KfK 5019
April 1992

ZERBERUS - the Code for Reliability Analysis of Crack Containing Structures

L. Cizelj, H. Riesch-Oppermann
Institut für Materialforschung
Projekt Kernfusion

Kernforschungszentrum Karlsruhe

Kernforschungszentrum Karlsruhe
Institut für Materialforschung
Projekt Kernfusion

KfK 5019

ZERBERUS - the Code for Reliability Analysis of Crack Containing Structures

L. Cizelj*
H. Riesch-Oppermann

*Reactor Engineering Division
"Jožef Stefan" Institute, Ljubljana

Kernforschungszentrum Karlsruhe GmbH, Karlsruhe

Als Manuskript gedruckt
Für diesen Bericht behalten wir uns alle Rechte vor

Kernforschungszentrum Karlsruhe GmbH
Postfach 3640, 7500 Karlsruhe 1

ISSN 0303-4003

ABSTRACT

The description of the ZERBERUS probabilistic fracture mechanics code is given in this report. Brief description of the First- and Second Order Reliability Methods, being the theoretical background of the code, is given. The code structure is described in detail, with special emphasis to the new application fields. The numerical example investigates failure probability of steam generator tubing affected by stress corrosion cracking. The changes necessary to accommodate this analysis within the ZERBERUS code are explained. Analysis results are compared to different Monte Carlo techniques.

ZUSAMMENFASSUNG

Dieser Bericht enthält eine Programmbeschreibung des Programmpakets ZERBERUS für die probabilistische **Z**uverlässigkeitanalyse **ER**müdungs**BE**anspruchter **R**ißbehafteter **(U)** Strukturen. Einem kurzen Überblick über die im Programm benutzten First- und Second-Order Näherungsmethoden zur Berechnung der Ausfallwahrscheinlichkeit folgt eine detaillierte Beschreibung der Programmstruktur, wobei der besondere Schwerpunkt auf neuen Anwendungmöglichkeiten liegt. Ein Rechenbeispiel zeigt, wie die Berechnung der Ausfallwahrscheinlichkeit von Dampferzeugerrohren aufgrund von Spannungsrißkorrosion durchgeführt wurde und welche Änderungen für die Bearbeitung dieses Beispiels an ZERBERUS notwendig waren. Zusätzlich werden einige Ergebnisse dargestellt und mit Resultaten verschiedener Monte-Carlo verglichen.

ACKNOWLEDGEMENTS

This work was supported by the International Office of the KFA Jülich and Ministery for Science and Technology of Slovenia under the joint research project **Probabilistic Fracture Mechanics Analysis of Heat Exchanger Tubes.**

CONTENTS

LIST OF TABLES . . . v

1 INTRODUCTION . . . 1

2 FIRST-ORDER RELIABILITY METHOD . . . 2
2.1 Failure probability calculations . . . 2
2.2 Design point determination . . . 3
2.3 Sensitivity factors . . . 4

3 SECOND-ORDER RELIABILITY METHOD . . . 4

4 FAILURE MODES . . . 5
4.1 Linear elastic fracture mechanics . . . 5
4.2 Plastic limit load . . . 5
4.3 Two criteria approach . . . 6

5 CRACK PROPAGATION LAWS . . . 8

6 NUMERICAL EXAMPLES . . . 8
6.1 Failure function . . . 8
6.2 Crack length distribution . . . 9
6.3 Crack propagation . . . 9
6.4 Data summary . . . 10
6.5 Results . . . 11

7 CONCLUDING REMARKS . . . 12

8 REFERENCES . . . 13

APPENDIX A: *PROGRAMMING CONSIDERATIONS* . . . A-1

APPENDIX B: *FIRST-ORDER RELIABILITY METHOD LIBRARY* . . . B-1

APPENDIX C: *SECOND-ORDER RELIABILITY METHOD LIBRARY* . . . C-1

APPENDIX D: *PROBABILITY DISTRIBUTION LIBRARY* . . . D-1

APPENDIX E: *FAILURE FUNCTION LIBRARY* . . . E-1

APPENDIX F: *CRACK GROWTH LIBRARY* . . . F-1

APPENDIX G: *DATA PREPARATION* . . . G-1

APPENDIX H: *USER DEFINED ROUTINES* . . . H-1
H.1 Failure function . . . H-3
H.2 Additional probability distributions . . . H-4

LIST OF TABLES

Table I Options for stress intensity factors 6

Table II Plastic collapse load options 7

Table III Steam generator data summary 10

Table IV Comparison of failure probabilities (in %), without the tube-sheet reinforcing effect 10

Table V Comparison of failure probabilities (in %), with the tube-sheet reinforcing effect 11

Table VI First-order reliability related variables A-3

Table VII Code control variables A-4

Table VIII Second-order reliability related variables A-5

Table IX Probability distribution related variables A-5

Table X Failure function related variables A-6

Table XI Crack propagation related variables A-7

Table XII Distribution functions D-3

Table XIII Distribution access routines D-4

1 INTRODUCTION

The probabilistic fracture mechanics utilizes the failure criteria provided by fracture mechanics to estimate the failure probability of a cracked component. Usually, such approach is applied in the field of highly reliable components, e.g. aircraft and nuclear industry [1], which implies that very low failure probabilities (say below the 10^{-6}) are expected. Such estimation requires, besides accurate data, an efficient computational procedure, provided that an analytical solution (see eq. (1)) is hardly to be expected [2]. In most cases, different types of Monte Carlo simulation techniques were utilized. Usually, simulation results are considered "exact". The simulation types are ranging from simple but computationally exhaustive Direct Monte Carlo to the various techniques of advanced importance sampling, offering acceptable computational requirements but having a significant degree of sophistication (see for example refs. [3], [4] and [5]).

A need for efficient approximate methods with acceptable accuracy was therefore obvious. The development of the First- and Second Order Reliability Methods (FORM and SORM, respectively) in the statistical reliability theory was of utmost importance ([6], [7] and refs. therein). The application of the FORM/SORM methods in probabilistic fracture mechanics was initiated with the development of the **ZERBERUS** (**Z**uverlässigkeitanalyse **ER**müdungs**BE**anspruchter **R**ißbehafteter (**U**) **S**trukturen) code [8], [9], being described in this report. Indeed, FORM and SORM provide an easy to use and computationally very efficient way of determination of failure probabilities, unfortunately in the absence of an error estimation possibility (FORM), although they tend to be asymptotically correct (SORM) [10]. Under such circumstances, the applicability of the FORM/SORM should be demonstrated in each application field separately. The research already performed proved its efficiency and accuracy in cases of fatigue and creep cracking induced structural failure modes [9], [11].

The **ZERBERUS** code utilizes, as mentioned above, the FORM and SORM approach in the probabilistic fracture mechanics analysis of crack containing structures. In the present report, the theoretical background of the code, together with its FORTRAN 77 implementation, is presented in order to allow for new applications of the code, as for example the assessment of reliability of axially stress-corrosion cracked tubes of nuclear power plant steam generators [12], [13], to prove the applicability of the code also for the stress-corrosion cracking induced failure modes.

A brief review of the basic FORM and SORM formulations is therefore given in Chapters 2 and 3. The review is centered towards explaining the code structure rather than the background of the methods, which is given elsewhere ([9] and references therein). Further, the description of cracked structure failure modes covered by the current version of code is given in Chapter 4. We would like to point out the compatibility of failure modes with the Monte Carlo based probabilistic fracture mechanics code **PARIS** [2]. The currently implemented stable crack growth laws are covered in Chapter 5. The detailed description of the code, including the data preparation and hints for the coding of user supplied subroutines is given in Appendices.

2 FIRST-ORDER RELIABILITY METHOD

2.1 Failure probability calculations

Statistical reliability theory in structural mechanics (probabilistic fracture mechanics being its special case) deals with the determination of failure probabilities P_f of structural components from the scatter of the applied loads and structural resistance properties. The failure behaviour of the structure is described by a failure function $g(x)$, depending on *basic random variables* $x=(x_1, \ldots, x_n)$ which denote applied loads and structural resistance parameters such as dimensions and material properties. By definition, $g(x)<0$ implies failure, whereas no failure occurs for $g(x)>0$. $g(x)=0$ defines the so called *failure surface*, being a hyper-surface in the n-dimensional space of the basic variables. The failure surface separates the *failure domain F* (with $g(x)>0$) and the *safe domain* ($g>0$). The failure probability P_f can be calculated as the probability content of the failure domain F:

$$P_f = \int_F f_1(x_1) \ldots f_n(x_n)\, dx_1 \ldots dx_n \tag{1}$$

where $f_i(x_i)$ represent the probability densities of respective basic variables x_i, which are for the sake of simplicity assumed to be stochastically independent. Eq. (1) is sometimes referred to as *failure integral.*

The analytical solutions of the failure integral (eq. (1)) are limited to a few of very special cases. However, for *standard normally* distributed basic variables and linear failure function, the analytical solution of the failure integral (eq. (1)) is given:

$$P_f = \Phi(-\beta) \tag{2}$$

$\Phi(\ )$ being the cumulative standard normal distribution and β the so called *reliability index*, which is representing the distance between the origin of the space of basic variables and the *design point* x^* on the failure surface:

$$\beta = \frac{g(x^*) - a \cdot x}{|a|} \tag{3}$$

with

$$a = \nabla g(x) = \left(\frac{\partial g(x^*)}{\partial x_1}, \ldots, \frac{\partial g(x^*)}{\partial x_n} \right) \tag{4}$$

The design point (x^*) is a point on the failure surface having the minimum distance to the origin of the space of basic variables, thus contributing the most to the failure probability. Because of the last property, some authors call it the *most likely failure point.*

In the case of non-linear failure function, a linearisation at the design point provides an approximate value of the failure probability:

$$P_f \approx \Phi(-\beta) \tag{5}$$

For non-normal basic variables, a transformation from the physical (x) space to the standard normal (u) space is performed. If the basic variables are assumed stochastically independent, the transformation is defined by:

$$U_i = \Phi^{-1}\left(F_i(X_i)\right) \tag{6}$$

with standard normal variables U_i and its inverse by:

$$X_i = F_i^{-1}\left(\Phi(U_i)\right) \tag{7}$$

Application of the inverse transformation (eq. (7)) allows the evaluation of the failure function in the physical ($\boldsymbol{x}$) space.

2.2 Design point determination

The iterative determination of the design-point is in general performed in normal $\boldsymbol{u}$ space. As discussed above, the design point is a point on the failure surface with the minimum distance to the coordinate origin. Starting with an initial value $\boldsymbol{u}^{(0)}$, the design point $\boldsymbol{u}^*$ follows as the limit of the series [14]:

$$\boldsymbol{u}^{(m+1)} = \left(\boldsymbol{u}^{(m)} \cdot \boldsymbol{a}^{(m)}\right) \cdot \boldsymbol{a}^{(m)} + \frac{g_u\left(\boldsymbol{u}^{(m)}\right)}{\left|\, \nabla g_u\left(\boldsymbol{u}^{(m)}\right) \right|} \cdot \boldsymbol{a}^{(m)} \tag{8}$$

with

$$\boldsymbol{a}^{(m)} = -\frac{\nabla g_u\left(\boldsymbol{u}^{(m)}\right)}{\left|\, \nabla g_u\left(\boldsymbol{u}^{(m)}\right) \right|} \tag{9}$$

and

$$\nabla g_u(\boldsymbol{u}) = \left(\frac{\partial g_u}{\partial u_1}(\boldsymbol{u}),\ \ldots,\ \frac{\partial g_u}{\partial u_n}(\boldsymbol{u})\right) \tag{10}$$

The first term of eq. (8) represents a projection of $\boldsymbol{u}^{(m)}$ on the unit vector $\boldsymbol{a}^{(m)}$ being normal to the tangent hyper-plane of the failure function in the $\boldsymbol{u}$ space $g_u(\boldsymbol{u}^{(m)})$ in the current design point approximation. The second term may be regarded as a correction term.

