1, 1] - \text{normal}(0, t, p, \mu, \sigma))</td>
</tr>
</tbody>
</table>

For instance, consider that the lifetime of a mechanical tool is modeled by a Weibull distribution with parameters \( k = 2 \) and \( \lambda = 10000 \) hours. The real probability that the mechanical tool will not fail in the first 8000 hours and related execution time computation are \( R(8000) = 0.527292424043049 \) and 0.089822 seconds.

The computation of an interval enclosure for \( R(8000) \) with Weibull distribution, considering \( p = 100 \), is given by

\[
R_e(8000) = [1, 1] - P_x([0, 8000])
= \text{conf Weibull}(8000, 100, 2, 10000)
= [1, 1] - \text{weibull}(0, 8000, 100, 2, 10000)
= [0.52729242404274, 0.52729242404335].
\]
148 INTERVAL ENCLOSURES FOR RELIABILITY METRICS

The Table 2 exhibits interval enclosures for the reliability of the mechanical tool cited above after 8000 working-hours, as \( p \) are incremented. The width and computation time of the intervals can also be observed.

Table 2: Reliability enclosure, Width of reliability enclosure and execution time, as \( p \) are incremented, for Weibull failure distribution system.

<table>
<thead>
<tr>
<th>( p )</th>
<th>( R_v(8000) )</th>
<th>( w(R_v(8000)) )</th>
<th>Execution time(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>([0.52729134642417, 0.52729322980637])</td>
<td>(&lt; 2.0 \cdot 10^{-4})</td>
<td>0.888380</td>
</tr>
<tr>
<td>10</td>
<td>([0.527292239237141, 0.52729245148924])</td>
<td>(&lt; 6.0 \cdot 10^{-8})</td>
<td>1.737012</td>
</tr>
<tr>
<td>50</td>
<td>([0.527292242403336, 0.527292242405247])</td>
<td>(&lt; 2.0 \cdot 10^{-11})</td>
<td>8.599330</td>
</tr>
<tr>
<td>100</td>
<td>([0.527292242404274, 0.527292242404335])</td>
<td>(&lt; 6.1 \cdot 10^{-13})</td>
<td>17.875280</td>
</tr>
<tr>
<td>200</td>
<td>([0.527292242404303, 0.527292242404307])</td>
<td>(&lt; 4.0 \cdot 10^{-14})</td>
<td>35.186983</td>
</tr>
</tbody>
</table>

Similarly, we bring another two examples with Exponential and Normal distribution of failures. The Table 3 shows the results in the reliability enclosures computation for an Exponential failure distribution system with parameter \( \alpha = 4 \). After 2 working-hours, the real-valued reliability function, \( R(2) \), and corresponding execution time computation are equal to \( 0.9999683282 \) seconds and 0.050303. The Table 4 exhibits data from the reliability enclosures computation for Normal failure distribution with parameters \( \mu = 8 \) and \( \sigma = 2 \). After 80 working-hours, the real-valued reliability function, \( R(80) \), and related execution time computation are equal to 0.999968328758167 and 0.087603 seconds.

Table 3: Reliability enclosure, Width of reliability enclosure and execution time, as \( p \) are incremented, for Exponential failure distribution system.

<table>
<thead>
<tr>
<th>( p )</th>
<th>( R_v(2) )</th>
<th>( w(R_v(2)) )</th>
<th>Execution time(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>([-0.085621563302, 0.278345187372]) \cdot 10^{-2}</td>
<td>(&lt; 3.7 \cdot 10^{-3})</td>
<td>0.423623</td>
</tr>
<tr>
<td>10</td>
<td>([0.28875911756531, 0.40249872715171]) \cdot 10^{-3}</td>
<td>(&lt; 1.2 \cdot 10^{-4})</td>
<td>0.771025</td>
</tr>
<tr>
<td>50</td>
<td>([0.33544508784544, 0.33548148452612]) \cdot 10^{-3}</td>
<td>(&lt; 3.7 \cdot 10^{-6})</td>
<td>3.624166</td>
</tr>
<tr>
<td>100</td>
<td>([0.33546206948853, 0.33546320689571]) \cdot 10^{-3}</td>
<td>(&lt; 1.2 \cdot 10^{-7})</td>
<td>7.292063</td>
</tr>
<tr>
<td>200</td>
<td>([0.3354621028183, 0.33546264584506]) \cdot 10^{-3}</td>
<td>(&lt; 3.6 \cdot 10^{-11})</td>
<td>14.282848</td>
</tr>
</tbody>
</table>

As we can note in Table 2, Table 3 and Table 4, when \( p \) increases, the width of \( R_v(A) \) decreases. We can also observe that doubling \( p \) the related interval width is decreased by a factor of 32. All the computed intervals indeed encapsulate the related real-valued reliability function.

We also observe in Table 2, 3 and 4 that the execution time increases linearly when \( p \) is incremented. The execution time computation of the real-valued reliability function is approximately ten times lower than the corresponding one of interval enclosure, considering \( p = 5 \), which implies a considerable overhead computation cost.
Table 4: Reliability enclosure, Width of reliability enclosure and execution time, as \( p \) are incremented, for normal failure distribution system.

<table>
<thead>
<tr>
<th>( p )</th>
<th>( R_v(80) )</th>
<th>( w(R_v(80)) )</th>
<th>Execution time(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>[0.99992849259341, 1.00247004587546]</td>
<td>&lt; 2.6 \cdot 10^{-3}</td>
<td>0.750204</td>
</tr>
<tr>
<td>10</td>
<td>[0.99996023275219, 1.00003965629226]</td>
<td>&lt; 8.0 \cdot 10^{-5}</td>
<td>1.338139</td>
</tr>
<tr>
<td>50</td>
<td>[0.99996831854230, 0.99996834395784]</td>
<td>&lt; 2.6 \cdot 10^{-8}</td>
<td>6.396252</td>
</tr>
<tr>
<td>100</td>
<td>[0.99996832840014, 0.99996832919439]</td>
<td>&lt; 8.0 \cdot 10^{-10}</td>
<td>14.061650</td>
</tr>
<tr>
<td>200</td>
<td>[0.99996832874636, 0.99996832877119]</td>
<td>&lt; 2.5 \cdot 10^{-11}</td>
<td>30.258739</td>
</tr>
</tbody>
</table>

2.2 Mean Time to Failure

Mean Time to Failure (\( T_{med} \)) of a system is given by the expected value \([17]\) of the random variable \( T \) that specifies its failure process. As pointed out by Kuo and Zuo \([13]\), \( T_{med} \) is the lifetime of a system when repairs are not allowed. This metric is

\[
T_{med} = \int_{0}^{+\infty} t \cdot f_T(t) \, dt. \tag{2.5}
\]

In Equation (2.5), we have that the limits of the integral are 0 e \(+\infty\), since \( T \) assumes only non-negative values.

