Tema

Tendências em Matemática Aplicada e Computacional, **17**, N. 2 (2016), 143-172
© 2016 Sociedade Brasileira de Matemática Aplicada e Computacional
www.scielo.br/tema
doi: 10.5540/tema.2016.017.02.0143

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 calcutation 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 give by: $X = [\underline{X}, \overline{X}] = \{x \in |\underline{X} \leq x \leq \overline{X}\}$.

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 $\mathbb{F} \subseteq \mathbb{R}$. Since $X = [\underline{X}, \overline{X}]$ are a high accuracy interval and encloses the real number x, then (i) $\underline{X} \in \mathbb{F}$, (ii) $\overline{X} \in \mathbb{F}$, (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 ∇ (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 $\triangle x \in \mathbb{F}$, $\triangle x \geq x$ and $\nabla x \in \mathbb{F}$, $\nabla x \leq x$.

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 rouding 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:

$$R(t) = P(T \geq t), \tag{2.1}$$

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

$$R(t) = \int_t^{+\infty} f_T(t)dt = 1 - \int_0^t f_T(t)dt. \tag{2.2}$$

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

$$S_i = \int_{a_{i-1}}^{a_i} f(t)dt = \frac{w(A_i)}{6}(F(a_{i-1}) + 4f(m(A_i)) + F(a_i)) - \frac{w(A_i)^5}{2880}G(\xi_i), \tag{2.3}$$

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 α is the parameter of the distribution and the values of a and b are the lower and higher integration bounds, respectively, and thus,

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

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 continously 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 λ are the shape and scale parameters of the Weibull distribution, respectively, while μ and σ 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,*

$$R_v(t) = [1, 1] - P_v([0, t]).$$

The Definition 2.1 supposes that systems must operate on the period $A = [0, t]$, where $t > 0$. However, the domain of the Normal distribution is the set of all real numbers, including negative values. Thus, Definition 2.1 only can be applied to Normal distribution in the case of negligible probabilities for negative values.

We know that $X * Y = \{x * y | x \in X, y \in Y\}, * \in \{+, -, \cdot, /\}$, where $0 \notin Y$, when $*$ is $/$. Since $x \in X$ and $y \in Y$, it follows that the real value $x * y$ also belong to the interval $X * Y$. Therefore,

$$x * y \in X * Y. \tag{2.4}$$

where $0 \notin Y$, when $*$ is $/$.

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

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, $R(t) \in R_v(t)$.*

Proof. $1 \in [1, 1]$ and $P(0 < T < t) \in P_v([0, t])$ (Equation 8.7 of [3]). Then $1 - P(0 < T < t) \in [1, 1] - P_v([0, t]) \Rightarrow R(t) \in R_v([0, t])$. □

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 1: Signatures of reliability enclosure.

Distribution	Implementation
Exponential	$confexp(t, p, \alpha) = [1, 1] - exp1(0, t, p, \alpha)$
Weibull	$confweibull(t, p, k, \lambda) = [1, 1] - weibull1(0, t, p, k, \lambda)$
Normal	$confnormal(t, p, \mu, \sigma) = [1, 1] - normal1(0, t, p, \mu, \sigma)$

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

$$\begin{aligned} R_v(8000) &= [1, 1] - P_v([0, 8000]) \\ &= confWeibull(8000, 100, 2, 10000) \\ &= [1, 1] - weibull1(0, 8000, 100, 2, 10000) \\ &= [0.52729242404274, 0.52729242404335]. \end{aligned}$$

The Table 2 exibiths 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.

p	$R_v(8000)$	$w(R_v(8000))$	Execution time(s)
5	$[0.52729134642417, 0.52729322980637]$	$< 2.0 \cdot 10^{-6}$	0.888380
10	$[0.52729239237141, 0.52729245148924]$	$< 6.0 \cdot 10^{-8}$	1.737012
50	$[0.52729242403336, 0.52729242405247]$	$< 2.0 \cdot 10^{-11}$	8.599330
100	$[0.52729242404274, 0.52729242404335]$	$< 6.1 \cdot 10^{-13}$	17.875280
200	$[0.52729242404303, 0.52729242404307]$	$< 4.0 \cdot 10^{-14}$	35.186983

Similarly, we bring another two examples with Exponential and Normal distribution of failures. The Table 3 shows the results in the reliability enclousures 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 $3.354626279025164 \cdot 10^{-4}$ and 0.050303 seconds. The Table 4 exibiths data from the reliability enclousures 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.

p	$R_v(2)$	$w(R_v(2))$	Execution time(s)
5	$[-0.085621563302, 0.278345187372] \cdot 10^{-2}$	$< 3.7 \cdot 10^{-3}$	0.423623
10	$[0.28875911756531, 0.40249872715171] \cdot 10^{-3}$	$< 1.2 \cdot 10^{-4}$	0.771025
50	$[0.33544508784544, 0.33548148452612] \cdot 10^{-3}$	$< 3.7 \cdot 10^{-8}$	3.624166
100	$[0.33546206948853, 0.33546320689571] \cdot 10^{-3}$	$< 1.2 \cdot 10^{-9}$	7.292063
200	$[0.33546261028183, 0.33546264584506] \cdot 10^{-3}$	$< 3.6 \cdot 10^{-11}$	14.282848

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.

p	$R_v(80)$	$w(R_v(80))$	Execution time(s)
5	$[0.99992849259341, 1.00247004587546]$	$< 2.6 \cdot 10^{-3}$	0.750204
10	$[0.99996023275219, 1.00003965629226]$	$< 8.0 \cdot 10^{-5}$	1.338139
50	$[0.99996831854230, 0.99996834395784]$	$< 2.6 \cdot 10^{-8}$	6.396252
100	$[0.99996832840014, 0.99996832919439]$	$< 8.0 \cdot 10^{-10}$	14.061650
200	$[0.99996832874636, 0.99996832877119]$	$< 2.5 \cdot 10^{-11}$	30.258739

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

$$T_{med} = \int_0^{+\infty} t f_T(t) dt. \tag{2.5}$$

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]. □

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:

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

where $\xi_1 \in [x_0, x_1]$, $\xi_2 \in [x_1, x_2]$, ..., $\xi_n \in [x_{n-1}, x_n]$. The intervals $[x_{i-1}, x_i]$, $i = 1, 2, \ldots, n$, 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 $w([x_{i-1}, x_i])$ 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., $T_{med} < +\infty$. □

Based on Equation 2.6, this paper defines the interval function which yields an interval enclosure for T_{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_{MAX} and a positive integer n, where $n \to \infty$. Suppose the interval $[0, t_{MAX}]$ and its partitions $[x_0, x_1]$, $[x_1, x_2]$, $\ldots$, $[x_{n-1}, x_n]$, where $x_0 = 0$ and $x_n = t_{MAX}$. Also consider that all partitions have the same width* $\mathbf{w}= \frac{t_{MAX}}{n}$. *So, the interval function that encloses the Mean Time To Failure, considering T, is defined as*

$$T_{med_v} = \lim_{n\to+\infty} [R_v(t_{m_1}) + R_v(t_{m_2}) + \cdots + R_v(t_{m_n})]\mathbf{w}. \tag{2.7}$$

The values $t_{m1}, t_{m_2}, \ldots, t_{m_n}$ 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_{MAX} and n. So, we must choose the value of t_{MAX} in such way that $R(t_{MAX}) \approx 0$. The value of n represents the number of partitions of the interval $[0, t_{MAX}]$, 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 5: Signatures of interval function for mean time to failure.