The following transformation should be defined in order to obtain g_u and ∇g_u:

$$T\colon\ u_i = \Phi^{-1}\left(F_i(x_i)\right);\ i = 1,\ \ldots,\ n \tag{11}$$

with the inverse

$$T^{-1}\colon\ x_i = F^{-1}\left(\Phi(x_i)\right);\ i = 1,\ \ldots,\ n \tag{12}$$

Failure function in the normal space g_u is then obtained through:

$$g(x) = g\left(T^{-1}(\Phi(u)\right) = g_u(u) \tag{13}$$

and its partial derivatives as:

$$\frac{\partial g_u}{\partial u_i} = \frac{\partial g}{\partial x_i}\,\frac{\partial x_i}{\partial u_i} = \frac{\partial g(x)}{\partial x_i}\,\frac{\varphi(u_i)}{f_i(x_i)} = \frac{\partial g(x)}{\partial x_i}\,\frac{\varphi(\Phi^{-1}(F_i(x_i)))}{f_i(x_i)} \tag{14}$$

2.3 Sensitivity factors

The reliability index β is defined through the minimum distance between the failure surface and the origin of the normal space. Therefore, the failure probability sensitivity to the scatter of each basic variable may be expressed as follows:

$$\frac{\partial \beta}{\partial u_i} = \frac{\partial}{\partial u_i} \sqrt{\sum_{i=1}^{n} u_j^2} = \frac{u_i^*}{|u|} = \alpha_i^* \tag{15}$$

α_i^* being denoted as sensitivity factors.

3 SECOND-ORDER RELIABILITY METHOD

Various methods have been suggested to improve the accuracy of FORM calculations and to give a rough estimate of the approximation quality ([14], [15]). The general idea is to approximate the failure surface by a quadratic hyper-surface rather than by a hyper-plane. The main curvatures κ_j of the quadratic hyper-surface at the design point are equal to those of the failure surface. This leads to the so called Second-Order Reliability Methods (SORM). Using all n-1 main curvatures, the SORM approximation of the failure probability is obtained from [14]:

$$P_f \approx S_1 + S_2 + S_3 \tag{16}$$

with

$$S_1 = \Phi(-\beta) \prod_{j=1}^{n-1} (1 - \beta\kappa_j)^{-0.5} \tag{17}$$

$$S_2 = [\beta\Phi(-\beta) - \varphi(\beta)] \left\{ \prod_{j=1}^{n-1} (1 - \beta\kappa_j)^{-0.5} - \prod_{j=1}^{n-1} (1 - (\beta+1)\kappa_j)^{-0.5} \right\} \tag{18}$$

$$S_3 = (\beta + 1)\, [\beta\Phi(-\beta) - \varphi(\beta)] \left\{ \prod_{j=1}^{n-1} (1 - \beta\kappa_j)^{-0.5} - Re\left[\prod_{j=1}^{n-1} (1 - (\beta+i)\kappa_j)^{-0.5} \right] \right\} \tag{19}$$

Re[ ] represents the real part of the complex argument and i imaginary unit. S_1 is an asymptotic approximation of P_f being exact for $\beta \to \infty$. S_2 and S_3 are correction terms.

The curvatures κ_j are obtained in the following way. First, the normal (u) space coordinate system of the standard normal space is rotated, so that the design point becomes part of the last axis of the rotated normal (y) space. In other words, the last basic vector of the (y) space coincides with the design point position vector in the normal (u) space. This is achieved through the following transformation:

$$Y = \mathbf{D}^T U \tag{20}$$

and, provided that $\mathbf{D}^{-1} = \mathbf{D}^T$:

$$U = \mathbf{D} Y \tag{21}$$

The last (n^{th}) basis vector of $\mathbf{D}$ is simply vector of sensitivity factors α (eq. (15)), while the others are obtained through orthonormalisation. Defining $\mathbf{G}_y$ as the matrix of second derivatives of failure function in (y) space at the design point, the curvatures are obtained as solutions of characteristic equation:

$$\det\left(\frac{\mathbf{G}_y}{\partial g(y^*)/\partial y_n} - \kappa \mathbf{I}\right) = 0 \tag{22}$$

$\mathbf{I}$ represents the unit matrix and $y^* = \mathbf{D}^T u^*$ the position vector of the design point in (y) space.

4 FAILURE MODES

The FORM and SORM methods for approximate calculation of failure probabilities were described in previous chapters. It should be pointed out again here that they are capable of solving any failure probability calculation problem, as long as the linear (or quadratical) approximation of the failure surface in the normal space is reasonable. In this chapter, the formulation of the fracture mechanics failure surface library, being part of the standard **ZERBERUS** code version is described. This description also intends to serve as an example for development of other applications (in other words, including other failure surface definitions) by modifications or replacement of subroutines described in App. E.

Three different failure surface descriptions are used in the standard version of the code. They cover the fracture mechanics theory ranging from linear-elastic to fully plastic behaviour. The way of implementing the time dimension (e.g. stable crack propagation) is presented in Chapter 5.

4.1 Linear elastic fracture mechanics

The (brittle) failure is assumed to occur when the stress intensity factor K_I reaches the critical value K_{Ic}. This implies the following definition of the failure function g:

$$g(a, a/c, K_{Ic}, \sigma) = K_{Ic} - K_I \tag{23}$$

assuming that the crack behaviour is influenced mostly by crack depth a, crack depth to length ratio a/c, applied stress σ and material characteristic value of critical stress intensity factor K_{Ic}. The stress intensity factor K_I is usually defined as:

$$K_I = \sigma\sqrt{a\pi}\ Y(a/t, a/c) \tag{24}$$

where *Y(a/t, a/c)* is a function of the problem geometry (*t* being a characteristic length dimension). Therefore, the problem of failure function calculation reduces to the determination of function *Y()*.

Options already included into the code are listed in Table I and extensively discussed in refs [2], [16], [17], [18], [19], [20], [21], [22], [23] and [24].

4.2 Plastic limit load

The usual assumption made in assessing ductile failure is that the applied stress σ should remain below the so called limit load stress σ_L. Therefore, the failure function may be written as:

Table I Options for stress intensity factors

Option	Crack model	Loading	Reference
1	Center cracked plate	Tension	[16]
2	Three-point bending specimen	Bending	[16]
3	Axial through-wall crack in a tube	Internal pressure	[17]
4	Circumferential through-wall crack in a tube	Tension	[18]
5	Circumferential through-wall crack in a tube	Tension & bending	[19]
6	Axial through-wall crack in a tube	Tension & bending	[20]
7	User option for **one-dimensional** cracks	N/A	N/A
8	Surface crack in a plate	Tension	[21]
9	Surface crack in a plate	Tension & bending	[21]
10	Axial part-through crack in a tube (inside)	Internal pressure	[22]
11	Axial part-through crack in a tube (inside)	Internal pressure	[23]
12	Circumferential part-through crack in a tube (inside)	Tension	[24]
13	Circumferential part-through crack in a tube (outside)	Tension & bending	[19], [21]
14	Axial part-through crack in a tube (outside)	Tension & bending	[20], [21]
15	User option for **two-dimensional** cracks	N/A	N/A

One-dimensional cracks are covered in options **1-7**.
Two-dimensional cracks are covered in options **8-15**.

$$g(a,a/c,\sigma_f,\sigma) = \sigma_L - \sigma \tag{25}$$

where the limit load stress is defined in terms of material characteristic flow stress σ_f :

$$\sigma_L = \sigma_f\, M(a/t,\ a/c) \tag{26}$$

where *M()* is again a function of the problem geometry. Options already included in the code are listed in Table II and extensively discussed in refs. [2], [25], [17], [26], [19], [20], [24], [27] and [28].

4.3 Two criteria approach

To assess the wide transition region between brittle and fully ductile fracture, a two criteria approach was proposed [29]. Thus, the failure function is defined as:

Table II Plastic collapse load options

Option	Crack model	Loading	Reference
1	Center cracked plate	Tension	[25]
2	Three-point bending specimen	Bending	[25]
3	Axial through-wall crack in a tube	Tension & bending	[17]
4	Axial through-wall crack in a tube	Internal pressure	[26]
5	Circumferential through-wall crack in a tube	Tension & bending	[19]
6	Circumferential through-wall crack in a tube	Tension	[24]
7	Axial through-wall crack in a tube	Tension & bending	[20]
8	User option for **one-dimensional** cracks	N/A	N/A
9	Surface crack in a plate	Tension	[25]
10	Surface crack in a plate	Tension	[27]
11	Surface crack in a plate	Tension	[28]
12	Surface crack in a plate	Tension & bending	[28]
13	Axial part-through crack in a tube (outside)	Internal pressure	[17]
14	Circumferential part-through crack in a tube (inside)	Tension	[24]
15	User option for **two-dimensional** cracks	N/A	N/A

One-dimensional cracks are covered in options 1-8.
Two-dimensional cracks are covered in options 9-15.

$$g(a,\ a/c,\ K_{Ic},\ \sigma_f,\ \sigma) = \frac{S_R}{\sqrt{\frac{-8}{\pi^2}\ln\left[\cos\left(\frac{\pi}{2}\ S_R\right)\right]}} - K_R \tag{27}$$

where

$$K_R = \frac{K_I}{K_{Ic}} \quad \text{and} \quad S_R = \frac{\sigma}{\sigma_L} \tag{28}$$

with K_{Ic} and σ_L as in eqs. (23) and (25), respectively. Obviously, both *Y()* and *M()* function calculation are again the most important parts of the failure function determination.

5 CRACK PROPAGATION LAWS

Crack geometry (and its probability distribution!) usually change with time because of the stable crack propagation, induced for example by load cycling. To account for this phenomenon, the crack geometry should be formulated as a function of the time history of the operational conditions. Since the crack geometry before the operation is usually known, only the crack growth should be evaluated within the code. In the case of fatigue, the crack growth with respect to number of load cycles N may be expressed as [30]:

$$\frac{da}{dN} = C(\Delta K)^n \tag{29}$$

with $\Delta K = K_{max} - K_{min}$ being the range of stress intensity factors during the load cycle. Some other authors have proposed different approaches, for example [31]:

$$\frac{da}{dn} = C(\Delta K_{eff})^n \tag{30}$$

with

$$\Delta K_{eff} = \frac{\Delta K}{\sqrt{1-R}} \qquad \text{and} \qquad R = \frac{\sigma_{min}}{\sigma_{max}} \tag{31}$$

or [32]:

$$\frac{da}{dN} = \frac{C(\Delta K)^n}{(1-R)K_{Ic} - \Delta K} \tag{32}$$

Other appropriate expressions may be easily incorporated into the code by changing or replacing the subroutines described in App. F.

6 NUMERICAL EXAMPLES

The numerical example has been chosen to represent both the ZERBERUS code performance and upgrade capabilities. The brief description of the example background is given in this paragraph. For more details, the reader is referred to [12]. Changes of the ZERBERUS code required to enable calculations are described in App. H. Some further numerical examples are presented in [9].

An axial through-wall crack in a pressurized tube is considered. The tubes are part of the steam generator tube bundle. Cracks are generated by the progressive stress-corrosion process. Some effects of the non-destructive examination and tube bundle maintenance are also included. The only maintenance action considered is removing the tube with long cracks from service, i.e. plugging.

6.1 Failure function

The failure function is based on the plastic limit load model:

$$g(a_1, R, t, K, \delta, \sigma_Y + \sigma_M) = \sigma_f - m\sigma \tag{33}$$

with bulging factor m:

$$m = 0.614 + 0.386e^{\left(\frac{-2.25\,a}{\sqrt{Rt}}\right)} + 0.866\left(\frac{a}{\sqrt{Rt}}\right) \tag{34}$$

a_1, R and t being crack half-length at the end of inspection cycle, tube mean radius and tube wall thickness, respectively. Flow stress σ_f is defined by the means of yield stress σ_Y and ultimate tensile strength σ_M and adjusted for the operating temperature conditions δ where appropriate:

$$\sigma_f = K(\sigma_Y + \sigma_M)\delta \tag{35}$$

The membrane stress perpendicular to the crack direction σ is pressure (p) induced tube hoop stress:

$$\sigma = p\left(\frac{R}{t} - \frac{1}{2}\right) \tag{36}$$

The tubes are fixed into a tube sheet which provides additional circumferential rigidity, thus obstructing the bulging. The *Tube Sheet Reinforcing Factor* has been proposed to account for this effect. It is defined as correction coefficient to the flow stress factor K at critical crack length (see eq. (35)):

$$RF(a_c) = 1 + 10\cdot e^{\left(-1.8\cdot\frac{a_c}{\sqrt{R\cdot t}}\right)} \tag{37}$$

a_c represents the critical crack length in a free span tube. Unfortunately, its use is restricted only to the cracks, tangent to the tube sheet, which may not be true for the numerous cracks propagating from the tube sheet. Two cases are therefore considered in this numerical example. The first considers the beneficial effect of the tube sheet, the second not.