**Proposition 2.2.** Let \( R(t) \) be the real-valued reliability function of a system. The Mean Time to Failure is

\[
T_{med} = \int_{0}^{+\infty} R(t) \, dt. \tag{2.6}
\]

**Proof.** On Ebeling [5]. \(\square\)

**Proposition 2.3.** Let \( T_{med} \) be the real number evaluated by Equation 2.6. If \( R(t) \) is a reliability function, so \( T_{med} < +\infty \).

**Proof.** The Equation (2.6) can be represented by the following infinitesimal sum:

\[
T_{med} = \lim_{n \to +\infty} \sum_{i=1}^{n} R(\xi_i) w([\xi_{i-1}, \xi_i]),
\]

where \( \xi_1 \in [x_0, x_1], \xi_2 \in [x_1, x_2], \ldots, \xi_n \in [x_{n-1}, x_n] \). The intervals \( [\xi_{i-1}, \xi_i] \), \( i = 1, 2, \ldots, n \), are identical partitions of the interval \( \lim_{n \to +\infty} [0, n] \), considering that \( x_0 = 0 \) and \( x_n = n \). We note that the width \( w([\xi_{i-1}, \xi_i]) \) is identical for all the partitions.

We know that \( R(+\infty) = 0 \), so, according to [25], we have that the infinitesimal sum is convergent, i.e., \( T_{med} < +\infty \). \(\square\)

Based on Equation 2.6, this paper defines the interval function which yields an interval enclosure for \( T_{\text{med}} \).

**Definition 2.2.** Let \( T \) be a random variable that represents the time to failure of a system and \( R_v(t) \) be the reliability enclosure of a specified real-valued reliability function \( R \). Consider the real number \( t_{\text{MAX}} \) and a positive integer \( n \), where \( n \to \infty \). Suppose the interval \([0, t_{\text{MAX}}]\) and its partitions \([x_0, x_1], [x_1, x_2], \ldots, [x_{n-1}, x_n]\), where \( x_0 = 0 \) and \( x_n = t_{\text{MAX}} \). Also consider that all partitions have the same width \( w = \frac{t_{\text{MAX}}}{n} \). So, the interval function that encloses the Mean Time To Failure, considering \( T \), is defined as

\[
T_{\text{med}} = \lim_{n \to +\infty} \left[ R_v(t_{m_1}) + R_v(t_{m_2}) + \cdots + R_v(t_{m_n}) \right] w. \tag{2.7}
\]

The values \( t_{m_1}, t_{m_2}, \ldots, t_{m_n} \) are, respectively, the midpoints of the partitions \([x_0, x_1], [x_1, x_2], \ldots, [x_{n-1}, x_n]\). For the implementation of the interval function for the mean time to failure, we use the parameters \( t_{\text{MAX}} \) and \( n \). So, we must choose the value of \( t_{\text{MAX}} \) in such way that \( R(t_{\text{MAX}}) \approx 0 \). The value of \( n \) represents the number of partitions of the interval \([0, t_{\text{MAX}}]\), defining the precision in which the interval enclosure is obtained.

The Table 5 shows the signatures of interval enclosures for mean time to failure for Exponential, Weibull and Normal distributions.

<table>
<thead>
<tr>
<th>Distribution</th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exponential</td>
<td>( \text{mttIntervalExp}(\alpha, t_{\text{MAX}}, n, p) )</td>
</tr>
<tr>
<td>Weibull</td>
<td>( \text{mttIntervalWeibull}(k, \lambda, t_{\text{MAX}}, n, p) )</td>
</tr>
<tr>
<td>Normal</td>
<td>( \text{mttIntervalNormal}(\mu, \sigma, t_{\text{MAX}}, n, p) )</td>
</tr>
</tbody>
</table>

In the Appendix A, we can observe more details about the implementations of the interval enclosures for the mean time to failure for Exponential, Weibull and Normal distributions.

To illustrate the use of \( T_{\text{med}} \) in bounding the related real-valued metric, assume that \( T \) is a Normal random variable with parameters \( \mu = 8 \) and \( \sigma = 2 \). In this case, suppose \( t_{\text{MAX}} = 16 \), which implies that \( R(16) = 6.6613 \cdot 10^{-16} \) or \( R(16) \approx 0 \), \( n = 50 \) and \( p = 5 \). The real value of mean time to failure for the considered system and the related execution time computation are \( T_{\text{med}} = 8 \) and 2.634465 seconds (this execution cost and all real value of mean time to failure measures were based on Equation 2.6).

The interval enclosure for \( T_{\text{med}} \) is given by

\[
T_{\text{med}} = \text{mttIntervalNormal}(8, 2, 16, 50, 5) = [7.98389618066430, 8.01618738659977].
\]

In fact, we have that \( T_{\text{med}} \in T_{\text{med}} \). However, it can be seen that when the condition \( R(t_{\text{MAX}}) \approx 0 \) is not satisfied, the interval obtained is not a enclosure. Table 6 presents computed intervals for
different values of $t_{MAX}$. It is observed that when $t_{MAX} \leq 12$ the intervals do not encapsulate the real value for the mean time to failure ($\mu = 8$).