Distribution	Implementation
Exponential	$mttfIntervalExp(\alpha, t_{MAX}, n, p)$
Weibull	$mttfIntervalWeibull(k, \lambda, t_{MAX}, n, p)$
Normal	$mttfIntervalNormal(\mu, \sigma, t_{MAX}, n, p)$

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_{med_v} 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 $t_{MAX} = 16$, 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 $T_{med} = \mu = 8$ 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_{med} is given by

$$\begin{aligned} T_{med_v} &= mttfIntervalNormal(8, 2, 16, 50, 5) \\ &= [7.98389618066430, 8.01618738659977]. \end{aligned}$$

In fact, we have that $T_{med} \in T_{med_v}$. However, it can be seen that when the condition $R(t_{MAX}) \approx 0$ 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.

t_{MAX}	T_{med_v}	$R(t_{MAX})$
16	[7.98388828158800, 8.01618934801887]	$6.6613 \cdot 10^{-16}$
14	[7.99026276808957, 8.00936514850080]	$9.8659 \cdot 10^{-10}$
12	[7.98065312859909, 7.98618574479369]	$3.1671 \cdot 10^{-5}$
10	[7.83278842107344, 7.83448378185583]	0.0227

Once the computation of T_{med_v} involves n evaluations of R_v, the parameter p is also used to enclose 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.

p	T_{med_v}	$w(T_{med_v})$	Execution Time(s)
5	[7.95139633268973, 8.06968007167273]	$< 1.2 \cdot 10^{-1}$	46.126105
6	[7.97723844794519, 8.02123846433367]	$< 4.5 \cdot 10^{-2}$	53.791730
7	[7.98380610504386, 8.01751702986318]	$< 3.4 \cdot 10^{-2}$	62.819581
8	[7.99300780057101, 8.00823643857211]	$< 1.6 \cdot 10^{-2}$	70.775011
9	[7.99595743932975, 8.00531274894078]	$< 9.4 \cdot 10^{-3}$	80.984591
10	[7.99776142904487, 8.00352002468972]	$< 5.8 \cdot 10^{-3}$	89.759203

We also bring two examples with Exponential and Weibull distribution of failures. The Table 8 exibiths the results in the mean time to failure enclousure 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 enclousure 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.

p	T_{med_v}	$w(T_{med_v})$	Execution Time(s)
5	[99.390693383917,101.241198466660]	$< 1.9 \cdot 10^{0}$	63.858197
6	[99.720562136152, 100.464238625810]	$< 7.5 \cdot 10^{-1}$	76.539810
7	[99.851172010656, 100.195244622283]	$< 3.5 \cdot 10^{-1}$	87.797895
8	[99.909610963321, 100.086088880303]	$< 1.8 \cdot 10^{-1}$	98.893686
9	[99.938339164273, 100.036271870733]	$< 1.0 \cdot 10^{-1}$	110.999310
10	[99.953555520054, 100.011383803893]	$< 6.0 \cdot 10^{-2}$	122.218566

Table 9: Interval enclosures for mean time to failure, since p increases, considering $t_{MAX} = 120$ and $n = 150$, for Weibull failure distribution system.

p	T_{med_v}	$w(T_{med_v})$	Execution Time(s)
5	[44.58350915710464, 44.72000067070699]	$< 1.4 \cdot 10^{-1}$	206.545242
6	[44.62385044500201, 44.67604159505329]	$< 5.3 \cdot 10^{-2}$	236.261671
7	[44.63623868424217, 44.66247773340652]	$< 2.7 \cdot 10^{-2}$	253.500081
8	[44.64224234478933, 44.65604948443448]	$< 1.4 \cdot 10^{-2}$	286.737339
9	[44.64532405850133, 44.65279245624996]	$< 8.0 \cdot 10^{-3}$	323.563472
10	[44.64669686045618, 44.65133965553606]	$< 5.0 \cdot 10^{-3}$	366.583825

As it is shown in Tables 7, 8 and 9, the interval enclosures for mean time to failure bound corrresponding 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_v 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 $\lambda(t)$ [5] is based on the conditional probability:

$$P(t \leq T \leq t + \triangle t | T \geq t) = \frac{R(t) - R(t + \triangle t)}{R(t)}, \tag{2.8}$$

where $R(t) > 0$.

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

$$\frac{R(t) - R(t + \triangle t)}{R(t)\triangle t}. \tag{2.9}$$

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 occurences per time unit, when $\triangle t \to 0$. Then,

$$\begin{aligned} \lambda(t) &= \lim_{\triangle t \to 0} \frac{-[R(t + \triangle t) - R(t)]}{\triangle t} \cdot \frac{1}{R(t)} \\ &= \frac{-dR(t)}{dt} \cdot \frac{1}{R(t)} = \frac{f_T(t)}{R(t)}. \end{aligned} \tag{2.10}$$

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:*

$$\lambda_v(t) = \frac{[\nabla f_T(t), \triangle f_T(t)]}{R_v(t)},$$

where $0 \notin R_v(t)$.

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

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

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

when $0 \notin R_v(t)$. □

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

Table 10: Signatures of interval function for hazard rate function.

Distribution	Implementation
Exponential	$failureRateIntervalExp(t, p, \alpha)$
Weibull	$failureRateIntervalWeibull(t, p, k, \lambda)$
Normal	$failureRateIntervalNormal(t, p, \mu, \sigma)$

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 11: Interval enclosures for hazard rate, considering p variation, for Exponential failure distribution system.

p	T_{med_v}	$w(T_{med_v})$	Execution Time(s)
5	[0.00999999900224, 0.01000000091145]	$< 2.0 \cdot 10^{-9}$	0.491079
10	[0.00999999996949, 0.01000000002916]	$< 6.0 \cdot 10^{-11}$	0.942291
50	[0.00999999999999, 0.01000000000001]	$< 2.0 \cdot 10^{-14}$	3.970694
100	[0.00999999999999, 0.01000000000001]	$< 2.0 \cdot 10^{-14}$	8.780558
200	[0.00999999999999, 0.01000000000001]	$< 2.0 \cdot 10^{-14}$	16.230988

Similarly, we bring another two examples with Normal and Weibull distribution of failures. The Table 12 shows the results in the hazard rate enclousures 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 exibiths data from the hazard rate enclousures 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.

p	T_{med_v}	$w(T_{med_v})$	Execution Time(s)
5	[1.51863640703179, 1.53229224115192]	$< 1.4 \cdot 10^{-2}$	1.008209
10	[1.52460379156082, 1.52507613749420]	$< 4.8 \cdot 10^{-4}$	2.071646
50	[1.52483080854381, 1.52483096220953]	$< 1.6 \cdot 10^{-7}$	11.009327
100	[1.52483088240966, 1.52483088722032]	$< 4.9 \cdot 10^{-9}$	22.419977
200	[1.52483088473101, 1.52483088488142]	$< 1.6 \cdot 10^{-10}$	52.203656

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

p	T_{med_v}	$w(T_{med_v})$	Execution Time(s)
5	$[0.15999975550201, 0.16000032699005] \cdot 10^{-3}$	$< 5.8 \cdot 10^{-10}$	1.216593
10	$[0.15999999167181, 0.16000000961035] \cdot 10^{-3}$	$< 1.8 \cdot 10^{-11}$	2.054348
50	$[0.15999999999714, 0.16000000000294] \cdot 10^{-3}$	$< 5.8 \cdot 10^{-15}$	9.105660
100	$[0.15999999999990, 0.16000000000010] \cdot 10^{-3}$	$< 2.1 \cdot 10^{-16}$	17.899182
200	$[0.15999999999999, 0.16000000000001] \cdot 10^{-3}$	$< 2.0 \cdot 10^{-17}$	35.608336

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 $\mathbf{x} = (\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n})$ 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 ϕ, is a function of the vector $\mathbf{x}$. Then, $\phi = \phi(\mathbf{x}) = \phi(\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n})$.