6.2 Crack length distribution

When steam generator in-service inspection is completed, the probability density of measured crack lengths can be estimated. Tubes with cracks equal or exceeding the plugging limit *PL* are removed from service. Thus, the crack length after the in-service inspection is defined by the random variable a_0:

$$a_0 = \begin{cases} a_m, & a_m < PL \\ 0, & a_m \geq PL \end{cases} \tag{38}$$

a_m being random variable representing measured crack length.

Plugging limits are *allowable* crack lengths. A parametric study of the plugging limit influence to the failure probability has been performed.

6.3 Crack propagation

A fraction of cracks is propagating in a stable manner during the period between two consecutive inspections. A simple stochastic combination of stochastic propagation law proposed in [33] and measured crack length is applied at the moment to yield end of inspection cycle crack length a_1

$$a_1 = a_0 + a_g \tag{39}$$

6.4 Data summary

A typical steam generator as installed in Slovenian Krško nuclear power plant subjected to hypothetical accidental operating conditions is taken as a numerical example. The accidental condition considered is feed-line break with differential pressure of 196 bar. The summary of geometrical and material data is outlined in Table III , together with assumed "as measured" crack length distribution.

Table III Steam generator data summary

Variable	Distribution		Unit	Comment
	Type	Parameters		
R	Normal	μ=9.525, σ=0.0254	mm	Mean radius
t	Normal	μ=1.055, σ=0.0464	mm	-
a_m	Gamma	α=10.3, β=1.0	mm	assumed
a_g	Gamma	α=1.25, β=0.8	mm	-
K	Normal	μ=0.545, σ=0.03	-	-
δ	Normal	μ=0.928, σ=0.003	-	-
$(\sigma_Y+\sigma_M)$	Normal	μ=1080., σ=54.3	MPa	

Distributions are not truncated.

Table IV Comparison of failure probabilities (in %), without the tube-sheet reinforcing effect

Plugging limit	FORM	SORM	DMC	ISM	ASM	ESM
∞	2.374	3.135	3.084	3.038	3.030	3.093
20.	2.126	2.667	2.592	2.557	2.536	2.497
18.	1.743	2.012	1.967	1.945	1.949	1.900
16.	1.147	1.171	1.187	1.174	1.186	1.139
14.	0.5819	0.5357	0.5602	0.5566	0.5548	0.5351
12.	0.2351	0.2066	0.2157	0.2136	0.2127	0.2084
10.	0.0799	0.0702	0.0733	0.0725	0.0717	0.0754
8.	0.0239	0.0215	0.0221	0.0219	0.0219	0.0234

DMC direct Monte Carlo method; **ISM** importance sampling MC method;
ASM adaptive sampling MC method; **ESM** efficient sampling MC method;

6.5 Results

The results of the first case (without tube sheet reinforcing effects) are listed in Table IV . In addition to FORM and SORM results, different Monte Carlo simulation results are presented. Column DMC represents direct MC simulation with 10^7 numerical experiments, while ISM, ASM and ESM stand for importance sampling, adaptive and efficient sampling MC methods, respectively. For illustrative purposes, the direct MC simulation required as much as 20 hours of CPU on VAX 4000 series computers. Equivalent FORM/SORM computations are completed within 2 minutes of the same computer CPU. The very good agreement of SORM results with various Monte Carlo techniques is obvious. FORM results are within expected error of the magnitude of 10%.

Table V Comparison of failure probabilities (in %), with the tube-sheet reinforcing effect

Plugging limit	FORM	SORM	DMC	ISM	ASM	ESM
∞	1.520	1.944	1.908	1.881	1.862	1.896
20.	1.270	1.472	1.428	1.405	1.410	1.413
18.	0.9450	0.9443	0.936	0.9274	0.9287	0.9342
16.	0.5497	0.4631	0.468	0.4686	0.4666	0.4660
14.	0.2526	0.1923	0.198	0.1984	0.1973	0.2022
12.	0.0959	0.0710	0.0729	0.0736	0.0735	0.0659
10.	0.0314	0.0236	0.0253	0.0243	0.0243	0.0274
8.	0.0092	0.0071	0.0079	0.0072	0.0072	0.0078

DMC direct Monte Carlo method; **ISM** importance sampling MC method;
ASM adaptive sampling MC method; **ESM** efficient sampling MC method;

7 CONCLUDING REMARKS

The ZERBERUS code is a probabilistic fracture mechanics code based on approximate failure probability determination techniques. In particular, the First and Second Order Reliability methods (FORM and SORM, respectively) are implemented. Its main advantage over conventional Monte Carlo simulation codes is lower computational intensity at the same level of accuracy. However, difficult approximation error determination requests verification in each new application field.

The description of the theoretical background and the code structure is given in this report. Besides being some kind of Users Guide, this report intends to help the user to change the code to allow for new applications. As the applicability of ZERBERUS code has already been proven in cases of fatigue and creep induced cracks, the numerical example considered introduces stress-corrosion cracking. Further, the changes in code structure to accommodate the analysis of numerical example are explained. Finally, the ZERBERUS results are compared to different Monte Carlo techniques.

8 REFERENCES

[1] Provan, J.W., ed. (1987): **Probabilistic Fracture Mechanics and Reliability**, Martinus Nijhoff Publishers.

[2] Brueckner-Foit, A. (1987): *Probabilistic Analysis of Crack Containing Structures with PARIS Code*, Kernforschungszentrum Karlsruhe, Report KfK 4286.

[3] Brueckner, A. (1987): *Numerical Methods in Probabilistic Fracture Mechanics*, in: J.W. Provan, ed(s).: **Probabilistic Fracture Mechanics and Reliability**, Martinus Nijhoff Publishers, pp. 351-386.

[4] Schueller, G.I., R.Stix (1987): *A Critical Appraisal of Methods to Determine Failure Probabilities*, Structural Safety, Vol. 4, pp. 293-309.

[5] Schueller, G.I., C.G.Bucher, U.Bourgund, W.Ouypornprasert (1989): *On Efficient Computational Schemes to Calculate Structural Failure Probabilities*, Probabilistic Engineering Mechanics, Vol. 4, No. 1, pp. 10-18.

[6] Hohenbichler, M., R.Rackwitz (1983): *First-Order Concepts in System Reliability*, Structural Safety, Vol. 1, pp. 177-188.

[7] Dolinski, K. (1983): *First-Order Second-Moment Approximation in Reliability of Structural Systems: Critical Review and Alternative Approach*, Structural Safety, Vol. 1, pp. 211-231.

[8] Riesch-Oppermann, H., A.Brueckner-Foit (1988): *First- and Second-Order Approximations of Failure Probabilities in Probabilistic Fracture Mechanics*, Reliability Engineering and System Safety, Vol. 23, No. 3, pp. 183-194.

[9] Riesch-Oppermann, H. (1989): *Anwendung von Naeherungsmethoden zur Berechnung der Ausfallwahrscheinlichkeit rissbehafteter Strukturen unter Ermuedungs- und Kriechbeanspruchung*, Dissertation, University of Karlsruhe.

[10] Hohenbichler, M., S.Gollwitzer, W.Kruse, R.Rackwitz (1987): *New Light on First- and Second-Order Reliability Methods*, Structural Safety, Vol. 4, pp. 267-284.

[11] Riesch-Oppermann, H., A.Brueckner-Foit (1991): *Probabilistic Fracture Mechanics Applied to High Temperature Reliability*, Nuclear Engineering and Design, Vol. 128, pp. 193-200.

[12] Mavko, B., L.Cizelj (1992): *Failure Probability of Axially Cracked Steam Generator Tubes: A Probabilistic Fracture Mechanics Model*, Nuclear Technology, in printing.

[13] Mavko, B., L.Cizelj, G.Roussel (1991): *Steam Generator Tube Rupture Probability Estimation - Study of the Axially Cracked Tubes Case*, CSNI/UNIPEDE Specialist Meeting on Operating Experience with Steam Generators, Brussels, Belgium.

[14] Madsen, H.O, S.Krenk, N.C.Lind (1986): *Methods of Structural Safety*, Prentice Hall, Englewood Cliffs.

[15] Fiessler, B., H.-J.Neumann, R.Rackwitz (1979): *Quadratic Limit States in Structural Reliability Theory*, Journal of Engineering Mechanics ASCE, Vol. 105, pp.661-676.

[16] Tada, H., P.C.Paris, G.R.Irwin (1985): *The Stress Analysis of Cracks Handbook*, 2nd ed., Paris Prod. Inc., St. Louis, Missouri.

[17] Kiefner, J.F., W.A.Maxey, R.J.Eiber, A.R.Duffy (1973): *Failure Stress Levels of Flaws in Pressurized Cylinders*, ASTM STP 536, pp. 461-481.

[18] Forman, R.G., J.C.Hickman, V.Shivakumar (1985): *Stress Intensity Factors for Circumferential Through-Wall Cracks in Hollow Cylinders Subjected to Combined Tension and Bending Loads*, Engineering Fracture Mechanics, Vol 21., pp. 563-571.

[19] Erdogan, F. (1982): *Theoretical and Experimental Study of Fracture in Pipelines Containing an Initial Surface Flaw*, report DOT-RSPA-DMA-50/83/3, National technical Information Service, Springfield, Virginia.

[20] Erdogan, F., M.Ratwani (1974): *Fracture Initiation and Propagation in a Cylindrical Shell Containing an Initial Surface Flaw*, Nuclear Engineering and Design, Vol. 27, pp. 14-29.

[21] Newman, J.C., I.S.Raju (1981): *An Empirical Stress Intensity Factor Equation for the Surface Crack*, Engineering Fracture Mechanics, Vol. 15, pp. 185-192.

[22] Newman, J.C, I.S.Raju (1980): *Stress Intensity Factor for Internal Surface Cracks in Cylindrical Pressure Vessels*, Journal of Pressure Vessel Technology ASME, Vol. 102, pp. 342-346.

[23] ASME Boiler and Pressure Vessel Code, Section XI.

[24] Harris, D.O, E.Y.Lim, D.D.Dedhia (1981): *Probability of Pipe Fracture in the Primary Coolant Loop of PWR Plant*, Report NUREG/CR-2189, Vol. 5, Lawrence Livermore Lab.

[25] Chell, G.G. (1979): *Elastic-Plastic Fracture Mechanics*, in G.G.Chell (ed.): **Development in Fracture Mechanics**, Vol. 1, Applied Science Publ., London.

[26] Kitching, R., J.K.Davies, S.S.Gill (1970): *Limit Pressure for Cylindrical Shells with Unreinforced Openings of Various Shapes*, Journal of Mechanical Engineering Science, Vol. 12, pp. 313-330.

[27] Mattheck, C., P.Morawietz, D.Munz, B.Wolf (1984): *Ligament Yielding of a Plate with Semi-Elliptical Surface-Cracks Under Uniform Tension*, Int. Journal of Pressure Vessels and Piping, Vol. 16, pp. 131-143.

[28] Harrison, R., K.Loosemore, I.Milne, A.R.Dowling (1980): *Assessment of the Integrity of Structures Containing Defects*, CEGB/R/H/6, Rev. 2, 1980.

[29] Dowling, A.R., C.H.A.Townley (1975): *The Effect of Defects on Structural Failure: A Two Criteria Approach*, Int. Journal of Pressure Vessels and Piping, Vol. 3, No. 2, pp. 77-107.

[30] Paris, C. (1964): *The Fracture Mechanics Approach to Fatigue*, in **Fatigue**, Proc 10th Sajamore Army Materials Research Conference, Syracuse, N.Y., pp. 107-138.

[31] Walker, K. (1970): *The Effect of Stress Ratio during Crack Propagation and Fatigue for 2024-T3 and 7075-T6 Aluminium Alloy*, ASTM STP 462, pp. 1-13.

[32] Forman, R.G., V.E.Kearney, R.M.Engle (1967): *Numerical Analysis of Crack Propagation in Cyclic-Loaded Structures*, Journal of Basic Engineering, Vol. 89, pp. 459-470.