Table 6: Interval enclosures for mean time to failure, considering $t_{MAX}$ variation.

<table>
<thead>
<tr>
<th>$t_{MAX}$</th>
<th>$T_{med}$</th>
<th>$R(t_{MAX})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>[7.98388828158800, 8.01618934801887]</td>
<td>$6.6613 \cdot 10^{-16}$</td>
</tr>
<tr>
<td>14</td>
<td>[7.99026276808957, 8.00936514800080]</td>
<td>$9.8659 \cdot 10^{-10}$</td>
</tr>
<tr>
<td>12</td>
<td>[7.98065312859909, 7.9816857449369]</td>
<td>$3.1671 \cdot 10^{-5}$</td>
</tr>
<tr>
<td>10</td>
<td>[7.83278842107344, 7.8348378155583]</td>
<td>$0.0227$</td>
</tr>
</tbody>
</table>

Once the computation of $T_{med}$ involves $n$ evaluations of $R_v$, the parameter $p$ is also used to encapsulate the mean time to failure. The Table 7 illustrates the variation of the interval widths and related execution times, since $p$ increases, considering $t_{MAX} = 20$ and $n = 50$.

Table 7: Interval enclosures for mean time to failure, considering $p$ variation, for Normal failure distribution system.

<table>
<thead>
<tr>
<th>$p$</th>
<th>$T_{med}$</th>
<th>$w(T_{med})$</th>
<th>Execution Time(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>[7.95139633268973, 8.06988007167273]</td>
<td>$&lt; 1.2 \cdot 10^{-1}$</td>
<td>46.126105</td>
</tr>
<tr>
<td>6</td>
<td>[7.97723844794519, 8.02123846333671]</td>
<td>$&lt; 4.5 \cdot 10^{-2}$</td>
<td>53.791730</td>
</tr>
<tr>
<td>7</td>
<td>[7.98380610504386, 8.01751702986318]</td>
<td>$&lt; 3.4 \cdot 10^{-2}$</td>
<td>62.819581</td>
</tr>
<tr>
<td>8</td>
<td>[7.99300780057101, 8.00531274894078]</td>
<td>$&lt; 9.4 \cdot 10^{-3}$</td>
<td>80.984591</td>
</tr>
<tr>
<td>9</td>
<td>[7.99595743932975, 8.00136274894078]</td>
<td>$&lt; 9.4 \cdot 10^{-3}$</td>
<td>89.759203</td>
</tr>
</tbody>
</table>

We also bring two examples with Exponential and Weibull distribution of failures. The Table 8 exhibits the results in the mean time to failure enclosure computation for Exponential failure distribution with parameter $\alpha = 0.01$. The real-valued mean time to failure and related execution time are equal to $T_{med} = 100$ and $5.925969$ seconds. The Table 9 shows data from mean time to failure enclosure computation for Weibull distribution of failures with parameters $k = 3$, $\lambda = 50$. The real-valued mean time to failure and corresponding execution time are equals to $T_{med} = 44.642793484075547$ and $8.679764$ seconds.

Table 8: Interval enclosures for mean time to failure, since $p$ increases, considering $t_{MAX} = 1000$ and $n = 150$, for Exponential failure distribution system.

<table>
<thead>
<tr>
<th>$p$</th>
<th>$T_{med}$</th>
<th>$w(T_{med})$</th>
<th>Execution Time(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>[99.390693839171, 101.241198466660]</td>
<td>$&lt; 1.9 \cdot 10^{-2}$</td>
<td>63.858197</td>
</tr>
<tr>
<td>6</td>
<td>[99.720562136152, 100.464238625810]</td>
<td>$&lt; 7.5 \cdot 10^{-3}$</td>
<td>76.539810</td>
</tr>
<tr>
<td>7</td>
<td>[99.851177010656, 100.195244622823]</td>
<td>$&lt; 3.5 \cdot 10^{-2}$</td>
<td>87.797895</td>
</tr>
<tr>
<td>8</td>
<td>[99.909610963321, 100.086088880303]</td>
<td>$&lt; 1.8 \cdot 10^{-2}$</td>
<td>98.893688</td>
</tr>
<tr>
<td>9</td>
<td>[99.938339164273, 100.036271870733]</td>
<td>$&lt; 1.0 \cdot 10^{-2}$</td>
<td>110.999310</td>
</tr>
<tr>
<td>10</td>
<td>[99.953555520054, 100.011383803893]</td>
<td>$&lt; 6.0 \cdot 10^{-2}$</td>
<td>122.218566</td>
</tr>
</tbody>
</table>

As it is shown in Tables 7, 8 and 9, the interval enclosures for mean time to failure bound corresponding real values mean time to failure. It also could be observed that the use of interval approach implies a high computation cost, since \( n \) evaluations of \( R \) must be done to achieve the interval proposed. The results brought in Tables 8 and 9, for example, the \( n \) value employed was 150 to guarantee precision in the interval enclosures.

### 2.3 Hazard rate function

Hazard rate function, \( \lambda(t) \), is a function that determines an instantaneous failure rate of a system in a specified instant \( t \). The definition of \( \lambda(t) \) \cite{5} is based on the conditional probability:

\[
P(t \leq T \leq t + \Delta t | T \geq t) = \frac{R(t) - R(t + \Delta t)}{R(t)}\, ,
\]

(2.8)

where \( R(t) > 0 \).

Equation 2.8 defines the chance of a system performs its intended function satisfactorily for a specified period of time \( \Delta t \), since do not occur faults in the interval \([0, t]\). So, the conditional probability presented in (2.8) divided by the time unit \( \Delta t \) is

\[
\frac{R(t) - R(t + \Delta t)}{R(t) \Delta t}.
\]

(2.9)

Equation 2.9 determines the failure rate per time unit of a system. Thus, the instantaneous failure rate is given by the conditional probability of fault occurrences per time unit, when \( \Delta t \to 0 \). Then,

\[
\lambda(t) = \lim_{\Delta t \to 0} \frac{-[R(t + \Delta t) - R(t)]}{\Delta t}, \quad \frac{1}{R(t)}
\]

(2.10)

\[
= -\frac{dR(t)}{dt} \frac{1}{R(t)} = f_T(t) \frac{1}{R(t)}
\]

Based on the Equation 2.10, this paper defines an interval function that encloses real value of the hazard rate function for a specified system.