Thus, the function ϕ is determined by the components' configuration of a given system. The following configurations are addressed in this section: **series** and **parallel**.

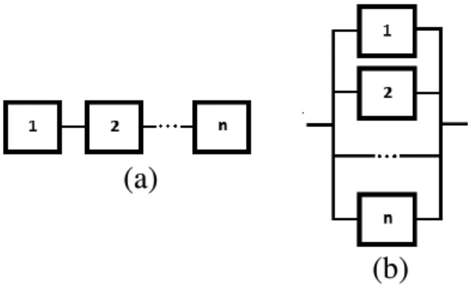


Figure 1: (a) Series configuration and (b) Parallel configuration.

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 ϕ that defines the state of L is

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

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

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

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 ϕ that defines the state of L is

$$\phi(\mathbf{x}) = 1 - \prod_{i=1}^{n}(1 - x_i) = \max(x_1, x_2, \ldots, x_n). \tag{3.3}$$

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

$$R(t) = 1 - \prod_{i=1}^{n}[1 - R_{(i)}(t)]. \tag{3.4}$$

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 $\alpha_0 = 5 \cdot 10^{-12}$ 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).

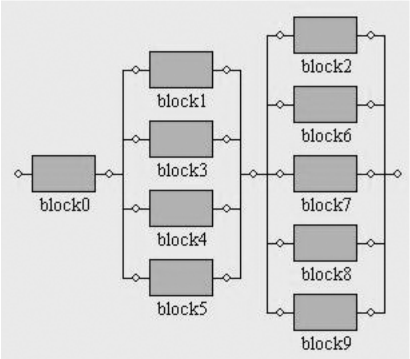


Figure 2: Representation of the system W, modeled by the SHARPE software.

Table 14: Comparison between the real-valued reliability function of the system W evaluated by SHARPE, $\widetilde{R}(20)$, and the specified computational platform, $R(20)$, and the reliability enclosure, considering short and long configurations.

$\widetilde{R}(20)$	1.00000000
$R_v(20)$ (format single)	[0.9999, 1.0000]
$R(20)$	0.999999999900000
$R_v(20)$ (format long)	$[0.99999999989999, 0.99999999990001]$

We can observe that the interval enclosure with format single configuration contains the value of $\widetilde{R}(20)$. 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 forbidden 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

(formed by the components block10 and block12 connected in series). Consider that the components of M_1 and M_5 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).

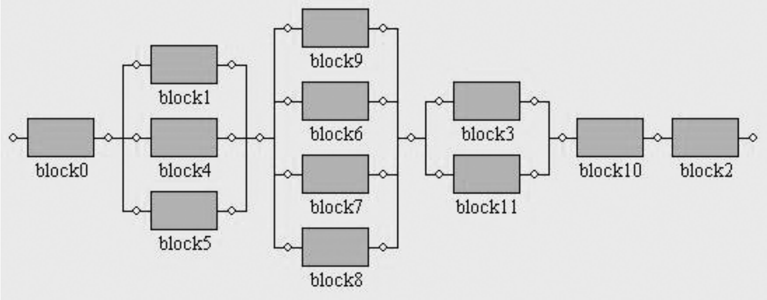


Figure 3: Representation of the system M, modeled by the SHARPE software.

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

$\widetilde{R}(20)$	0.00000000
$R_v(20)$ (format single)	$[-0.3364, 0.3615] \cdot 10^{-29}$
$R(20)$	$2.169808968150444 \cdot 10^{-36}$
$R_v(20)$ (format long)	$[-0.33637728687207, 0.36140951783338] \cdot 10^{-29}$

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).

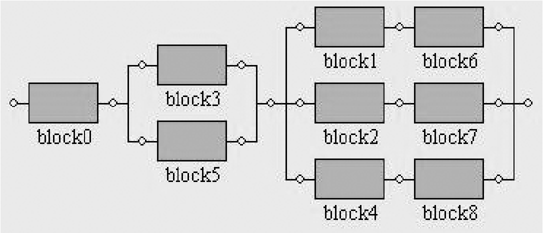


Figure 4: Representation of the system G, modeled by the SHARPE software.

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

$\widetilde{R}(1)$	0.0780906406
$R_v(1)$ (format single)	$[0.0780, 0.0781]$
$R(1)$	0.078090640623532
$R_v(1)$ (format long)	$[0.07809064062237, 0.07809064062470]$

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-

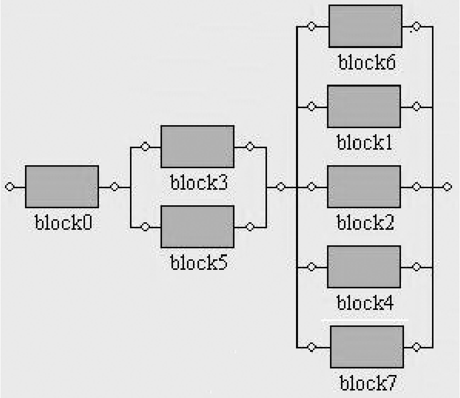


Figure 5: Representation of the system V, modeled by the SHARPE software.

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.

$R(20)$	$6.958564143474359 \cdot 10^{-13}$
$R_v(20)$ (format long)	$[0.69546030782011, 0.69625251864393] \cdot 10^{-12}$

tional platform and the reliability enclosure with long configuration. As could be observed in the other cases, the interval enclosure obtained for the complex system bounds the related punctual value.

4 CONCLUSIONS

This paper presented a wide set of interval enclosures definitions for reliability metrics, that controls round-off and truncation errors produced by the computation of the corresponding real values. The intervals are obtained using high accuracy in the floating-point system employed.

In order to summary this article, the Tables 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.

Reliability metric	Interval definition proposed
Reliability function	$R_v(t) = [1, 1] - P_v([0, t])$
Mean time to failure	$T_{med_v} = \lim\limits_{k \to +\infty} \sum\limits_{n=1}^{k} R_v(t_{m_n})\mathbf{w}$
Hazard rate function	$\lambda_v(t) = \frac{[\nabla f_T(t), \triangle f_T(t)]}{R_v(t)}$

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

Failure distribution	Reliability metric	IntLab implementation
Exponential	Reliability function	$confexp(t, p, \alpha)$
	Mean time to failure	$mttfIntervalExp(\alpha, \text{max}, n, p)$
	Hazard rate function	$failureRateIntervalExp(t, p, \alpha)$
Weibull	Reliability function	$confweibull(t, p, k, \lambda)$
	Mean time to failure	$mttfIntervalWeibull(k, \lambda, \text{max}, n, p)$
	Hazard rate function	$failureRateIntervalWeibull(t, p, k, \lambda)$
Normal	Reliability function	$confnormal(t, p, \mu, \sigma)$
	Mean time to failure	$mttfIntervalNormal(\mu, \sigma, \text{max}, n, p)$
	Hazard rate function	$failureRateIntervalNormal(t, p, \mu, \sigma)$

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 succesfully 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 numericos 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

[1] M.A. Campos. Interval probability: applications to discrete random variables. *TEMA – Trends in Applied and Computational Mathematics*, **1**(2) (2000), 333–343.