[33] Hernalsteen, P. (1991): *Prediction Models for the PWSCC degradation Process in Tube Roll Transitions*, NEA-CSNI-UNIPEDE Specialist Meeting on Operating Experience with Steam Generators, Brussels, Belgium.

APPENDIX A: *PROGRAMMING CONSIDERATIONS*

A.1 General

The **ZERBERUS** code is written in FORTRAN 77. It consists of 6 parts, which may be regarded as subroutine and function libraries. The brief description of libraries together with related global variables is given below. Detailed description of the libraries may be found in the following Appendices.

Data storage and transfer is organized by a set of COMMON blocks to enable transparent and fast data access. Real variables are implicitly assumed to be of the DOUBLE PRECISION (REAL*8) type.

Table VI First-order reliability related variables

COMMON block	Variable	Description
BLOCK1	XMAT(10)	Initial values of the design point coordinates. Should be supplied by the means of sub. **EINGAB**, page G-3. Later, the current value of the design point physical coordinates is stored into this variable (sub. **BXMAT**, page B-11).
	FI(10)	Probability densities of basic variables at current design point, as calculated by sub. **BANMAT**, page B-9.
	FIKUM(10)	Cumulative probability distribution values of basic variables at current design point (sub. **BANMAT**, page B-9).
	DELTA	Reliability index, obtained in sub. **BDELTA**, page B-7.
	XAMAT(10)	Used in sub. **EINGAB**, page G-3, for reading of the design point initial values, which are immediately transferred to variable XMAT.
BLOCK3	XNEUMT(10)	Not implemented.
	UMAT(10)	Current design point coordinates in normal space (sub. **BUMAT**, page B-6).
BLOCK4	ANMAT(10)	Failure function gradient in normal space (sub. **BANMAT**, page B-9).
	AMAT(10)	Failure function gradient in physical space (sub. **BAMAT**, page B-8).
BLOCK5	UNMAT(10)	new design point coordinates in normal space (sub. **BUNMAT**, B-10).

A.2 First-Order reliability method library

The set of routines which enable the first order reliability calculations (Chapter 2) are described in App. B. It requires external functions providing the probability distributions of basic variables (eqs. (6) and (7), see also App. D) and failure function value (App. E). The data is supplied by an input file as described in App. H and saved in the COMMON blocks (see Table VI). It should be noted that this set of routines represent a FORM solver and need not be changed to implement another failure mode.

Table VII Code control variables

COMMON block	Variable	Description
BLOCK6	N	Number of basic variables. Input data.
	K	Counter of the design point iterations (sub. **DIRTRA**, page B-4).
	IDRUCK	Printing control variable. See sub. **EINGAB**, page G-3. Input. data.
BLOCK7	FM1	CHARACTER*60 message variable.
	FM2	CHARACTER*60 message variable.
	FM3	CHARACTER*60 message variable.
	FM4	CHARACTER*60 message variable.
BLOCK9	ABBRGR	The convergence criterium for the design point iterative determination. See sub. **ABKRIT**, page B-12. Input data.
	GP(10)	Failure function parameters. Useful in cases when only a fitted curve or surface is available instead of failure function. Input data.
	NGPAR	Number of failure function parameters. Input data.

In the current code version, the FORM part is also controlling the execution. The control variables are presented in Table VII .

A.3 Second-Order reliability method library

This set of routines represent the SORM solver (for theoretical outline see Chapter 3). Routines are described in App. C. It requires the preexisting FORM solution (at least design point!) and the same set of external routines (distribution and failure functions) as FORM part or the code. The related data blocks are outlined in Table VIII .

A.4 Distribution functions library

A set of probability distribution functions is supplied with the **ZERBERUS** code. The routines and access to them are described in App. D. To extend the library, the changes to access routines (Table XIII) are necessary, together with a new function in the form of the in App. D described routines. The probability distribution related variables are explained in Table IX .

A.5 Failure mode library

The set of functions is supplied to provide the failure function values (see App. E-1). Especially the **BGVONX** and **BABLVG** routines may consider special attention. The **BGVONX** routine (page E-3) provides the link between the failure function and the FORM/SORM solvers. Additionally, the link between basic variables and physical quantities

Table VIII Second-order reliability related variables

COMMON block	Variable	Description
BLCK14	RAD(10)	Not implemented.
	PF(10)	Not implemented.
	FLAG(10)	Not implemented.
BLCK15	GXMAT(10, 10)	Matrix of failure function second derivatives in physical *x* space (sub. **BGXMAT**, page C-5).
	GYMAT(10, 10)	Matrix of failure function second derivatives in rotated normal *y* space (sub. **BGYMAT**, page C-10).
	GUMAT(10, 10)	Matrix of failure function second derivatives in normal *u* space (sub. **BGUMAT**, page C-6).
BLCK16	ALFA(10)	Normalized vector directed from the normal *u* space origin towards the design point (sub. **BALFA**, page C-8).
	YMAT(10)	Normal *u* to rotated normal *y* space transformation matrix (eqs. (20) and (21), sub. **ORTHO**, page C-9).
BLCK18	AKAPPA(10)	Main curvatures of the failure surface in normal space at the design point (eq. (22), sub. **KRUMM** on page C-11).
	PTVEDT	SORM approximation of the failure probability (eq. (16), sub. **TVEDT** on page C-13).

Table IX Probability distribution related variables

COMMON block	Variable	Description
BLCK10	VUP(10)	CHARACTER*10 variable containing names of the basic variable probability distributions (func. **VTYP**, page D-19).
PARVER	VP1(10)	First parameters of the basic variable probability distributions. Should be supplied with input data according to the probability distributions used (see App. D). Input data.
	VP2(10)	Second parameters of the basic variable probability distributions. Should be supplied with input data according to the probability distributions used (see App. D). Input data.
	VP3(10)	Probability normalization factors to account for upper and lower limit truncation. Calculated in sub. **EINGAB**, page G-3).
	IVER(10)	Integer indicating the probability distribution options (see Table XII , page D-3). Input data.
GRENZE	UG(10)	Basic variable lower limits. Input data.
	OG(10)	Basic variable upper limits. Input data.

is created by this routine. It also calls the failure function options libraries (App. E). The **BABLVG** provides the partial derivatives of the failure function. This may be achieved

Table X Failure function related variables

COMMON block	Variable	Description
GEOM	BREIT	Characteristical length dimension. Used for the determination of geometry functions for the failure function determination. Input data.
	DICK	Tube wall thickness. Used for the determination of geometry functions for the failure function determination. Input data.
	RADIUS	Tube radius. Used for the determination of geometry functions for the failure function determination. Input data.
OPT	IRWOPT	Switch for crack propagation law option. **1** Paris, eq. (29), **2** Walker, eq. (30) and **3** Forman, eq. (32). Implemented in sub. **RWACHS**, page F-3.
	IKOPT	Switch for stress intensity factor option (Table I). Implemented in functions **YA** and **YB**, pages E-8 and E-8, respectively.
	IPGOPT	Switch for plastic limit load option (Table II). Implemented in func. **PG**, page E-7.
	IVKRIT	Switch for failure mode option: **1** brittle fracture (eq. (23)), **3** ductile fracture (eq. (25)), **2** two criteria approach (eq. (27)). Implemented in sub. **BGVONX**, page E-3.
SPAN	SIGMA	Applied stress level. Linked in sub. **BGVONX** (page E-3) to the 5th basic variable.
	SIGMAF	Flow stress value. Linked in sub. **BGVONX** (page E-3) to the 4th basic variable.
	SIGMAB	Bending stress. Input data.
	SIGV	Not implemented.
	QUOB	The ratio of the bending to membrane stress. Input data.

through numerical differentiation or by user supplied closed form derivatives. Usually, both routines are required to be changed if (1) the number or the meaning of the basic variables is to be changed, (2) another type of failure function is to be applied and (3) new crack propagation law is supplied.

A.6 Crack propagation library

This library provides a means of assessing the time dependent effects of the operational conditions. In particular, the initial crack length is extended by a value supplied by one of the implemented crack growth laws (App. F) according to the number of applied load cycles. A numerical differential equation solver is an essential part of this library. New crack propagation laws may be introduced by replacement or changes in the library.

Table XI Crack propagation related variables

COMMON block	Variable	Description
RISSW	**DLASTW(20)**	Number of load cycles per loadcase per year. Input data.
	LASTSP	Number of loadcases. Input data.
	NJAHRE	Number of years to operate. Input data.
FUNK	**CRISSW**	Constant *C* for the crack growth laws. Linked to the 6th basic variable (sub. **BGVONX**, page E-3).
	ENRISS	Exponent *n* for the crack growth laws. Input data.
	RSIGMA(20)	Ratios between the minimal and maximal stresses applied in each loadcase.
	DSIGMA(20)	Fractions of the absolute applied stress maximum achieved in each loadcase.
	DKTHR	Stress intensity factor threshold value. Input data.
	AKIC	Critical stress intensity factor value. Linked to the 3th basic variable (sub. **BGVONX**, page E-3).
	IDSIG	Loadcase counter in sub. **RWACHS**, page F-3.

A.7 User defined routines

All routines constituting failure function and crack growth calculations may be regarded as *user defined*. Additionally, user may want to supply new distribution functions. The recommendations on writing or modifying user defined routines are given in App. H.

APPENDIX B: *FIRST-ORDER RELIABILITY METHOD LIBRARY*

Program ZERBERUS

Description:

The main program is controlling the execution of input processing and first- and second-order reliability calculations. Additionally, the input and output channels are opened and closed.[1]

Parameters IN:

None.

Parameters OUT:

None.

External Subroutines:

GETARG[1] returns the values of the arguments following the MS-DOS command. The first argument, if exists, is used as the input/output filename. If not, the code requests the name to be typed via input channel (terminal). For further information see Microsoft FORTRAN Reference, Version 5.0.■[2]

DUEBER prints program header to the output channel.■

EINGAB reads and processes input data. **User defined** routine (see page G-3).

DIRTRA first-order calculations, including sensitivity factors (page B-4).

SORM performs second-order reliability calculations (see page C-3).

External Functions:

NARG[1] returns the number of arguments following the MS-DOS command. Used in the determination of the input/output file name from the first command line argument. For further information see Microsoft FORTRAN Reference, Version 5.0.■

Local variables:

IARG[1] number of arguments in the MS-DOS command line.

IFILE[1] counter in the loop determining the length of the file path and name.

INFILE[1] name and path of the input file. The .dat extension is assumed.

OUTFILE[1] name and path of the output file. The .out extension is assumed.

[1] PC version only.

[2] ■ indicates that no further information is given in this report.

Subroutine DIRTRA

Description:

This subroutine performs first-order reliability calculations. The general algorithm applied is as follows:

1. Transform the initial values XMAT(I) from physical to normal space (UMAT(I), subroutine **BUMAT**, page B-6).
2. Store normal space initial values UMAT(I) as "new" design point UNMAT(I).
3. Calculate the reliability index DELTA at the "new" design point (subroutine **BDELTA**, page B-7).
4. Calculate the failure function gradient AMAT(I) in the physical space (sub. **BAMAT**, page B-8).
5. Calculate the failure function gradient BANMAT(I) in the normal space (sub. **BANMAT**, page B-9).
6. Repeat the sequence 6.1 to 6.9 until the convergence conditions are satisfied:
   - 6.1 Replace the old normal space design point coordinates UMAT(I) with new ones (UNMAT(I)).
   - 6.2 Print the coordinates of current design point (sub. **DALLES**).
   - 6.3 Print current value of the reliability index DELTA (sub. **DDELTA**).
   - 6.4 Determine new iterative design point UNMAT(I) (normal space, sub. **BUNMAT**, page B-10).
   - 6.5 Transform the new design point to the physical space (XMAT(I), sub. **BXMAT**, page B-11).
   - 6.6 Calculate new reliability index (as in step 3. above).
   - 6.7 Calculate failure function gradient in physical space (as in step 4. above).
   - 6.8 Calculate failure function gradient in normal space (as in step 5. above).
   - 6.9 Check the convergence of the algorithm (sub. **ABKRIT**, page B-12).
7. Replace the old design point coordinates with new ones (as in step 6.1 above).
8. Print the final design point (sub. **DALLES**).
9. Calculate and print the failure probability (sub. **DRUAW**, page B-13).
10. Calculate and print joint probability density at the design point (subs. **BWAHRD**, page B-14, and **DWAHRD**, respectively).
11. Calculate and print sensitivity factors (sub. **SENSIT**, page B-15).