Definition 2.3. Let $T$ be a random variable that represents the time to failure of a system and $R_v(t)$ be the reliability enclosure for its real-valued reliability function $R$. The interval enclosure for the real-valued hazard rate function in an instant of time $t$ is given by:

$$\lambda_v(t) = \left[ \nabla f_T(t), \triangle f_T(t) \right] R_v(t),$$

where $0 \not\in R_v(t)$.

Proposition 2.4. Let $\lambda(t) = \frac{f_T(t)}{R(t)}$. So, $\lambda(t) \in \lambda_v(t)$.

Proof. We know that $f_T(t) \in \left[ \nabla f_T(t), \triangle f_T(t) \right]$ and $R(t) \in R_v(t)$. Thus,

$$\frac{f_T(t)}{R(t)} \in \left[ \frac{\nabla f_T(t)}{R_v(t)}, \frac{\triangle f_T(t)}{R_v(t)} \right] \Rightarrow \lambda(t) \in \lambda_v(t).$$

when $0 \not\in R_v(t)$. □

The Table 10 illustrates the signatures of interval functions that encapsulate the real-valued hazard rate function for Exponential, Weibull and Normal distributions.

<table>
<thead>
<tr>
<th>Distribution</th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exponential</td>
<td>failureRateIntervalExp($t, p, \alpha$)</td>
</tr>
<tr>
<td>Weibull</td>
<td>failureRateIntervalWeibull($t, p, k, \lambda$)</td>
</tr>
<tr>
<td>Normal</td>
<td>failureRateIntervalNormal($t, p, \mu, \sigma$)</td>
</tr>
</tbody>
</table>

For instance, suppose $T$ is an Exponential random variable with parameter $\alpha = 0.01$. In this case, considering the instant $t = 100$, the real-valued hazard rate function and related execution time are equal to $\lambda(100) = \alpha = 0.01$ and 0.054298 seconds. The Table 11 illustrates the variation of the hazard rate interval widths and related execution times, since $p$ increases.

<table>
<thead>
<tr>
<th>$p$</th>
<th>$T_{med}$</th>
<th>$w(T_{med})$</th>
<th>Execution Time(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>[0.009999999900224, 0.01000000009145]</td>
<td>&lt; 2.0 $\cdot$ 10$^{-9}$</td>
<td>0.491079</td>
</tr>
<tr>
<td>10</td>
<td>[0.009999999996949, 0.01000000002916]</td>
<td>&lt; 6.0 $\cdot$ 10$^{-11}$</td>
<td>0.942291</td>
</tr>
<tr>
<td>50</td>
<td>[0.009999999999999, 0.010000000000001]</td>
<td>&lt; 2.0 $\cdot$ 10$^{-14}$</td>
<td>3.970694</td>
</tr>
<tr>
<td>100</td>
<td>[0.009999999999999, 0.010000000000001]</td>
<td>&lt; 2.0 $\cdot$ 10$^{-14}$</td>
<td>8.780558</td>
</tr>
<tr>
<td>200</td>
<td>[0.009999999999999, 0.010000000000001]</td>
<td>&lt; 2.0 $\cdot$ 10$^{-14}$</td>
<td>16.230988</td>
</tr>
</tbody>
</table>

Similarly, we bring another two examples with Normal and Weibull distribution of failures. The Table 12 shows the results in the hazard rate enclosures computation for a Normal failure distribution system with parameters $\mu = 8$ and $\sigma = 2$. After 10 working-hours, the real-valued
hazard rate function, \( \lambda(10) \), and the related execution time are equal to 1.524830884806074 and 0.095419 seconds. The Table 13 exhibits data from the hazard rate enclosures computation for Weibull failure distribution with parameters \( k = 2 \) and \( \lambda = 10000 \). After 8000 working-hours, the real-valued reliability function, \( \lambda(8000) \), and corresponding execution time computation are equal to \( 1.6 \cdot 10^{-4} \) and 0.101747 seconds.

Table 12: Interval enclosures for hazard rate function, considering \( p \) variation, for Normal failure distribution system.

<table>
<thead>
<tr>
<th>( p )</th>
<th>( T_{med} )</th>
<th>( w(T_{med}) )</th>
<th>Execution Time(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>[1.51863640703179, 1.53229224115192]</td>
<td>(&lt; 1.4 \cdot 10^{-4})</td>
<td>1.008209</td>
</tr>
<tr>
<td>10</td>
<td>[1.52460379156082, 1.52507613749420]</td>
<td>(&lt; 4.8 \cdot 10^{-4})</td>
<td>2.071646</td>
</tr>
<tr>
<td>50</td>
<td>[1.52483080854381, 1.52483096220953]</td>
<td>(&lt; 1.6 \cdot 10^{-4})</td>
<td>11.009327</td>
</tr>
<tr>
<td>100</td>
<td>[1.52483088240966, 1.52483088722032]</td>
<td>(&lt; 4.9 \cdot 10^{-6})</td>
<td>22.419977</td>
</tr>
<tr>
<td>200</td>
<td>[1.52483088473101, 1.52483088488142]</td>
<td>(&lt; 1.6 \cdot 10^{-7})</td>
<td>52.203656</td>
</tr>
</tbody>
</table>

Table 13: Interval enclosures for hazard rate function, considering \( p \) variation, for Weibull failure distribution system.