[2] M.A. Campos & M.G. Santos. Interval Probabilities and Enclosures. *Computational & Applied Mathematics*, **32**(1) (2013), 413–423.

[3] O. Caprani, K. Madsen & H.B. Nielsen. "Introduction to Interval Analysis". Technical University of Denmark, Copenhagen (2002).

[4] F.P.A. Coolen & M.J. Newby. "Bayesian Reliability Analysis with Imprecise Prior Probabilities". Eindhoven University of Technology, Eindhoven (1992).

[5] C.E. Ebeling. "An Introduction to Reliability and Maintainability Engineering". Waveland Press, Illinois (1997).

[6] B.V. Goldberg. What every computer scientist should know about floating-point arithmetic. *ACM Computing Surveys*, **23**(1) (1991), 153–230.

[7] P.S. Grigoletti, G.P. Dimuro & L.V. Barboza. Módulo python para matemática intervalar. *TEMA – Trends in Applied and Computational Mathematics*, **8**(1) (2007), 73–82.

[8] N.J. Higham. Accuracy and stability of numerical algorithms, 2nd edn. SIAM Publications, Philadelphia (2002).

[9] C. Hirel, X. Sahner, X. Zang & K. Trivedi. "Reliability and Performability Modeling using SHARPE 2000" (2011). (avaliable in: <http://people.ee.duke.edu/∼kst/>.)

[10] IEEE Standard 754-2008: IEEE Standard for Floating-Point Arithmetic. IEEE Computer Society, New York (2008).

[11] R. Klatte, U. Kulisch, C. Lawo, M. Rauch & A. Wietho. "C-XSC - A C++ class library for extended scientific computing". Springer, Heidelberg (1993).

[12] U.W. Kulisch & W.L. Miranker. "Computer Arithmetic in Theory and Practice", Academic Press, New York (1981).

[13] W. Kuo & M.J. Zuo. "Optimal Reliability Modeling: Principles and Applications". John Wiley & Sons Inc, New Jersey (2003).

[14] G. Levitin, L. Xing, H. Ben-Haim & Y. Dai. Reliability of series-parallel systems with random failure propagation time. *IEEE Transactions on Reliability*, **PP**, Issue: 99, 1–11 (2013).

[15] The MathWorks Inc, "MATLAB 7.5" (2007). (available in: `http://www.mathworks.com/products/matlab/`.)

[16] A.F. Mendonça & M.A. Campos. Confiabilidade Autovalidável de Sistemas com Processo Exponencial de Falhas. *TEMA – Trends in Applied and Computational Mathematics*, **14**(3) (2013), 383–398.

[17] P.L. Meyer. " Probabilidade Aplicações à Estatística". Livros Técnicos e Científicos, Rio de Janeiro (1983).

[18] R.E. Moore, W. Strother & C.T. Yang. "Interval Integrals". Technical Report Space Div. Report LMSD703073, Lockheed Missiles and Space Co., (1960).

[19] R.E. Moore. "Interval Analysis". Englewood Cliffs, New Jersey (1966).

[20] R.E. Moore. "Methods and Applications of Interval Analysis". Society for Industrial and Applied Mathematics Philadelphia, Philadelphia, (1979).

[21] R.E. Moore, R. B. Kearfott & M.J. Cloud. "Introduction to Interval Analysis". Society for Industrial and Applied Mathematics Philadelphia, Philadelphia, (2009).

[22] S.M. Rump: INTLAB – INTerval LABoratory. In Tibor Csendes, editor, Developments in Reliable Computing, pages 77-104. Kluwer Academic Publishers, Dordrecht (1999).

[23] R. Sahner, K.S. Trivedi & A. Puliafito. "Performance and Reliability of Computer Systems: An Example-Based Approach Using the SHARPE Software Package". Kluwer Academic Publishers, Boston (1996).

[24] R. Sahner & K. Trivedi. Reliability Modeling Using SHARPE. *Reliability, IEEE Transactions*, **36**(2) (1987), 186–193.

[25] M. Spivak. "Calculus". Publish or Perish, Houston (1994).

[26] T. Sunaga. Theory of An Interval Algebra and its Application to Numerical Analysis. *RAAG Memoirs*, **2** (1958), 29–46.

[27] L.V. Utkin. Imprecise reliability of cold standby systems. *International Journal of Quality & Reliability Management*, **20** (2003), 722–739.

[28] L.V. Utkin. Interval reliability of typical systems with partially known probabilities. *European Journal of Operational Research*, **153**(3) (2004), 790–802.

[29] Y. Wang. Imprecise probabilities based on generalized intervals for system reliability assessment. *International Journal of Reliability & Safety*, **1** (2009), 1–23.

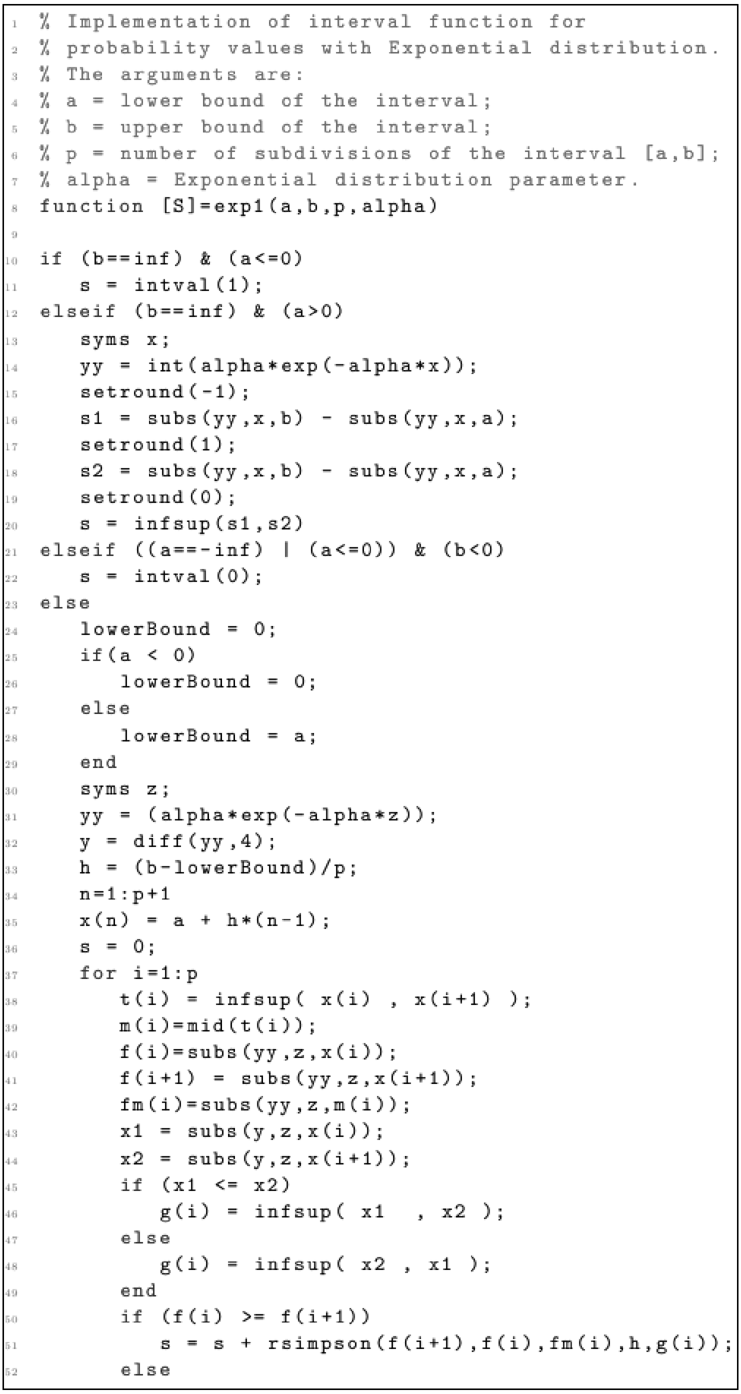
[30] J. Yang & H. Sun. Discrete method for structural interval reliability analysis. *Chinese Control and Decision Conference*, **1** (2008), 2441–2446.