Parameters IN:

None.

Parameters OUT:

None.

External Subroutines:

BUMAT transforms current design point coordinates from physical to normal space (page B-6).

BDELTA calculates the reliability index for current design point (page B-7).

BAMAT partial derivatives of failure function in physical space (page B-8).

BANMAT partial derivatives of failure function in normal space (page B-9).

DALLES printout of the current design point calculated in the current iteration. ■

BUNMAT determines next design point in normal space (page B-10).

BXMAT transforms the design point from normal to physical space (page B-11).

DZUSAE[3] prints some additional information about values of basic variables, failure function and its derivatives. ■

ABKRIT decides whether to continue the iterative design point determination process or not (page B-12).

DRUAW calculates and prints the failure probability value (page B-13).

BWAHRD calculates the joint probability density at the design point (page B-14).

DWAHRD prints the joint probability density at the design point. ■

SENSIT calculates sensitivity factors (page B-15).

External Functions:
None.

Local variables:

I counter for the loop over basic variables.

HILF CHARACTER*10 control variable. Its value of this variable can be set to the value DURCHLAUF or ERGEBNIS, for the current or final design point iteration, respectively. It controls the output from the **DALLES** routine, described on page B-5.

ABBRU CHARACTER*10 control variable. Its value is set in routine **ABKRIT** (see Page B-12) to either WEITER or ABBRECHEN, meaning continue or stop the iterative design point determination, respectively.

WAHRDI variable containing the joint probability density calculated in subroutines **BWAHRD** and printed in **DWAHRD**.

[3] In the standard version of the code the subroutine call is disabled. However, its use is highly recommended during the application development phase.

Subroutine **BUMAT (XMAT, UMAT)**

Description:

This subroutine transforms the values of the physical variables XMAT(I) to the normal space variables UMAT(I) (see eq. (6)).

Parameters IN:

XMAT vector of basic variable values. In the first call of subroutine **BUMAT**, the initial values for the design point determination are transformed.

Parameters OUT:

UMAT vector of basic variables transformed into the normal space. The vector is used in the design point determination algorithm.

External Subroutines:

None.

External Functions:

BFIKUM cumulative distribution function in the physical space. See description on page D-6.

PHIINV inverse of the cumulative standard normal distribution. See description on page D-6.

Local variables:

I counter for the loop over all basic variables.

Subroutine **BDELTA**

Description:

This subroutine calculates the value of the reliability index (see eq. (3)). The calculation is performed in the normal (*u*) space, therefore a square root of sum of squared coordinates of the design point provides result identical to eq. (3) up to the sign, which is determined just before printing the failure probability in subroutine **VORZ** (page B-13). The reliability index value is saved in variable **DELTA**.

Parameters IN:

None.

Parameters OUT:

None.

External Subroutines:

None.

External Functions:

None.

Local variables:

I	counter for the loop over basic variables.

Subroutine **BAMAT (XMAT)**

Description:

The partial derivatives of the failure function in point XMAT(I) are calculated (eq. (4)). Results are saved in vector AMAT(I).

Parameters IN:

XMAT vector containing the coordinates of the point where the partial derivatives are to be calculated.

Parameters OUT:

None.

External Subroutines:

BABLVG this subroutine supplies the value of partial derivative of the failure function with respect to the Ith basic variable (physical space, see page E-5). **User defined subroutine.**

External Functions:

None.

Local variables:

I counter for the loop over basic variables.

Subroutine BANMAT

Description:

Calculates partial derivatives of the failure function in the normal space (eq. (14)), using the physical space derivatives obtained by subroutine **BAMAT** (page B-8) according to eq. (4). The results are stored in vector **ANMAT(I)**.

Parameters IN:

None.

Parameters OUT:

None.

External Subroutines:

None.

External Functions:

BFI	probability density function in the physical space (page D-5).
BFIKUM	cumulative distribution function in the physical space (page D-6).
PHI	standard normal probability density function (page D-17).
PHIKUM	cumulative standard normal distribution (page D-18).

Local variables:

I	counter for loop over basic variables.

Subroutine **BUNMAT (XMAT)**

Description:

The coordinates of the new design point UNMAT(I) are determined in normal space (eq. (8)), using the value of the failure function in physical space according to the eq. (13).

Parameters IN:

XMAT coordinates of the old design point approximation in physical space.

Parameters OUT:

None.

External Subroutines:

BGVONX value of the failure function in physical space (page E-3). **User defined subroutine.**

DZUSAE prints some additional information about values of basic variables, failure function and its derivatives. ■

External Functions:

None.

Local variables:

FM1, FM2, FM3 CHARACTER*40 variables, containing error messages.

SUM1, SUM2 intermediate results of various summations.

I counter for loop over basic variables.

G value of the failure function.

Subroutine **BXMAT (XMAT, UMAT)**

Description:

Transforms the design point UMAT(I) from normal to physical space (XMAT(I)) by inverting eq. (6).

Parameters IN:

UMAT coordinates of the design point in normal space.

Parameters OUT:

XMAT coordinates of the design point in physical space.

External Subroutines:

None.

External Functions:

PHIKUM cumulative standard normal distribution (page D-18).

BINKUM inverse of cumulative distribution in physical space (page D-7).

Local variables:

I counter for loop over basic variables.

Subroutine **ABKRIT (DMTNEU, DMTALT, ABBRU)**

Description:

This subroutine checks the convergence of the iterative design point determination algorithm. The convergence conditions are satisfied (1) if all of the design point coordinate changes **DMTNEU(I)-DMTALT(I)** posses an absolute value of less than 10^{-20}, **OR**, (2) if all of the design point coordinate changes represent less or equal to **ABBRGR** % of the coordinate values itself.

Parameters IN:

DMTNEU the coordinates of the new design point.

DMTALT the coordinates of the previous (old) design point.

Parameters OUT:

ABBRU CHARACTER*10, possessing the values ABBRECHEN or WEITER, indicating whether the convergence criteria are satisfied or not, respectively.

External Subroutines:

None.

External Functions:

None.

Local variables:

I counter for the loop over basic variables.

DIFF storing the result of **DMTNEU(I)-DMTALT(I)**.

Subroutine **DRUAW**

Description:

The failure probability estimate is calculated from the reliability index (eq. (2)) and printed.

Parameters IN:

None.

Parameters OUT:

None.

External Subroutines:

VORZ determines the sign of the reliability index. The reliability index calculation in sub. **BDELTA**, page B-7, is exact up to the sign (eq. (3)). The sign is determined just before the printing of the failure probability in this subroutine. ■

External Functions:

PHIKUM cumulative standard normal distribution (page D-18).

Local variables:

AUSFW failure probability.

Subroutine **BWAHRD (YMAT, WAHRDI)**

Description:

Calculates joint probability density in the point YMAT(I) (in physical space). The result is saved in WAHRDI for subsequent printing in subroutine **DWAHRD** (page B-5).

Parameters IN:

YMAT the coordinates of the design point.

Parameters OUT:

WAHRDI joint probability density values in point YMAT(I).

External Subroutines:

None.

External Functions:

BFI probability density function (in physical space, see page D-5).

Local variables:

I counter for loop over basic variables.

Subroutine SENSIT

Description:
Calculates and prints sensitivity factors according to eq. (15).

Parameters IN:
None.

Parameters OUT:
None.

External Subroutines:
None.

External Functions:
None.

Local variables:

I	counter for loop over basic variables.
ALFA	vector of sensitivity factors.
ANORM	normalization variable, length of **ANORM(I)** (eq. (15)).

APPENDIX C: *SECOND-ORDER RELIABILITY METHOD LIBRARY*

Subroutine SORM

Description:

This subroutine performs second-order reliability calculations. The main curvatures of the failure surface in the design point are calculated (eq. (22)) to allow for quadratic failure surface approximation. For this reason, SORM calculations can not start before the design point is determined. In general, the following tasks are performed in SORM:

1. The matrix **BGXMAT(10, 10)**[4] of second partial derivatives of the failure function is calculated in physical space (sub. **BGXMAT**, page C-5).
2. The second derivatives are determined also in normal space, forming matrix **BGUMAT(10, 10)** in sub. **BGUMAT**, page C-6.
3. A unit vector **ALFA(10)** directed from the origin to the design point is determined in normal space (sub. **BALFA**, page C-8).
4. The orthonormal basic vectors **YMAT(10, 10)** of the rotated normal (*y*) space are determined, **ALFA(10)** being the last ($N^{th} \leq 10$) of them (sub. **ORTHO**, page C-9).
5. The second derivative matrix is calculated into (*y*) space (**GYMAT(10, 10)**, sub. **BGYMAT**, page C-10).
6. Main curvatures **AKAPPA(10)** are determined based on the failure function derivatives in rotated normal (*y*) space (sub. **KRUMM**, page C-11).
7. Finally, the failure probability approximation **PTVEDT** is obtained in sub. **TVEDT**, page C-13.

Parameters IN:

None.

Parameters OUT:

None.

External Subroutines:

DUBERS prints program header to the output channel. ■

BGXMAT calculates the second derivatives of the failure function in the physical space (page C-5).

BUMAT transforms current design point coordinates from physical to normal space (page B-6).

BGUMAT calculates the failure function second derivatives in normal space (page C-6).

BALFA calculates the unit vector directed from the normal space origin towards the design point (page C-8).

ORTHO uses the unit vector calculated in **BALFA** as the last (N^{th}) basic vector to

[4] The value 10 is maximum field dimension of the current code version, limiting the number of basic variables to 10.

construct other (N-1) basic vectors of the (rotated normal *y*) space (page C-9).

BGYMAT calculates the second failure function derivatives in the rotated normal space (page C-10).

KRUMM calculates main curvatures of the failure surface at the design point (page C-11).

TVEDT uses main curvatures to determine the SORM failure probability approximation (page C-13).

Local variables:

A(10, 10) space provided for the eigenvalue solution in subroutine **KRUMM.**

D(10) space provided for calculated eigenvalues in sub. **KRUMM**.

E(10) working space for the eigenvalue solution.

M number of the eigenvalues, generally N-1.

Subroutine **BGXMAT (N, XMAT)**

Description:

The matrix GXMAT(10,10) containing second derivatives of the failure function is composed. The derivatives are obtained numerically (sub. **HESSE**, page C-14) in physical space.

Parameters IN:

N number of basic variables.

XMAT(10) design point coordinates in physical space.

Parameters OUT:

None.

External Subroutines:

None.

External Functions:

HESSE numerical second derivative determination using the Romberg[5] method (page C-14).

Local variables:

K, L counter of loops over basic variables.

[5] see **Formelsammlung zur Numerischen Mathematik mit Standard-FORTRAN-Programmen** by G.Jordan-Engeln & F.Reutter

Subroutine BGUMAT (N, UMAT, XMAT)

Description:

Subroutine determines the failure function second derivatives **GUMAT(10,10)** in the normal (*u*) space.

Parameters IN:

N number of basic variables.

UMAT(N) design point coordinates in normal space.

XMAT(N) design point coordinates in physical space.

Parameters OUT:

None.

External Subroutines:

None.

External Functions:

PHIKUM cumulative standard normal distribution function (page D-18).

BINKUM inverse of cumulative distribution in physical space (page D-7).

BFI probability density function in the physical space (page D-5).

DNORMV normal density function (page D-11).

BFISTR first derivative of the probability density in physical space (page D-8).

BABLVG this subroutine supplies the value of partial derivative of the failure function with respect to the I^{th} basic variable (physical space, see page E-5). **User defined subroutine.**

Local variables:

DNTPHI temporary saves the cumulative standard normal distribution.

DNVKUM temporary saves the cumulative (physical) distribution.

FI temporary saves the (physical) probability density.

PHI1 temporary saves the standard normal probability density.

GK temporary saves the K^{th} component of the failure function gradient (physical space).

K, L counters of loops over basic variables.

DMY0, SIG1 mean and standard deviation, set to 0. and 1., respectively. Used to obtain the standard normal probability density from routine **DNORMV**, page D-11.

D2XDU2(10) second derivatives of x with respect to u.

IND CHARACTER*10 variable, containing text ABL1, thus requesting the value of the first probability density derivative to be returned from distribution routines (see App. D).

D2XDU temporary result in calculating second derivatives of x with respect to u.