<table>
<thead>
<tr>
<th>( p )</th>
<th>( T_{med} )</th>
<th>( w(T_{med}) )</th>
<th>Execution Time(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>[0.159999975550201, 0.1600000000000032699903]</td>
<td>(&lt; 5.8 \cdot 10^{-12})</td>
<td>1.216593</td>
</tr>
<tr>
<td>10</td>
<td>[0.15999999167181, 0.1600000000000061035]</td>
<td>(&lt; 1.8 \cdot 10^{-11})</td>
<td>2.054348</td>
</tr>
<tr>
<td>50</td>
<td>[0.15999999999714, 0.16000000000000000002941]</td>
<td>(&lt; 5.8 \cdot 10^{-15})</td>
<td>9.105660</td>
</tr>
<tr>
<td>100</td>
<td>[0.15999999999999999, 0.1600000000000000000010]</td>
<td>(&lt; 2.1 \cdot 10^{-16})</td>
<td>17.899182</td>
</tr>
<tr>
<td>200</td>
<td>[0.159999999999999999999999999999991, 0.160000000000000000000000001]</td>
<td>(&lt; 2.0 \cdot 10^{-17})</td>
<td>35.608336</td>
</tr>
</tbody>
</table>

As the computation process of hazard rate function is similar to reliability function, the execution time of interval enclosure for hazard rate function has the same order of magnitude as the one of reliability enclosure. This could be noted comparing the execution time in examples brought by Section 2.1 and Section 2.3. In Table 11, we can observe that, for \( p > 50 \), the interval width achieved the least amplitude possible, which shows us that the computation cost of increasing \( p \) value does not result in a higher precision.

3 VALIDATION OF INTERVALS FOR COMPLEX SYSTEMS MODELED IN SHARPE

Complex systems [5] are compound by a set of components that act together to perform a specified function that could not be done with the absence of one of its components. We suppose that a component can present two distinct and not simultaneous states: available and unavailable. In the last one, we consider that a component has at least one fault that does not allow its correct operation. Thus, the state of each component can be represented by the discrete random variable
that is assigned to two different values. Let $x_i$ be the state of the $i$-esimo component of a specified system. Then,

$$x_i = \begin{cases} 
1, & \text{if the component } i \text{ is available}, \\
0, & \text{if the component } i \text{ is unavailable}.
\end{cases}$$

Consider the complex system $L$ with $n$ components. So, the vector $x = (x_1, x_2, \ldots, x_n)$ represents the state of all the components. As well as its components, $L$ also holds one of the two mentioned states. The state of $L$, called $\phi$, is a function of the vector $x$. Then, $\phi = \phi(x) = \phi(x_1, x_2, \ldots, x_n)$.

Thus, the function $\phi$ is determined by the components’ configuration of a given system. The following configurations are addressed in this section: series and parallel.

![Series configuration and Parallel configuration](image)

The Figure 1(a) indicates that one complex system with components in series [5, 13] exhibits only one critical path. Let $L$ be the complex system compound by $n$ components in series. The function $\phi$ that defines the state of $L$ is

$$\phi(x) = \prod_{i=1}^{n} x_i = \min(x_1, x_2, \ldots, x_n). \quad (3.1)$$

The Equation 3.1 defines that the complex system will fail whether only one of its components fail. So, considering that $L$ has two components, if only one component fails $L$ also will fail. Thus, assuming that the components’ failure process of $L$ are independent, the reliability function of $L$ is given by

$$R(t) = R_1(t) \times R_2(t) \times \cdots \times R_n(t) = \prod_{i=1}^{n} R_i(t). \quad (3.2)$$

The Figure 1(b) outlines the parallel configuration [5, 13], also called redundant. Systems with this configuration fail whether all of its components are in an unavailable state in a specified
instant of time. Let $L$ be a system formed by $n$ components connected in parallel. The function $\phi$ that defines the state of $L$ is

$$\phi(x) = 1 - \prod_{i=1}^{n} (1 - x_i) = \max(x_1, x_2, \ldots, x_n).$$

(3.3)

Therefore, there are $n$ critical paths in the redundant configuration. Thus, the probability that $L$ does not fail in the period of time $[0, t]$ is given by

$$R(t) = 1 - \prod_{i=1}^{n} [1 - R_i(t)].$$

(3.4)

This section is devoted to validate the proposed reliability enclosures for complex systems. In this way, the SHARPE software [9, 23] was employed in such manner that the real values of reliability function were generated and compared with corresponding interval enclosures. This analysis was accomplished using four case of studies, in which complex systems were modeled in SHARPE. The integration method used in SHARPE was Simpson’s rule [24].

The Matlab software enables two formats to display the computed intervals. The first one is format short which results fixed-decimal format with 4 decimal digits after decimal point numbers [15]. The last option, format long, yields fixed-decimal format with 15 decimal digits after the decimal point numbers. The SHARPE software only exhibits 8 significants decimals digits. Therefore, to confirm whether the obtained intervals bound the values computed by SHARPE, we use interval computation with format short configuration. In order to perform the validation of intervals with format long configuration, we use values computed by the previously specified computational platform, which results binary64 numbers displayed, in this paper, with 16 decimal digits after the decimal point. In all cases, we use $p = 200$ for all the interval calculations, since, as we will observe, the width of the obtained intervals achieved a reasonable precision.

The computation of the real-valued reliability function was based on systems that have components with series and parallel configurations simultaneously. According to [5], the reliability function computation of these systems involve their decomposition into subsystems. Thus, the reliability function’s evaluation of each subsystems are done and, then, the obtained values are combined in such way that the reliability function of all the system is calculated. Due to this computation process, the reliability metrics calculated related to complex systems is prone to errors caused by the propagation of round-off and truncation errors.

3.1 Case 1

Let $W$ be the system outlined in Figure 2 compound by three subsystems: $W_1$ (formed uniquely by the component block0), $W_2$ (formed by the components block1, block3, block4 and block5 connected in parallel) and $W_3$ (formed by the components block2, block6, block7, block8 and block9 also connected in parallel). Consider that the components of $W_1$ has Exponential distribution of failures with parameter $\lambda_0 = 5 \cdot 10^{-12}$ and that the components of $W_2$ and $W_3$ have
Exponential distribution with parameter $\alpha_1 = 2 \cdot 10^{-11}$. The real-valued reliability function of $W$, considering observation time $t = 20$, was evaluated by the SHARPE software and the previously specified computational platform and compared with corresponding interval enclosures (Table 14).