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.

```
% 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;
% z = interval that encapsulates the interval extension of the fourth order
% derivative.
function [rs]=rsimpson(a,b,m,d,z)
y=(d/6)*(a + 4*m + b)-((d^5)/2880)*z;
rs=y;
```

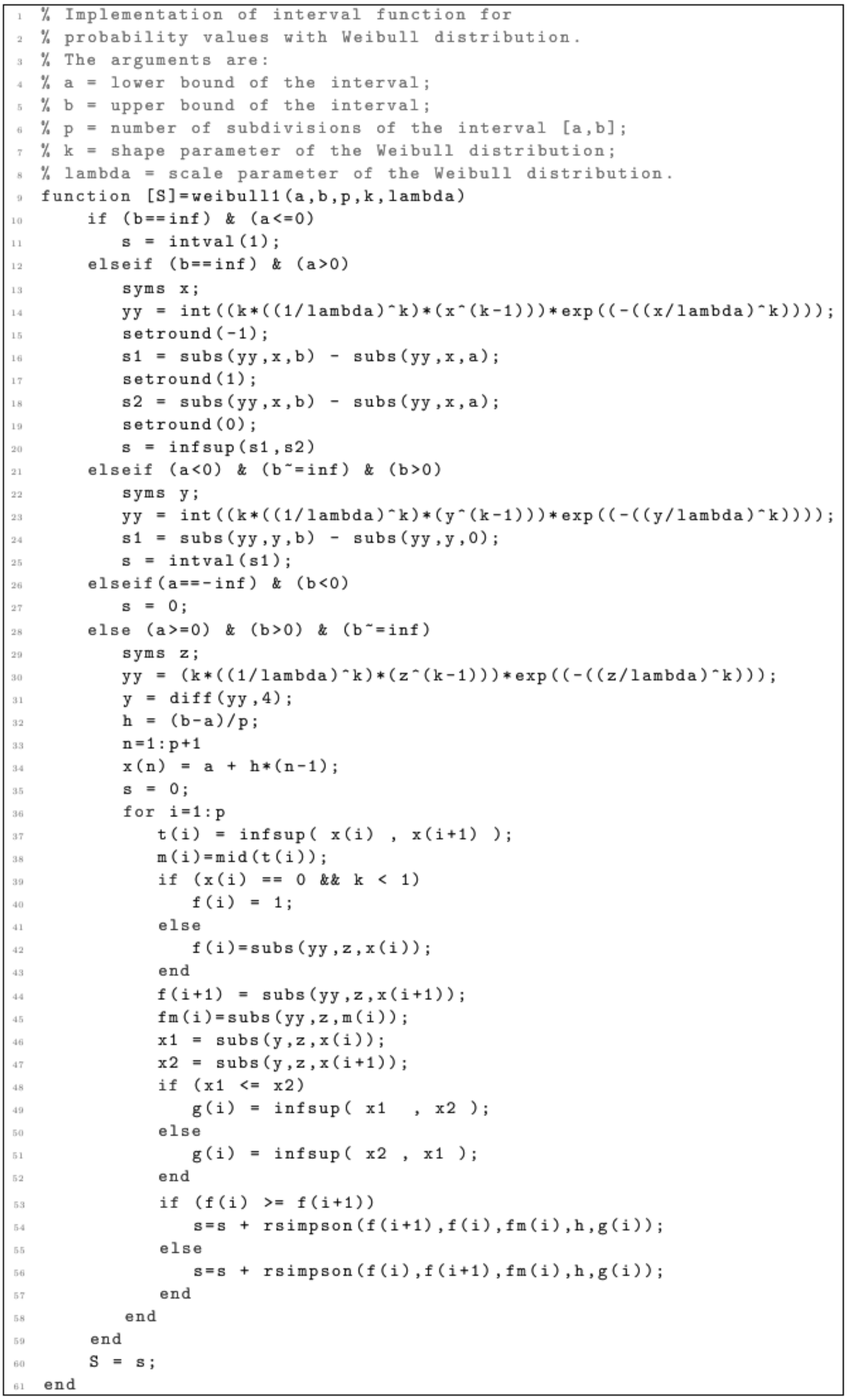
Figure 6: Interval Simpson Method.



```
% Implementation of interval function for
% probability values with Exponential distribution.
% The arguments are:
% a = lower bound of the interval;
% b = upper bound of the interval;
% p = number of subdivisions of the interval [a,b];
% alpha = Exponential distribution parameter.
function [S]=exp1(a,b,p,alpha)

if (b==inf) & (a<=0)
   s = intval(1);
elseif (b==inf) & (a>0)
   syms x;
   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)
elseif ((a==-inf) | (a<=0)) & (b<0)
   s = intval(0);
else
   lowerBound = 0;
   if(a < 0)
      lowerBound = 0;
   else
      lowerBound = a;
   end
   syms z;
   yy = (alpha*exp(-alpha*z));
   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) );
      m(i)=mid(t(i));
      f(i)=subs(yy,z,x(i));
      f(i+1) = subs(yy,z,x(i+1));
      fm(i)=subs(yy,z,m(i));
      x1 = subs(y,z,x(i));
      x2 = subs(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),fm(i),h,g(i));
      else
```

Figure 7: Interval enclosure for Exponential distribution.

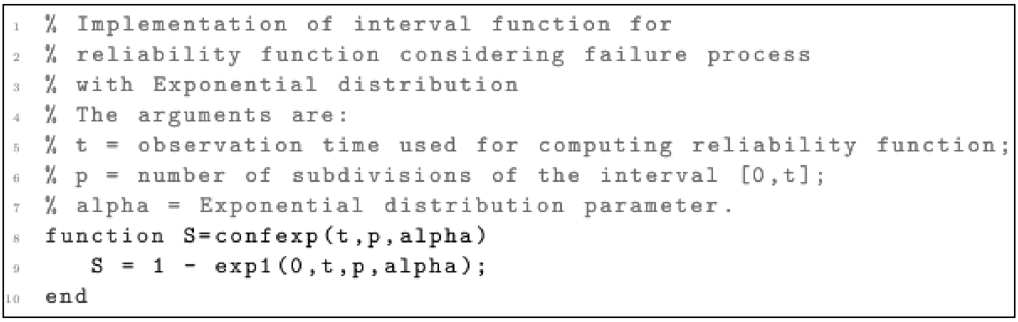


```
% Implementation of interval function for
% probability values with Weibull distribution.
% The arguments are:
% a = lower bound of the interval;
% b = upper bound of the interval;
% p = number of subdivisions of the interval [a,b];
% k = shape parameter of the Weibull distribution;
% lambda = scale parameter of the Weibull distribution.
function [S]=weibull1(a,b,p,k,lambda)
    if (b==inf) & (a<=0)
       s = intval(1);
    elseif (b==inf) & (a>0)
       syms x;
       yy = int((k*((1/lambda)^k)*(x^(k-1)))*exp((-((x/lambda)^k))));
       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)
    elseif (a<0) & (b~=inf) & (b>0)
       syms y;
       yy = int((k*((1/lambda)^k)*(y^(k-1)))*exp((-((y/lambda)^k))));
       s1 = subs(yy,y,b) - subs(yy,y,0);
       s = intval(s1);
    elseif(a==-inf) & (b<0)
       s = 0;
    else (a>=0) & (b>0) & (b~=inf)
       syms z;
       yy = (k*((1/lambda)^k)*(z^(k-1)))*exp((-((z/lambda)^k)));
       y = diff(yy,4);
       h = (b-a)/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) );
          m(i)=mid(t(i));
          if (x(i) == 0 && k < 1)
             f(i) = 1;
          else
             f(i)=subs(yy,z,x(i));
          end
          f(i+1) = subs(yy,z,x(i+1));
          fm(i)=subs(yy,z,m(i));
          x1 = subs(y,z,x(i));
          x2 = subs(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),fm(i),h,g(i));
          else
             s=s + rsimpson(f(i),f(i+1),fm(i),h,g(i));
          end
       end
    end
    S = s;
end
```

Figure 8: Interval enclosure for Weibull distribution.