DXDU(10) first derivative of x with respect to u.

Subroutine **BALFA (N, UMAT)**

Description:

The unit vector **ALFA(10)** in normal (*u*) space, which is directed form the origin to the design point, is calculated. This vector represents the Nth basic vector of the rotated normal (*y*) space.

Parameters IN:

N number of basic variables.

UMAT(10) coordinates of the design point in normal space.

Parameters OUT:

None.

External Subroutines:

None.

External Functions:

SKALAR performs dot product of two vectors. Both vectors and their length are the arguments of this function.■

Local variables:

DNORM temporary saves the dot product.

L counter for loop over basic variables.

Subroutine **ORTHO(N)**

Description:

Based on vector **ALFA(10)**, obtained by routine **BALFA**, page C-8, the other N-1 basic vectors of the rotated normal space are calculated, following the orthonormalization principle. The results are saved in **YMAT(10, 10)**.

Parameters IN:

N number of basic variables (problem dimension).

Parameters OUT:

None.

External Subroutines:

None.

External Functions:

SKALAR performs dot product of two vectors. Both vectors and their length are the argument of this function. ■

Local variables:

DNORM	normalization factor for the new basic vector.
YNEU(10)	temporary saves new basic vector.
I, J, L	counters for loops over basic variables.
K	counter of the basic vectors.
AKJ	temporary saves dot product.
X1MAT(10, 10)	identity matrix, used in orthonormalization.
HILFX(10)	auxiliary variable.
HILFY(10)	auxiliary variable.

Subroutine **BGYMAT (N, YMAT)**

Description:

Calculates the second derivatives of failure function in the rotated normal (y) space. Results are stored into **GYMAT(10, 10)**.

Parameters IN:

N	number of basic variables.
YMAT(10, 10)	transformation matrix.

Parameters OUT:

None.

External Subroutines:

None.

External Functions:

None.

Local variables:

I, J, K, L	loop counters.
TMAT(10)	auxiliary variable.
HMAT(10)	auxiliary variable.
GMAT	temporary saves the second derivative value.

Subroutine KRUMM (M, A, D, E, XMAT)

Description:

The main curvatures AKAPPA(10) of the failure surface in design point are calculated (eq. (22)).

Parameters IN:

M number of curvatures to be found (N-1).

A(M, M) storage for matrix to be solved for eigenvalues.

D(M) storage for eigenvalues.

E(M) working field for the eigenvalue solver routine.

XMAT(10, 10) design point coordinates (physical space).

Parameters OUT:

None.

External Subroutines:

SYMQR the eigenvalue solver. Includes subroutine **ROTATE.** ■

External Functions:

BGYN calculates the partial derivative of failure function with respect to y_n, say $\partial g(y^*)/\partial y_n$ (page C-16).

Local variables:

N number of basic variables.

I, J counters for loops over basic variables.

IVEC control variable for the **SYMQR** eigenvalue solver. Set to 0 to disable eigenvectors calculation.

EPS convergency tolerance control variable for **SYMQR** eigenvalue solver. Set to 10^{-10}.

IABSCN relative convergence control variable for the **SYMQR** eigenvalue solver. Set to 0.

ITRD control variable for the **SYMQR** eigenvalue solver. Set to 0 meaning that matrix to be processed is not tridiagonal.

JOBN control variable for the **EIGRS**[6] eigenvalue solver. Set to 10, thus requesting processing of full matrix without calculation of eigenvectors.

DK0 parameter for the **SYMQR** eigenvalue solver.

GYN temporary saves the value of the partial derivative of failure function with respect to y_n.

[6] different eigenvalue solvers may be applied. This one is available in IMSL library, but unfortunately not in source code, jeopardizing code transfer to other computer types. SYMQR solver is available in source code and implemented in the current code version.

Subroutine **TVEDT**

Description:

The SORM failure probability approximation **PTVEDT** is calculated in this routine (eqs. (16), (17), (18), (19)). The value of failure probability is printed together with its contributions (see eq. (16)) and main curvatures **AKAPPA(10)**.

Parameters IN:

None.

Parameters OUT:

None.

External Subroutines:

None.

External Functions:

PROD subroutine for product calculation (page C-17).

BRLT calculates real part of the product of complex numbers (page C-18).

PHIKUM cumulative standard normal distribution function (page D-18).

PHI standard normal probability density function (page D-17).

Local variables:

PROD1 the product over main curvatures for the asymptotic failure probability (eq. (17)).

PROD2 the product over main curvatures for the first failure probability correction term (eq. (18)).

PROD3 the product over main curvatures for the second failure probability correction term (eq. (19)).

SUM1 asymptotic failure probability (eq. (17)).

SUM2 first failure probability correction term (eq. (18)).

SUM3 second failure probability correction term (eq. (19)).

Double precision function **HESSE (X, K, L, N)**

Description:

Numerical determination of the second derivatives of failure function, using Romberg method (see footnote 5, page C-5). The choice of the initial step H may be very important, because a small value of H may cause numerical errors to dominate or, large value of H may not supply correct derivative value. In principle, the central difference (X+H, X-H) scheme is employed, dividing the step H by 2. until convergence is achieved. **HESSE** may use numerically determined first derivatives, as obtained from sub. **BABLVG** E-5.

Parameters IN:

X(10) the coordinates of the point in physical space, where the derivatives are to be calculated.

K the index of the derivative, namely $\partial^2 g/\partial x_k \partial x_l$.

L the index of the derivative, namely $\partial^2 g/\partial x_k \partial x_L$.

N number of basic variables.

Parameters OUT:

None.

External Subroutines:

None.

External Functions:

BABLVG this subroutine supplies the value of partial derivative of the failure function with respect to the Ith basic variable (physical space, see page E-5). **User defined subroutine.**

Local variables:

D(0:4, 0:4) Romberg scheme matrix, containing the values of derivatives at different H values.

H initial step value. As discussed in the routine description above, care must be taken when choosing its initial values. They may be different for different basic variables.

I counter for loop over basic variables.

J counter for the Romberg iterations.

LA, LB counters for the loops providing printing of the Romberg scheme. Used only if convergency is not achieved.

M allowable number of Romberg iterations (J = 0, 1 ,.., M).

XPLUS(10) vector containing upper point for the central difference scheme, namely XPLUS(I) = X(I) + H.

XMINUS(I) vector containing lower point for the central difference scheme, namely XMINUS(I) = X(I) - H.

EPS convergence criterion for the Romberg iterations.

HMAT(10) vector of the H steps for basic variables. Only the Lth member differs from 0.

POT auxiliary variable.

Double precision function **BGYN (N, X1MAT)**

Description:

Calculates the failure function derivative with respect to the last coordinate of the rotated normal space (basic vector ALFA(10)). First, the failure function gradient in physical space is determined, being transformed to the normal space and dot multiplied by **ALFA.**

Parameters IN:

N number of basic variables.

X1MAT(10) coordinates of the point where the derivative is to be calculated (in physical space).

Parameters OUT:

None.

External Subroutines:

None.

External Functions:

PHIKUM cumulative standard normal distribution function (page D-18).

BINKUM inverse of cumulative distribution in physical space (page D-7).

BFI probability density function in the physical space (page D-5).

PHI standard normal probability density function (page D-17).

BABLVG this subroutine supplies the value of partial derivative of the failure function with respect to the Ith basic variable (physical space, see page E-5). **User defined subroutine.**

Local variables:

GYN temporary variable in determining derivative.

GRAD temporary value of the failure function (normal space) gradient component.

DNVKUM temporary value of the cumulative distribution (physical space).

DNTPHI temporary cumulative standard normal distribution value.

I counter for loop over basic variables.

FI temporary probability density value (in physical space).

FK temporary standard normal density value divided by **FI**.

Double precision function **PROD**

Description:

This functions calculates the products with different combinations of β and κ_j, as stated in eqs. (17) and (18).

Parameters IN:

DELTA reliability index.

AKAPPA(10) main curvatures of the failure surface in design point.

Parameters OUT:

None.

External Subroutines:

None.

External Functions:

None.

Local variables:

J counter for loop over all main curvatures.

PROD1 temporary auxiliary variable for determining the product.

Double precision function **BRLT (DELTA, AKAPPA)**

Description:

Calculates the real part of the complex product involving β and κ_j (eq. (19)).

Parameters IN:

DELTA reliability index.

AKAPPA(10) main curvatures of the failure surface in design point.

Parameters OUT:

None.

External Subroutines:

None.

External Functions:

None.

Local variables:

A(9) real components of the complex product.

B(9) imaginary components of the complex product.

J counter for loops over main curvatures.

R temporary auxiliary variable.

RE temporary auxiliary variable.

ARG complex number argument.

DIMAG temporary auxiliary variable.

QU temporary auxiliary variable.

BETRAG square of the complex product absolute value.

REALT real part of the complex product.

APPENDIX D: *PROBABILITY DISTRIBUTION LIBRARY*

A variety of the probability distribution functions have been already implemented in the present version of **ZERBERUS** code, as listed in Table XII . All of them are designed to provide the values of the probability density f(x), its first derivative with respect to x d*f(x)*/d*x*, cumulative probability *F(x)* and its inverse $F^{1}(x)$. Additionally, a set of four functions is provided (Table XIII) to access the above described features.

Further, approximations are provided for functions without closed form. In particular, this is necessary for the cumulative standard normal probability (**PHIKUM**, page D-18), error function **DERF**[7], gamma function **DGAMMA**[8] and incomplete gamma function **GAMMP**[7].

Table XII Distribution functions

IVER	Distribution type	Routine
1	Exponential distribution	**DEXPOV**, page D-9.
2	Lognormal distribution	**DLOGNV**, page D-10.
3	Normal distribution	**DNORMV**, page D-11
4	Weibull distribution	**DWEIV**, page D-12.
5	Gamma distribution	**DGAMMV**, page D-13.
6	Uniform distribution	**DGLEIV**, page D-14.
7	Gumbel distribution of the largest extreme	**DGUMAV**, page D-15.
8	Constant value	Not implemented. A better way is to implement the constant value directly in the failure function definition, sub. **BGVONX**, page E-3.
9	Gumbel distribution of the smallest extreme	**DGUMIV**, page D-16.

The inverse cumulative normal and gamma distributions can not be obtained in closed form. Newton-Raphson iterative search is provided in both cases (**PHIINV**, page D-7, and **GAMINV**, page D-7).

[7] From **Numerical Recipes** by W.H.Press, B.P.Flannery, S.A.Teukolsky and W.T.Vetterling.

[8] From **Handbook of Mathematical Functions** by Abramowitz and Stegun.

Table XIII Distribution access routines

Routine, reference	Description
BFI, page D-5	probability density
BFIKUM, page D-6	cumulative probability distribution
BINKUM, page D-7	inverse cumulative probability distribution
BFISTR, page D-8	first derivative of probability density

Double precision function **BFI (I, X)**

Description:

This function returns the probability density of I^{th} basic variable in point X. Appropriate scaling is employed in cases of truncated distributions.

Parameters IN:

I number of basic variable.

X point where the probability density is to be supplied.

Parameters OUT:

None.

External Subroutines:

None.

External Functions:

DEXPOV	exponential distribution (page D-9).
DLOGNV	lognormal distribution (page D-10).
DNORMV	normal distribution (page D-11).
DWEIV	Weibull distribution (page D-12).
DGAMMV	gamma distribution (page D-13).
DGLEIV	uniform distribution (page D-14).
DGUMAV	Gumbel distribution of the largest extreme (page D-15).
DGUMIV	Gumbel distribution of the smallest extreme (page D-16).

Local variables:

P1 the first distribution parameter.

P2 the second distribution parameter.

P3 scaling factor (see **EINGAB**, page G-3)

HILF CHARACTER*10 set to DICHTE, thus requiring called functions to return probability density.

Double precision function **BFIKUM** (I, X)

Description:

This function returns the cumulative distribution of **I**th basic variable in point **X**. Appropriate scaling is employed in cases of truncated distributions.

Parameters IN:

I number of basic variable.

X point where the cumulative distribution is to be supplied.

Parameters OUT:

None.

External Subroutines:

None.

External Functions:

DEXPOV exponential distribution (page D-9).

DLOGNV lognormal distribution (page D-10).

DNORMV normal distribution (page D-11).

DWEIV Weibull distribution (page D-12).