We can observe that the interval enclosure with format single configuration contains the value of $\tilde{R}(20)$. We also note that the computed interval with double precision encapsulates the real value obtained by the specified computational platform. In the case above, we observe that the real-valued reliability function calculated by SHARPE is rounded to 1. So, if we do not use the interval enclosure, the computed value by SHARPE is rounded to the value 1. This might forbid that comparisons among two systems with different reliability function values close to the real number 1 could be done, since the computed values are rounded to the same punctual value.

### 3.2 Case 2

Let $M$ be the system outlined in Figure 3 compound by five subsystems: $M_1$ (formed uniquely by the component block0), $M_2$ (formed by the components block1, block4 and block5 connected in parallel), $M_3$ (formed by the components block9, block6, block7 and block8 also connected in parallel), $M_4$ (formed by the components block3 and block11 connected in parallel) and $M_5$.

![Figure 2: Representation of the system $W$, modeled by the SHARPE software.](image)

<table>
<thead>
<tr>
<th></th>
<th>$\tilde{R}(20)$</th>
<th>1.00000000</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_v(20)$ (format single)</td>
<td>0.999999999900000</td>
<td></td>
</tr>
<tr>
<td>$R_v(20)$ (format long)</td>
<td>[0.99999999989999, 0.99999999990001]</td>
<td></td>
</tr>
</tbody>
</table>

Table 14: Comparison between the real-valued reliability function of the system $W$ evaluated by SHARPE, $\tilde{R}(20)$, and the specified computational platform, $R(20)$, and the reliability enclosure, considering short and long configurations.
(formed by the components block10 and block12 connected in series). Consider that the components of $M_1$ and $M_4$ have Exponential distribution of failures with parameter $\alpha_0 = 1.2$, the components of $M_2$ have Exponential distribution with parameter $\alpha_1 = 0.05$, the components of $M_3$ have Exponential distribution with parameter $\alpha_3 = 0.08$ and the components of $M_4$ have Exponential distribution of failures with parameter $\alpha_4 = 0.5$. The real-valued reliability function of $M$, considering observation time $t = 20$, was evaluated by the SHARPE software and the previously specified computational platform and compared with corresponding interval enclosures (Table 15).

![Figure 3: Representation of the system $M$, modeled by the SHARPE software.](image)

**Table 15:** Comparison between the real-valued reliability function of the system $M$ evaluated by SHARPE, $\tilde{R}(20)$, and the specified computational platform, $R(20)$, and the reliability enclosure, considering short and long configurations.

<table>
<thead>
<tr>
<th></th>
<th>$\tilde{R}(20)$</th>
<th>$R(20)$ format single</th>
<th>$R(20)$ format long</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_s(20)$</td>
<td>$[0.00000000]$</td>
<td>$[-0.3364, 0.3615] \cdot 10^{-29}$</td>
<td>$[-0.33637728687207, 0.36140951733383] \cdot 10^{-29}$</td>
</tr>
</tbody>
</table>

In this study of case (Case 2), we also observe that the computed intervals with single and long configuration enclose the real values obtained by the SHARPE software and the specified computational platform, respectively. In this case, for instance, if we use the reliability value computed by the SHARPE, any multiplication using this numeric value will also result in zero. On other hand, the use of the calculated intervals avoids this type of numeric problem.

### 3.3 Case 3

Let $G$ be the system outlined in Figure 4 compound by three subsystems: $G_1$ (formed uniquely by the component block0), $G_2$ (formed by the components block3 and block5 connected in parallel) and $G_3$ (formed by the parallel composition of the subsystems block1 and block6 in series, block2 and block7 in series and block4 and block8 also in series). Consider that the components of $G_1$ and $G_2$ have Weibull distribution of failures with parameters $\lambda_0 = 1$ and

\( k_0 = 5 \), the components of \( G_3 \) have Weibull distribution with parameters \( \lambda_1 = 1 \) and \( k_1 = 3 \).

The real-valued reliability function of \( G \), considering observation time \( t = 1 \), was evaluated by the SHARPE software and the previously specified computational platform and compared with corresponding interval enclosures (Table 16).

![Image](representation_of_the_system_G_modelled_by_the_SHARPE_software.png)

Table 16: Comparison between the real-valued reliability function of the system \( G \) evaluated by SHARPE, \( \tilde{R}(1) \), and the specified computational platform, \( R(1) \), and the reliability enclosure, considering short and long configurations.

<table>
<thead>
<tr>
<th></th>
<th>( R(1) )</th>
<th>( \tilde{R}_v(1) ) (format single)</th>
<th>( R(1) )</th>
<th>( \tilde{R}_v(1) ) (format long)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.0780906406</td>
<td>[0.0780, 0.0781]</td>
<td>0.078090640623532</td>
<td>[0.07809064062237, 0.07809064062470]</td>
</tr>
</tbody>
</table>

As we could observe in Cases 1 and 2, in the Case of study 3 (formed by components with Weibull distribution of failures), we also validate that the computed intervals for Weibull distribution enclose the real values obtained by the SHARPE software and the specified computational platform.

### 3.4 Case 4

Let \( V \) be the system outlined in Figure 5 compound by three subsystems: \( V_1 \) (formed uniquely by the component block0), \( V_2 \) (formed by the components block3 and block5 connected in parallel) and \( V_3 \) (formed by the components block6, block1, block2, block4, block7 in parallel). Consider that the component of \( V_1 \) has Normal distribution of failures with parameters \( \mu_0 = 10 \) and \( \sigma_0 = 2 \), the components of \( V_2 \) have Normal distribution of failures with parameters \( \mu_1 = 8 \) and \( \sigma_1 = 2 \), the components of \( V_3 \) have Normal distribution with parameters \( \mu_2 = 12 \) and \( \sigma_2 = 3 \). The real-valued reliability function of \( V \), considering observation time \( t = 20 \), was evaluated by the previously specified computational platform and compared with corresponding interval enclosure (Table 17).