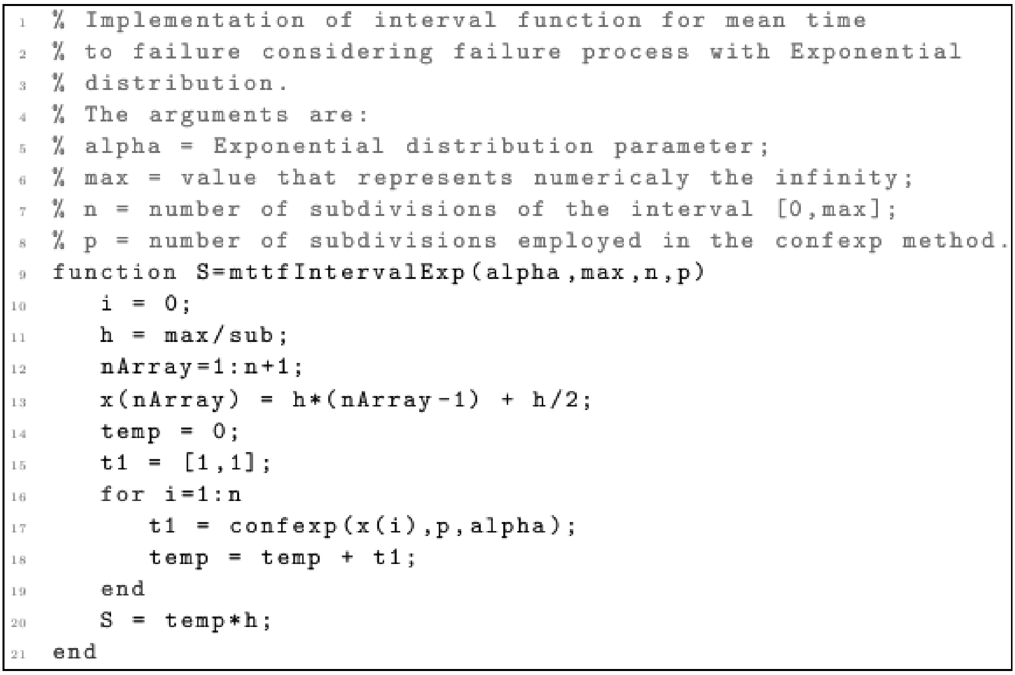
```
% Implementation of interval function for
% probability values with Normal distribution.
% The arguments are:
% a = lower bound of the interval;
% b = upper bound of the interval;
% p = number of subdivisions of the interval [a,b];
% mu1 = mean parameter of the Normal distribution;
% sig = standard deviation parameter of the Normal distribution.
function [S]=normal1(a,b,p,mu1,sig)
   a = (a - mu1)/sig;
   b = (b - mu1)/sig;
   syms z;
   yy = (((1/sqrt(2*pi*1))*(exp(-((z-0)^2/(2*(1)^2)))));
   y = diff(yy,4);
   h = (b-a)/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) );
      m(i)=mid(t(i));
      f(i)=subs(yy,z,x(i));
      f(i+1) = subs(yy,z,x(i+1));
      fm(i)=subs(yy,z,m(i));
      x1 = subs(y,z,x(i));
      x2 = subs(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),fm(i),h,g(i));
      else
         s=s + rsimpson(f(i),f(i+1),fm(i),h,g(i));
      end
   end
   S = s;
end
```

Figure 9: Interval enclosure for Normal distribution



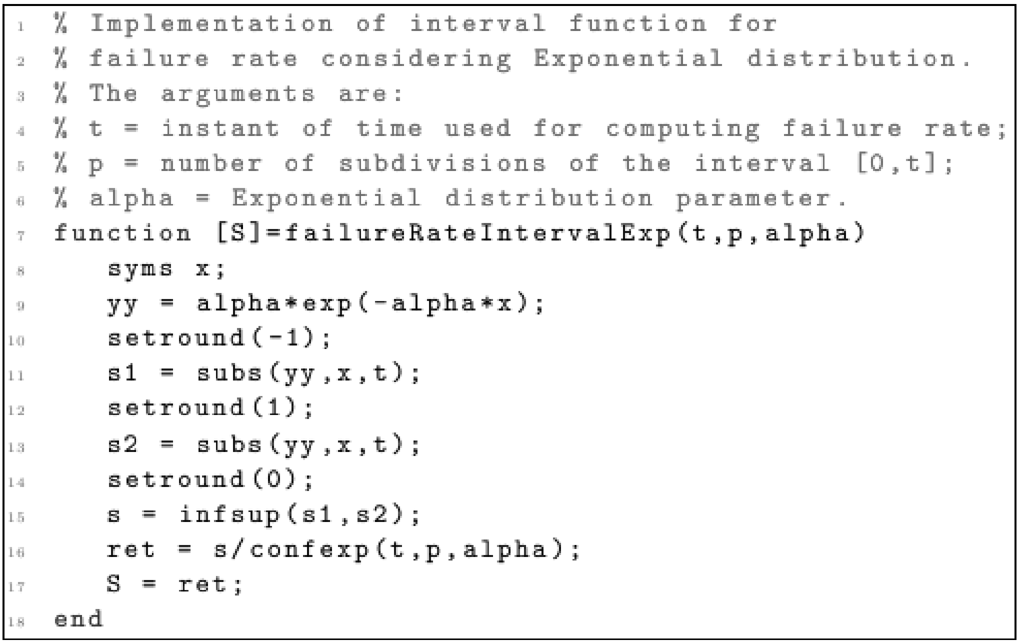
```
% Implementation of interval function for
% reliability function considering failure process
% with Exponential distribution
% The arguments are:
% t = observation time used for computing reliability function;
% p = number of subdivisions of the interval [0,t];
% alpha = Exponential distribution parameter.
function S=confexp(t,p,alpha)
   S = 1 - exp1(0,t,p,alpha);
end
```

Figure 10: Interval enclosure for reliability function considering component with Exponential failures distribution.