DGAMMV gamma distribution (page D-13).

DGLEIV uniform distribution (page D-14).

DGUMAV Gumbel distribution of the largest extreme (page D-15).

DGUMIV Gumbel distribution of the smallest extreme (page D-16).

Local variables:

P1 the first distribution parameter.

P2 the second distribution parameter.

P3 scaling factor (see **EINGAB**, page G-3).

HILF CHARACTER*10 set to KUM, thus requiring called functions to return cumulative distribution.

GRU distribution lower limit. The cumulative distribution value at this point is 0.

DKUM temporary auxiliary variable.

Double precision function **BINKUM (I, X)**

Description:

This function returns the inverse cumulative distribution of Ith basic variable in point X. Appropriate scaling is employed in cases of truncated distributions.

Parameters IN:

I number of basic variable.

X value of the cumulative distribution, for which the value of x is to be determined.

Parameters OUT:

None.

External Subroutines:

None.

External Functions:

DEXPOV exponential distribution (page D-9).

DLOGNV lognormal distribution (page D-10).

PHIINV inverse cumulative standard normal distribution. No closed form exists. Newton-Raphson iteration scheme is employed to obtain the inverse. ■

DWEIV Weibull distribution (page D-12).

GAMINV inverse gamma distribution. No closed form exists. Newton-Raphson iteration scheme is employed to obtain the inverse. ■

DGLEIV uniform distribution (page D-14).

DGUMAV Gumbel distribution of the largest extreme (page D-15).

DGUMIV Gumbel distribution of the smallest extreme (page D-16).

Local variables:

P1, P2 the first and the second distribution parameter, respectively.

P3 scaling factor (see **EINGAB**, page G-3).

HILF CHARACTER*10 set to KUM. Called functions return cumulative distribution.

GRU distribution lower limit. The cumulative distribution value at this point is 0.

DINKUM temporary auxiliary variable.

ARG temporary auxiliary variable.

Double precision function **BFISTR (I, X)**

Description:

This function returns the first derivative of the probability density function of I^{th} basic variable in point X. Appropriate scaling is employed in cases of truncated distributions.

Parameters IN:

I number of basic variable.

X point where the probability density derivative is to be supplied.

Parameters OUT:

None.

External Subroutines:

None.

External Functions:

DEXPOV	exponential distribution (page D-9).
DLOGNV	lognormal distribution (page D-10).
DNORMV	normal distribution (page D-11).
DWEIV	Weibull distribution (page D-12).
DGAMMV	gamma distribution (page D-13).
DGLEIV	uniform distribution (page D-14).
DGUMAV	Gumbel distribution of the largest extreme (page D-15).
DGUMIV	Gumbel distribution of the smallest extreme (page D-16).

Local variables:

P1 the first distribution parameter.

P2 the second distribution parameter.

P3 scaling factor (see **EINGAB**, page G-3).

IND CHARACTER*10 set to ABL1, thus requiring called functions to return cumulative distribution.

Double precision function **DEXPOV (X, DLAMB, C, IND)**

Description:

Exponential distribution with the probability density:

$$f(x) = \lambda e^{-\lambda(x-c)}$$

Function calculates the cumulative distribution, probability density, first and second probability density derivatives, mean or variance, depending on switch **IND**.

Parameters IN:

X distribution function argument, point where the value should be calculated.

DLAMB first distribution parameter (λ in equation above).

C second distribution parameter (c in equation above).

IND CHARACTER*10 switch variable:

KUM	calculate cumulative distribution.
DICHTE	calculate probability density.
ABL1	first probability density derivative.
ABL2	second probability density derivative.
MIT	mean value.
VAR	variance.

Parameters OUT:

None.

External Subroutines:

None.

External Functions:

None.

Local variables:

None.

Double precision function **DLOGNV (X, AMUE, SIGMA, IND)**

Description:

Lognormal distribution with the probability density:

$$f(x) = \frac{1}{x\sigma\sqrt{2\pi}} \, e^{-\frac{1}{2}\left(\frac{\ln(x/\mu)}{\sigma}\right)^2}$$

Function calculates the cumulative distribution, probability density, first and second probability density derivatives, mean and variance, depending on switch **IND**.

Parameters IN:

X distribution function argument, point where the value should be calculated.

AMUE first distribution parameter (μ in equation above).

SIGMA second distribution parameter (σ in equation above).

IND CHARACTER*10 switch variable, see description on page D-9.

Parameters OUT:

None.

External Subroutines:

None.

External Functions:

PHIKUM cumulative standard normal distribution (page D-18).

Local variables:

DNORM auxiliary variable.

C1, C2 auxiliary variables.

Double precision function **DNORMV (X, AMUE, SIGMA, IND)**

Description:

Normal distribution with the probability density:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

Function calculates the cumulative distribution, probability density, first and second probability density derivatives, mean and variance, depending on switch **IND**.

Parameters IN:

X	distribution function argument, point where the value should be calculated.
AMUE	first distribution parameter (mean, μ in equation above).
SIGMA	second distribution parameter (standard deviation, σ in equation above).
IND	CHARACTER*10 switch variable, see description on page D-9.

Parameters OUT:

None.

External Subroutines:

None.

External Functions:

PHIKUM	cumulative standard normal distribution (page D-18).
PHI	standard normal probability density function (page D-17).

Local variables:

DNORM	auxiliary variable.

Double precision function **DWEIV (X, SIGMA, ETA, IND)**

Description:

Weibull distribution with the probability density:

$$f(x) = \frac{\eta}{\sigma^{\eta}} x^{\eta - 1} e^{-\left(\frac{x}{\sigma}\right)^{\eta}}$$

Function calculates the cumulative distribution, probability density, first and second probability density derivatives, mean and variance, depending on switch **IND**.

Parameters IN:

X distribution function argument, point where the value should be calculated.

SIGMA first distribution (scale) parameter (σ in equation above).

ETA second distribution (shape) parameter (η in equation above).

IND CHARACTER*10 switch variable, see description on page D-9.

Parameters OUT:

None.

External Subroutines:

None.

External Functions:

DGAMMA returns the value of the gamma function (see footnote 8 on page D-3).

Local variables:

E, Z auxiliary variables.

DKNEU auxiliary variable.

Double precision function **DGAMMV (X, BETA, ALPHA, IND)**

Description:

Gamma distribution with the probability density:

$$f(x) = \frac{\beta^{\alpha}}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x}$$

$$\Gamma(\alpha) = \int_{0}^{\infty} x^{\alpha-1} e^{-x} dx$$

Function calculates the cumulative distribution, probability density, first and second probability density derivatives, mean and variance, depending on switch **IND**.

Parameters IN:

X distribution function argument, point where the value should be calculated.

SIGMA first distribution (scale) parameter (β in equation above).

ETA second distribution (shape) parameter (α in equation above).

IND CHARACTER*10 switch variable, see description on page D-9.

Parameters OUT:

None.

External Subroutines:

None.

External Functions:

DGAMMA returns the value of the gamma function ($\Gamma(\alpha)$, see footnote 8 on page D-3).

GAMMP incomplete gamma function (see footnote 7 on page D-3).

Local variables:

TMP auxiliary variable.

Double precision function **DGLEIV (X, GRU, GRO, IND)**

Description:

Uniform distribution between **GRU** and **GRO**. Function calculates the cumulative distribution, probability density, first and second probability density derivatives, mean and variance, depending on switch **IND**.

Parameters IN:

X distribution function argument, point where the value should be calculated.

GRU lower limit of the interval.

GRO upper limit of the interval.

IND CHARACTER*10 switch variable, see description on page D-9.

Parameters OUT:

None.

External Subroutines:

None.

External Functions:

None.

Local variables:

None.

Double precision function **DGUMAV (X, AMUE, SIGMA, IND)**

Description:

Gumbel distribution of the largest extreme with the probability density:

$$f(x) = \frac{1}{\sigma} e^{-\frac{x-\mu}{\sigma} - e^{-\frac{x-\mu}{\sigma}}}$$

Function calculates the cumulative distribution, probability density, first and second probability density derivatives, mean and variance, depending on switch **IND**.

Parameters IN:

X distribution function argument, point where the value should be calculated.

AMUE first distribution (scale) parameter (μ in equation above).

SIGMA second distribution (shape) parameter (σ in equation above).

IND CHARACTER*10 switch variable, see description on page D-9.

Parameters OUT:

None.

External Subroutines:

None.

External Functions:

None.

Local variables:

Z auxiliary variable.

Double precision function **DGUMIV (X, AMUE, SIGMA, IND)**

Description:

Gumbel distribution of the smallest extreme with the probability density:

$$f(x) = \frac{1}{\sigma} e^{\left(\frac{x-\mu}{\sigma} - e^{\frac{x-\mu}{\sigma}}\right)}$$

Function calculates the cumulative distribution, probability density, first and second probability density derivatives, mean and variance, depending on switch **IND**.

Parameters IN:

X	distribution function argument, point where the value should be calculated.
AMUE	first distribution (scale) parameter (μ in equation above).
SIGMA	second distribution (shape) parameter (σ in equation above).
IND	CHARACTER*10 switch variable, see description on page D-9.

Parameters OUT:

None.

External Subroutines:

None.

External Functions:

None.

Local variables:

Z auxiliary variable.

Double precision function **PHI (X)**

Description:
Returns the standard normal probability density.

Parameters IN:
X distribution function argument, point where the value should be calculated.

Parameters OUT:
None.

External Subroutines:
None.

External Functions:
None.

Local variables:
Y temporary auxiliary variable.

Double precision function **PHIKUM** (X)

Description:

Returns the cumulative normal standard distribution. The function employs the approximations described in **Handbook of Mathematical Functions** by Abramowitz and Stegun, eqs. 26.2.11 and 26.2.12.

Parameters IN:

X distribution function argument, point where the value should be calculated.

Parameters OUT:

None.

External Subroutines:

None.

External Functions:

DERF error function. See footnote 7 on page D-3.

Local variables:

CWARNG CHARACTER*60 variable containing warning message.

I counter.

DINKR temporary auxiliary variable.

AKLAM temporary auxiliary variable.

Character*10 function **VTYP (I)**

Description:

Function returns the name of the probability distribution function according to the option number I. Option numbers of the available distributions are listed in Table XII on page D-3.

Parameters IN:

I variable containing the probability distribution function option number.

Parameters OUT:

None.

External Subroutines:

None.

External Functions:

None.

Local variables:

None.

APPENDIX E: *FAILURE FUNCTION LIBRARY*

Double precision function **BGVONX(X)**

Description:

The failure function is calculated. Essentially, the physical meaning is assigned to the basic variables in this routine. The following links between basic variables and physical quantities are assumed (see Chapter 4):

Basic variable:	Physical meaning:
1	crack depth a
2	crack depth to length ratio a/c
3	critical stress intensity factor K_{Ic}
4	flow stress σ_f
5	applied stress σ
6	constant C in crack propagation laws.

All remaining influencing parameters such as geometry are assumed constant and are supplied to the code by sub. **EINGAB**, page G-3. The links between basic and physical variables are coded in the first few executable lines of **BGVONX** in the following manner: on the equation left side there is always a physical variable while the right side always contains the basic variable. In order to expand or change the above links, the left hand names should be from the list in Table X on page A-4 to assure proper function of the existing failure mode and crack propagation libraries. Nevertheless, users may want to supply his own failure and crack propagation routines, which should be done inside of this subroutine.

Parameters IN:

X(10) coordinates of the point in physical space, where the failure function is to be calculated.

Parameters OUT:

None.

External Subroutines:

RWACHS calculates the stable crack growth (page F-3).

External Functions:

YA calculates the geometrical function of the stress intensity factor (page E-8).

YB calculates the geometrical function of the stress intensity factor (page E-8).

PG calculates the geometrical function of the plastic limit load stress (page E-7).

Local variables:

A0 initial crack depth value (a).

ADC0 initial crack depth to length ratio (a/c).

A crack depth including crack propagation.

ADC crack depth to length ratio including crack propagation.

KOMPGO temporary variable containing the value of the failure criterion switch **IVKRIT** (see Table X , page A-4).

G temporary variable saving the value of the failure function.

SR applied to limit stress ratio (two criteria approach).

AKR applied to critical stress intensity ratio (two criteria approach).