The SHARPE software does not implement Normal failure distribution [24]. Therefore, the Case of study 4 only presents comparison between the real value computed by the specified computa-
Figure 5: Representation of the system $V$, modeled by the SHARPE software.

Table 17: Comparison between the real-valued reliability function of the system $V$ evaluated by the specified computational platform, $R(20)$, and the reliability enclosure, considering long configuration.

<table>
<thead>
<tr>
<th>$R(20)$</th>
<th>$6.95856414374359 \times 10^{-13}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_v(20)$ (format long)</td>
<td>$[0.69546030782011, 0.69625251864393] \times 10^{-12}$</td>
</tr>
</tbody>
</table>

Table 18 and 19 outline the main original contributions of this paper. Table 18 shows interval enclosures definitions for reliability metrics. Table 19 presents implementations’ signatures for interval enclosures, considering Exponential, Weibull and Normal distributions.

In Table 18, the interval proposed for reliability function is defined in [16]. The interval definitions proposed for mean time to failure and hazard rare function are original contributions of this work.

In this paper, all the intervals produced by the interval functions of Table 19 indeed bound the related real values. The quality of the results produced, i.e. interval width, can be improved increasing $p$ parameter. However, there is a tradeoff between width reduction and computation cost. The minimization of interval width only can be achieved with a bigger execution time.

Table 18: Summary of interval enclosures definitions for reliability metrics.

<table>
<thead>
<tr>
<th>Reliability metric</th>
<th>Interval definition proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reliability function</td>
<td>$R_v(t) = [1, 1] - P_v([0, t])$</td>
</tr>
<tr>
<td>Mean time to failure</td>
<td>$T_{medv} = \lim_{k \to +\infty} \frac{1}{n} \sum_{i=1}^{n} R_v(t_{ma})$</td>
</tr>
<tr>
<td>Hazard rate function</td>
<td>$\lambda_v(t) = \left[ \frac{fT_v(t)}{R_v(t)}, \frac{\Delta fT_v(t)}{R_v(t)} \right]$</td>
</tr>
</tbody>
</table>

Table 19: Summary of implementations’ signatures for interval enclosures, considering Exponential, Weibull and Normal distributions.

<table>
<thead>
<tr>
<th>Failure distribution</th>
<th>Reliability metric</th>
<th>IntLab implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exponential</td>
<td>Reliability function</td>
<td>confexp(t, p, α)</td>
</tr>
<tr>
<td></td>
<td>Mean time to failure</td>
<td>mttfIntervalExp(α, max, n, p)</td>
</tr>
<tr>
<td></td>
<td>Hazard rate function</td>
<td>failureRateIntervalExp(t, p, α)</td>
</tr>
<tr>
<td></td>
<td>Reliability function</td>
<td>confweibull(t, p, k, λ)</td>
</tr>
<tr>
<td></td>
<td>Mean time to failure</td>
<td>mttfIntervalWeibull(k, λ, max, n, p)</td>
</tr>
<tr>
<td></td>
<td>Hazard rate function</td>
<td>failureRateIntervalWeibull(t, p, k, λ)</td>
</tr>
<tr>
<td>Weibull</td>
<td>Reliability function</td>
<td>confnormal(t, p, μ, σ)</td>
</tr>
<tr>
<td></td>
<td>Mean time to failure</td>
<td>mttfIntervalNormal(μ, σ, max, n, p)</td>
</tr>
<tr>
<td></td>
<td>Hazard rate function</td>
<td>failureRateIntervalNormal(t, p, μ, σ)</td>
</tr>
</tbody>
</table>

In this context, the use of interval approach rather than employing punctual values brings an overhead computation cost that must be taken into account. For example, as shown in this work, reliability enclosure computation can spend 10 times more than the real-valued function. Finally, in the four study of cases presented, this paper successfully validates the reliability enclosure proposed with related real-valued reliability function of complex systems computed by SHARPE software and specified computational platform, considering Exponential, Weibull and Normal failure distributions.

RESUMO. A computação de métricas de confiabilidade (função de confiabilidade, tempo médio para falhas e taxa de falhas) envolve número reais. Portanto, problemas numéricos são gerados devido à limitação de representar e operar números reais em computadores. Esse artigo foca no cálculo de intervalos que limitam erros numéricos introduzidos durante o processo de computação de métricas de confiabilidade em máquinas digitais, considerando as distribuições de falhas Exponencial, Weibull e Normal. Funções intervalares foram propostas, baseadas na matemática intervalar e aritmética de exatidão máxima, para controlar erros numéricos introduzidos pelo cálculo de valores de métricas de confiabilidade para sistemas complexos. As funções intervalares, implementadas utilizando a biblioteca Intlab,
produzem intervalos encapsuladores para valores reais de métricas de confiabilidade e o software SHARPE foi usado para a validação dos resultados. A análise dos resultados numéricos obtidos com as funções propostas mostraram que os intervalos realmente encapsulam os números reais calculados pelo software SHARPE, indicando que essas funções, de fato, são uma alternativa para auto-validação desses valores de confiabilidade de sistemas complexos.

**Palavras-chave:** confiabilidade, matemática intervalar, aritmética de exatidão máxima, intervalos encapsuladores.

**REFERENCES**


A IMPLEMENTATIONS OF INTERVAL ENCLOSURES FOR EXPONENTIAL, WEIBULL AND NORMAL DISTRIBUTIONS

This appendix presents the implementation, coded in IntLab, of interval functions, for probabilities with Exponential, Weibull and Normal distributions. It is also provided the implementation of Interval Simpson Method and interval functions for reliability metrics for systems with Exponential, Weibull and Normal failure distributions.