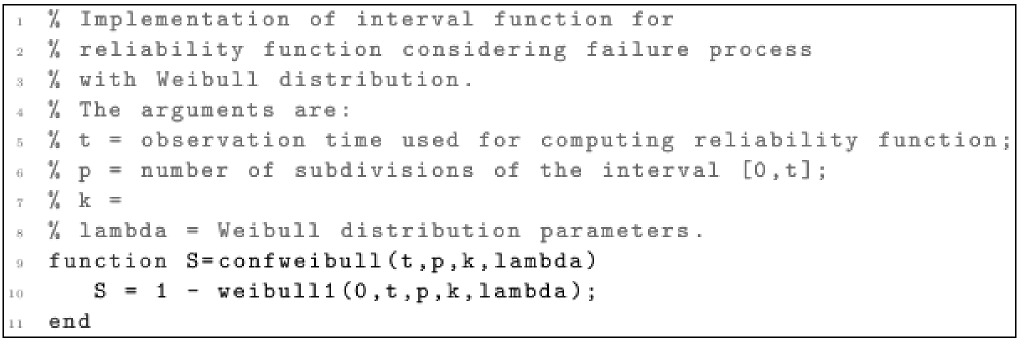
```
% Implementation of interval function for mean time
% to failure considering failure process with Exponential
% distribution.
% The arguments are:
% alpha = Exponential distribution parameter;
% max = value that represents numericaly the infinity;
% n = number of subdivisions of the interval [0,max];
% p = number of subdivisions employed in the confexp method.
function S=mttfIntervalExp(alpha,max,n,p)
    i = 0;
    h = max/sub;
    nArray=1: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 11: Interval enclosure for mean time to failure considering component with Exponential failures distribution.



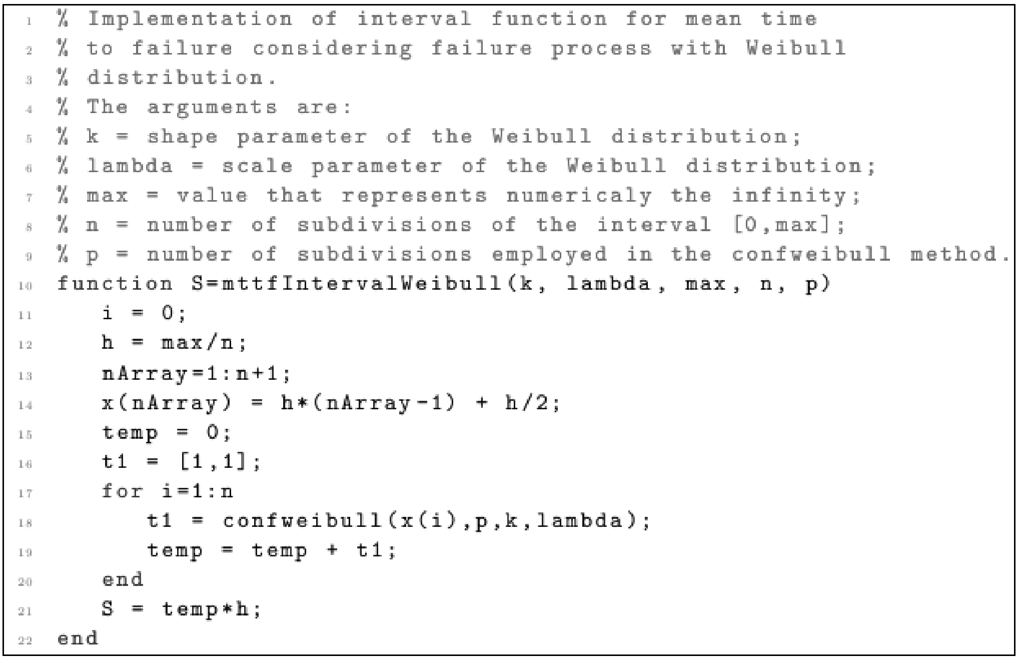
```
% Implementation of interval function for
% failure rate considering Exponential distribution.
% The arguments are:
% t = instant of time used for computing failure rate;
% p = number of subdivisions of the interval [0,t];
% alpha = Exponential distribution parameter.
function [S]=failureRateIntervalExp(t,p,alpha)
    syms x;
    yy = alpha*exp(-alpha*x);
    setround(-1);
    s1 = subs(yy,x,t);
    setround(1);
    s2 = subs(yy,x,t);
    setround(0);
    s = infsup(s1,s2);
    ret = s/confexp(t,p,alpha);
    S = ret;
end
```

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



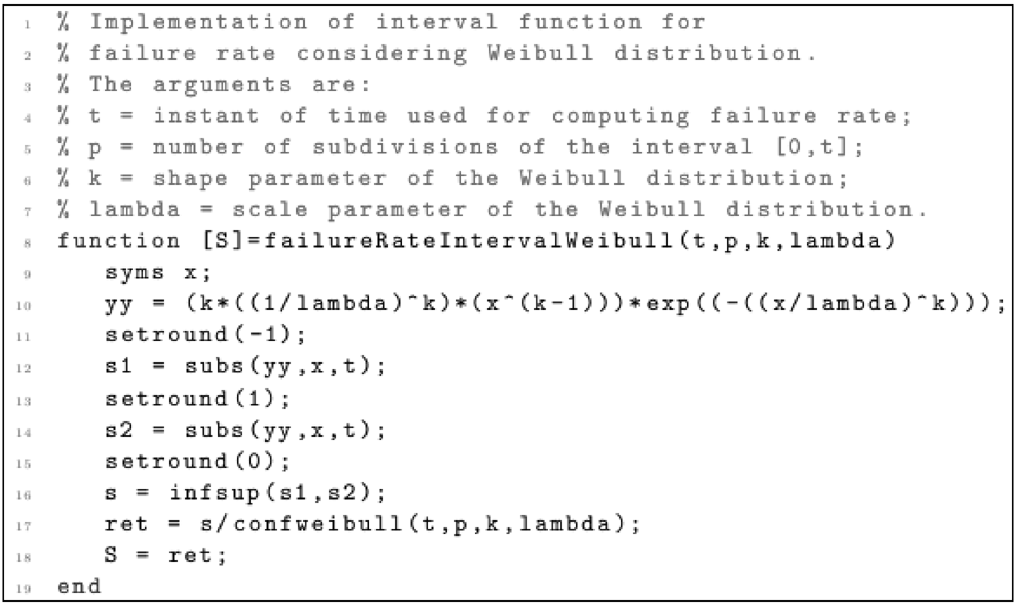
```
% Implementation of interval function for
% reliability function considering failure process
% with Weibull distribution.
% The arguments are:
% t = observation time used for computing reliability function;
% p = number of subdivisions of the interval [0,t];
% k =
% lambda = Weibull distribution parameters.
function S=confweibull(t,p,k,lambda)
    S = 1 - weibull1(0,t,p,k,lambda);
end
```

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



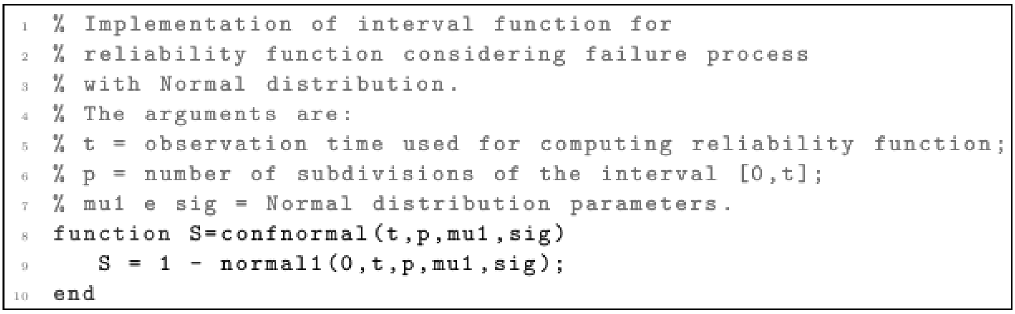
```
% Implementation of interval function for mean time
% to failure considering failure process with Weibull
% distribution.
% The arguments are:
% k = shape parameter of the Weibull distribution;
% lambda = scale parameter of the Weibull distribution;
% max = value that represents numericaly the infinity;
% n = number of subdivisions of the interval [0,max];
% p = number of subdivisions employed in the confweibull method.
function S=mttfIntervalWeibull(k, lambda, max, n, p)
    i = 0;
    h = max/n;
    nArray=1:n+1;
    x(nArray) = h*(nArray-1) + h/2;
    temp = 0;
    t1 = [1,1];
    for i=1:n
        t1 = confweibull(x(i),p,k,lambda);
        temp = temp + t1;
    end
    S = temp*h;
end
```

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



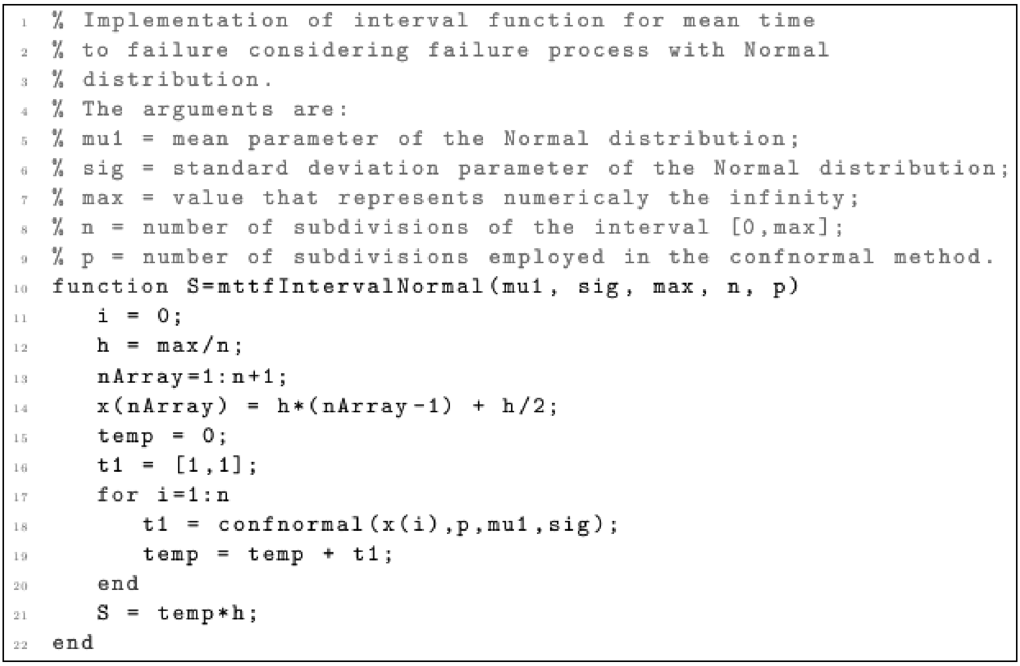
```
% Implementation of interval function for
% failure rate considering Weibull distribution.
% The arguments are:
% t = instant of time used for computing failure rate;
% p = number of subdivisions of the interval [0,t];
% k = shape parameter of the Weibull distribution;
% lambda = scale parameter of the Weibull distribution.
function [S]=failureRateIntervalWeibull(t,p,k,lambda)
    syms x;
    yy = (k*((1/lambda)^k)*(x^(k-1)))*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;
end
```

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



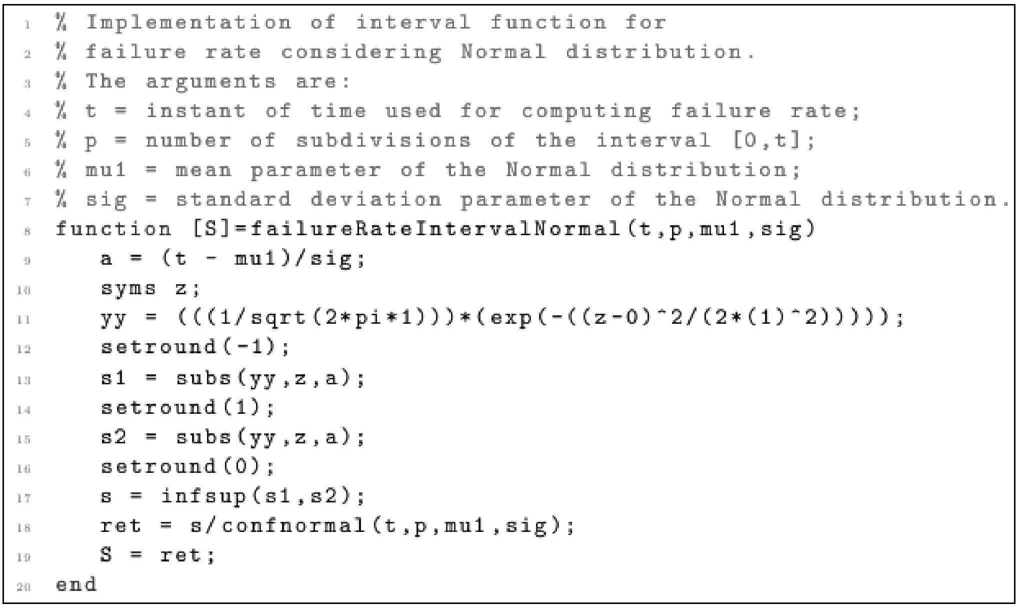
```
% Implementation of interval function for
% reliability function considering failure process
% with Normal distribution.
% The arguments are:
% t = observation time used for computing reliability function;
% p = number of subdivisions of the interval [0,t];
% mu1 e sig = Normal distribution parameters.
function S=confnormal(t,p,mu1,sig)
   S = 1 - normal1(0,t,p,mu1,sig);
end
```

Figure 16: Interval enclosure for reliability function considering component with Normal failures distribution.