Double precision function **BABLVG (I, X)**

Description:

Provides the partial derivatives of the failure function. A numerical procedure (Romberg method, footnote 5, page C-5) is provided (see sub. **HESSE**, C-14). Some derivatives (with respect to critical stress intensity value, flow stress and applied stress) are supplied in closed form to speed up the code execution.

Parameters IN:

I the partial derivative with respect to Ith basic variable.

X(10) at point X.

Parameters OUT:

None.

External Subroutines:

RWACHS calculates the stable crack growth (page F-3).

External Functions:

BGVONX failure function calculation (page E-3).

YA calculates the geometrical function of the stress intensity factor (page E-8).

YB calculates the geometrical function of the stress intensity factor (page E-8).

PG calculates the geometrical function of the plastic limit load stress (page E-7).

Local variables:

D(0:4, 0:4) Romberg scheme matrix, containing the values of derivatives at different H values.

H initial step value. As discussed in the routine description above, care must be taken when choosing its initial values. They may be different for different basic variables.

I counter for loop over basic variables.

J counter for the Romberg iterations.

LA, LB counters for the loops providing printing of the Romberg scheme. Used only if convergency is not achieved.

M allowable number of Romberg iterations (J = 0, 1 ,.., M).

XPLUS(10) vector containing upper point for the central difference scheme, namely XPLUS(I) = X(I) + H.

XMINUS(I) vector containing lower point for the central difference scheme, namely XMINUS(I) = X(I) - H.

EPS convergence criterion for the Romberg iterations.

HMAT(10) vector of the **H** steps for basic variables. Only the **L**th member differs from 0.

POT auxiliary variable.

A0 initial crack depth value (*a*).

ADC0 initial crack depth to length ratio (*a*/*c*).

A crack depth including crack propagation.

ADC initial crack depth to length ratio including crack propagation.

KOMPGO temporary variable containing the value of the failure criteria switch **IVKRIT** (see Table X , page A-4).

ABL temporary variable saving the derivative of the failure function.

Double precision function **PG (A, ADC)**

Description:

Function calculates the geometrical function M for the determination of the limit load stress (see eq. (26)). The function value depends on the option control variable **IPGOPT**, which follows the description in Table II . Option numbers 8 and 15 are **user defined** options. User is required to supply appropriate M function algorithm if options 8 or 15 are specified.

Parameters IN:

A crack depth.

ADC crack depth to length ratio.

Parameters OUT:

None.

External Subroutines:

None.

External Functions:

None.

Local variables:

A variety of local variables exist in this function. All of them are auxiliary variables introduced to calculate different values leading to different M values, according to the proper option controlled by variable **IPGOPT**.

Double precision functions **YA (A, ADC)** and **YB (A, ADC)**

Description:

Both functions are used to provide the geometrical function *Y* in order to calculate stress intensity factors (eq. (24)). Function **YA** specializes in the crack depth direction, while func. **YB** provides the geometry functions at the surface points of a semi-elliptical 2-D crack, thus neglecting the one-dimensional options 1 to 7. The function value depends on the option control variable **IKOPT**, which follows the description in Table I . Option numbers 7 and 15 are **user defined** options. User is required to supply appropriate *Y* function algorithm if options 7 or 15 are specified.

Parameters IN:

A	crack length (1-D cracks) or crack depth (2-D cracks).
ADC	crack depth to length ratio (2-D cracks only).

Parameters OUT:

None.

External Subroutines:

None.

External Functions:

None.

Local variables:

A variety of local variables exist in this function. All of them are auxiliary variables introduced to calculate different values leading to different *Y* values, according to the proper option controlled by variable **IKOPT**.

APPENDIX F: *CRACK GROWTH LIBRARY*

Subroutine **RWACHS (A0, ADC0, A, ADC)**

Description:

The amount of crack growth is calculated in this routine. The crack growth equation (*da/dN*) is cumulatively integrated over all load cases and all operational years.

Parameters IN:

A0 initial crack depth.

ADC0 initial crack depth to length ratio.

Parameters OUT:

A final crack depth.

ADC final crack depth to length ratio.

External Subroutines:

ADAMS numerical differential equation solver utilizing Adams‘ method. Initial values are obtained form the **KUTTA** routine. The crack growth parameters are evaluated in subroutine *FUNC*, page F-4. ■

External Functions:

None.

Local variables:

IDM number of crack dimensions (1 or 2).

KOMPGO temporary switch variable containing crack growth law option number.

LAUF counter over operational years.

LAUF1 counter over load cases.

OG upper integral limit.

UG lower integral limit.

YNT(2) starting value of crack depth (depth to length ratio).

Y(2, 1001) values of crack depths (depth to length ratios) for all integration steps.

ER (1001) auxiliary variable in **ADAMS**.

IDSIG auxiliary counter of load cases.

MAXSTP maximum number of integration steps.

Subroutine **FUNC (YSV, X, IDM, F)**

Description:

Returns the right hand side of crack growth eqs. (29), (30) and (32), depending on the value of switch variable **IRWOPT** (1, 2 and 3, respectively).

Parameters IN:

YSV(2) crack depth and crack length ratio.

X not implemented.

IDM number of crack dimensions (1 or 2).

Parameters OUT:

F(2) right hand side of crack growth equations.

External Subroutines:

None.

External Functions:

YA calculates the geometrical function of the stress intensity factor (page E-8).

YB calculates the geometrical function of the stress intensity factor (page E-8).

Local variables:

KOMPGO auxiliary switch variable.

Other local variables are implemented as auxiliary variables in order to calculate different values for different crack growth law options.

APPENDIX G: *DATA PREPARATION*

Input data reading and processing is performed by subroutine **EINGAB**. The order of reading, meaning of variables and amount of processing, performed in this subroutine for a typical fatigue crack analysis are given below. Certain user changes may be necessary to use the creep failure and crack propagation options.

CARD # 1

N	IDRUCK

N number of basic variables. Currently limited to 10.
IDRUCK controls the amount of printed information:
0 print failure probability, sensitivity factors, initial and final design point value,
1 as 0 and reliability index for each iteration,
2 as 0 and design point value for each iteration.

CARD # 2

ABBRGR

ABBRGR convergence criterion for the design point determination algorithm. The iterative search is stopped when all of the design point changes represent less or equal of ABBRGR %. See also page B-12.

CARD # 3

IVER	VP1	VP2	UG	OG

This card should be repeated for all basic variables (N-times).

IVER the code of the distribution function. For list of implemented functions and their **IVER** values see Table XII , page D-3.

VP1, VP2 the parameters of the chosen distribution function. The distribution and the meaning of the parameters are further described in App. D.

UG, OG lower and upper limit of the applied distribution. Normalization of the distribution function is performed to satisfy:

$$\int_{UG}^{OG} f(x)dx = 1$$

CARD # 4

NGPAR

NGPAR number of parameters describing failure function. Used only if the user knows the failure function in the form of response surface or similar.

CARD # 5

GP(1)	GP(2)	...	GP(NGPAR)

GP(NGPAR) parameters defining the parametric form of the failure function. A change of subroutine **BGVONX** (page E-3) is necessary to implement this feature.

CARD # 6

XAMAT(1)	XAMAT(2)	...	XAMAT(N)

XAMAT(N) initial values of basic variables. Subroutine **GRPRUF** checks if the initial values are between the lower and upper limit of distribution function.

CARD # 7

FM1	FM2	FM3

FM1, FM2, FM3 CHARACTER variables, containing the title of the problem to printed in the output file. Three lines containing up to 60 characters are allowed.

CARD # 8

BREIT	DICK	RADIUS

BREIT characteristic length dimension (e.g., plate width),

DICK tube wall thickness,

RADIUS tube inner radius.

CARD # 9

IKOPT	IPGOPT	IVKRIT	IRWOPT

IKOPT K factor calculation option. See Table I , page 5.

IPGOPT plastic limit load calculation option. See Table II , 6.

IVKRIT failure criterion option:
1 brittle fracture (linear elastic fracture mechanics, eq. (23), page 5),
2 two criteria approach (eq. (27), page 7),
3 plastic limit load (eq. (25), page 6).

IRWOPT crack propagation law option:
1 after Paris (eq. (29), page 8),
2 after Walker (eq. (30), page 8),
3 after Forman (eq. (32), page 8).

CARD # 10

LASTSP

LASTSP number of loadcases.

CARD # 11

SIGMAM	SIGMAB

SIGMAM membrane stress.

SIGMAB bending stress.

CARD # 12

DSIGMA(1)	DSIGMA(2)	...	DSIGMA(NLASTSP)

DSIGMA(NLASTSP) loading amplitudes normalized by SIGMAM. Absolute values are calculated within subroutine **EINGAB**.

CARD # 13

RSIGMA(1)	RSIGMA(2)	...	RSIGMA(NLASTSP)

RSIGMA(NLASTSP) ratios of stress minima and maxima.

CARD # 14

DLASTW(1)	DLASTW(2)	...	DLASTW(NLASTSP)

DLASTW(NLASTSP) number of load cycles per year of operation.

CARD # 15

NJAHRE

NJAHRE number of years of operation.

CARD # 16

DKTHR

DKTHR stress intensity factor threshold value.

CARD # 17

CRISS	ENRISS

CRISS crack growth parameter C.

ENRISS crack growth parameter n.

APPENDIX H: *USER DEFINED ROUTINES*

The structure of the ZERBERUS code assumes the analysis of the *standard* case. Thus, the following links of the basic and physical variables are predetermined in sub. **BGVONX** (page E-3):

Basic variable:	Physical meaning:
1	crack depth a
2	crack depth to length ratio a/c
3	critical stress intensity factor K_{Ic}
4	flow stress σ_f
5	applied stress σ
6	constant C in crack propagation laws.

In other words, this variables are assumed to vary stochastically and should be input as basic variables by repeating **Card #3** (App. G). All other relevant physical quantities are assumed to be constant. Further, the fatigue loading and crack propagation options are predefined.

The analysis of an arbitrary problem might require some changes in the above mentioned variable linking scheme. Further, it might require different loading description and crack propagation options. Another possible incompatibility is using probability distributions apart from those listed in Table XII on page D-3. All this *special* requirements may be easily accounted for with relatively simple code changes.

Typical changes of the ZERBERUS code are described in the following paragraphs. These changes were performed to calculate the results listed in Chapter 6.

H.1 Failure function

The links between basic and physical variables in **BGVONX** were defined as follows:

Basic variable:	Physical meaning:
1	tube mean radius R
2	tube wall thickness t
3	flow stress factor K
4	sum of stresses $(\sigma_Y + \sigma_M)$
5	operating temperature correction δ
6	measured crack length a_m
7	crack propagation a_p

Further, the **BABLVG** (page E-5) routine needed to be changed. It calculates the failure function partial derivatives with respect to basic variables. This routine can handle both closed form and numerical determination of derivatives. Both options are implemented in the *standard* case, assuming the *standard* meaning of basic variables and their typical magnitude. Again, the new order of basic variables must be defined.

Next problem may be non-existent geometry function for the stress intensity factor or limit stress calculation. The options already included are listed in Table I and Table II and allow for *user defined* option. However, user is required to supply appropriate FORTRAN coding to replace printing of message KEINE BENUTZER-OPTION FUER K-FAKTOREN IMPLIMENTIERT in routines **YA** and **YB** and KEINE BENUTZER OPTION FUER PLASTISCHE GRENZLAST IMPLEMENTIERT in routine **PG**. Further information may be found on pages E-7 and E-8. In certain cases, it may be useful to replace routines **YA**, **YB** and **PG** completely.

Additionally, input and output processing routines (**EINGAB**, page G-3, and **IDENT**, respectively) should be changed to allow for different loading data reading and printing.

H.2 Additional probability distributions

The probability distributions already implemented are listed in Table XII and further described App. D.
All routines are required to supply the following data:

1. **cumulative probability function**
2. **probability density**
3. **first derivative of probability density**
4. second derivative of probability density
5. mean
6. variance

The **ZERBERUS** code execution requests only first three values. The remaining are left in the code for historical reasons. The choice of value depends on the calling routine (see Table XIII). To implement new distribution, appropriate entries should be inserted in calling routines. Two potential problems may be foreseen:

1. there is no closed form of probability function. Numerical solver based on Newton-Raphson iteration scheme is supplied for this case and already implemented in **PHIINV** (page D-7) and **GAMINV** (page D-7).
2. care should be exercised in determining the normalized distribution values.