164 INTERVAL ENCLOSURES FOR RELIABILITY METRICS

% Implementation of Simpson Interval Method
% (Equation 3 of the presented work).
% The arguments are:
% a = lower bound of the interval;
% b = upper bound of the interval;
% m = midpoint of the interval;
% d = width of the interval;
% x = interval that encapsulates the interval extension of the fourth order derivative.
function [re]=rsimpson(a,b,m,d,z)
y=(d/6)*(a+4*m+b)-(d^5)/2880*z;
re=y;

Figure 6: Interval Simpson Method.
% Implementation of interval function for
% probability values with Exponential distribution.
% The arguments are:
% \( x \) = lower bound of the interval;
% \( b \) = upper bound of the interval;
% \( \alpha \) = number of subdivisions of the interval \([a,b]\);
% \( \alpha \) = Exponential distribution parameter.

function [S]=expi(a,b,p,alpha)
if (b==inf) & (a<0)
    s = intval(1);
elseif (b==inf) & (a>0)
    sym s;
    yy = int(alpha*exp(-alpha*x));
    setround(-1);
    s1 = subs(yy,x,b) - subs(yy,x,a);
    setround(1);
    s2 = subs(yy,x,b) - subs(yy,x,a);
    setround(0);
    s = infsup(s1,s2);
else if ((a==inf) & (a<0)) & (b<0)
    s = infsup(0);
else
    lowerBound = 0;
    if(a<0)
        lowerBound = 0;
    else
        lowerBound = a;
    end
    sym s;
    yy = (alpha*exp(-alpha*x));
    y = diff(yy,4);
    h = (b-lowerBound)/p;
    n=1:p+1
    x(n) = a + h*(n-1);
    s = 0;
    for i=1:p
        t(i) = infsup( x(i) , x(i+1) );
        n(i)=end(t(i));
        f(i)=subs(yy,x(i));
        f(i+1)=subs(yy,x(i+1));
        fn(i)=subs(yy,x(n(i)));
        x1 = subs(y,x(i));
        x2 = subs(y,x(i+1));
        if (x1 <= x2)
            g(i) = infsup( x1 , x2 );
        else
            g(i) = infsup( x2 , x1 );
        end
        if (f(i) > f(i+1))
            e = e + reimpos(f(i+1),f(i),fn(i),h,g(i));
        end
    end
end

tend

Figure 7: Interval enclosure for Exponential distribution.

166 INTERVAL ENCLOSURES FOR RELIABILITY METRICS

Figure 8: Interval enclosure for Weibull distribution.

Figure 9: Interval enclosure for Normal distribution

```
function [S]=normal2(a,b,mu,sig)
    a = (a - mu)/sig;
    b = (b - mu)/sig;
    syms z;
    yy = (((1/sqrt(2*pi^1)))*(exp((-((z-0)^2)/(2*(1)^2)))));
    y = diff(yy,4);
    b = (b-a)/p;
    n=1:p+1
    x(n) = a + b*(n-1);
    s = 0;
    for i=1:p
        f(i) = infsup( x(i) , x(i+1) );
        m(i)=mid(t(i));
        f(i+1) = sub(y,z,x(i+1));
        f(i+1) = sub(y,z,x(i+1));
        x1 = sub(y,z,x(i+1));
        x2 = sub(y,z,x(i+1));
        if (x1 <= x2)
            g(i) = infsup( x1 , x2 );
        else
            g(i) = infsup( x2 , x1 );
        end
        if (f(i) >= f(i+1))
            s = s + rsimpson(f(i+1),f(i),m(i),h,g(i));
        else
            s = s + rsimpson(f(i),f(i+1),m(i),h,g(i));
        end
    end
    S = s;
end
```
Figure 10: Interval enclosure for reliability function considering component with Exponential failures distribution.

```matlab
function S = confexp(t, p, alpha)
    S = 1 - expl(0, t, p, alpha);
end
```

Figure 11: Interval enclosure for mean time to failure considering component with Exponential failures distribution.

```matlab
function S = stintervalExp(alpha, max, n, p)
    i = 0;
    h = max/sub;
    nArray = i:n+1;
    x(nArray) = h*(nArray-1) + h/2;
    temp = 0;
    t1 = [1, 1];
    for i = 1:n
        t1 = confexp(x(i), p, alpha);
        temp = temp + t1;
    end
    S = temp*h;
end
```

Figure 12: Interval enclosure for hazard rate function considering component with Exponential failures distribution.

Figure 13: Interval enclosure for reliability function considering component with Weibull failures distribution.

Figure 14: Interval enclosure for mean time to failure considering component with Weibull failures distribution.

```matlab
i = 0;
b = max/n;
narray=1:n+1;
x(narray) = b*(narray-1) + b/2;
temp = 0;
tl = [1,tl];
for i=1:n
  tl = confweibull(x(i),p,k,lambda);
  temp = temp + tl;
end
S = temp+b;
```

Figure 15: Interval enclosure for hazard rate function considering component with Weibull failures distribution.

```matlab
% Implementation of interval function for failure rate considering Weibull distribution.
% The arguments are:
% t = instant of time used for computing failure rate;
% p = number of subdivisions of the interval [0,t];
% k = shape parameter of the Weibull distribution;
% lamba = scale parameter of the Weibull distribution.
function [S] = failureRateIntervalWeibull(t,p,k,lambda)
syms x;
yy = (k*(1/lambda)*x^k)*exp((-((x/lambda)^k)));
setround(-1);
s1 = subs(yy,x,t);
setround(1);
s2 = subs(yy,x,t);
setround(0);
s = infsup(s1,s2);
ret = s/confweibull(t,p,k,lambda);
S = ret;
```
Figure 16: Interval enclosure for reliability function considering component with Normal failures distribution.

```
function S = confnormal(t, p, mu1, sig)
  S = 1 - normal(0, t, p, mu1, sig);
end
```

Figure 17: Interval enclosure for mean time to failure considering component with Normal failures distribution.

```
function S = mttfIntervalNormal(mu1, sig, max, n, p)
  i = 0;
  h = max/n;
  mArray = i:n+1;
  z(mArray) = h*(mArray-1) + h/2;
  temp = 0;
  t1 = [1,1];
  for i=1:n
    t1 = confnormal(z(i), p, mu1, sig);
    temp = temp + t1;
  end
  S = temp*h;
end
```
Figure 18: Interval enclosure for hazard rate function considering component with Normal failures distribution.