```
% Implementation of interval function for mean time
% to failure considering failure process with Normal
% distribution.
% The arguments are:
% mu1 = mean parameter of the Normal distribution;
% sig = standard deviation parameter of the Normal distribution;
% max = value that represents numericaly the infinity;
% n = number of subdivisions of the interval [0,max];
% p = number of subdivisions employed in the confnormal method.
function S=mttfIntervalNormal(mu1, sig, max, n, p)
   i = 0;
   h = max/n;
   nArray=1:n+1;
   x(nArray) = h*(nArray-1) + h/2;
   temp = 0;
   t1 = [1,1];
   for i=1:n
      t1 = confnormal(x(i),p,mu1,sig);
      temp = temp + t1;
   end
   S = temp*h;
end
```

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



```
% Implementation of interval function for
% failure rate considering Normal distribution.
% The arguments are:
% t = instant of time used for computing failure rate;
% p = number of subdivisions of the interval [0,t];
% mu1 = mean parameter of the Normal distribution;
% sig = standard deviation parameter of the Normal distribution.
function [S]=failureRateIntervalNormal(t,p,mu1,sig)
    a = (t - mu1)/sig;
    syms z;
    yy = (((1/sqrt(2*pi*1)))*(exp(-((z-0)^2/(2*(1)^2)))));
    setround(-1);
    s1 = subs(yy,z,a);
    setround(1);
    s2 = subs(yy,z,a);
    setround(0);
    s = infsup(s1,s2);
    ret = s/confnormal(t,p,mu1,sig);
    S = ret;
end
```

Figure 18: Interval enclosure for hazard rate function considering component with Normal failures distribution.