1996 OMAE — Volume II, Safety and Reliability
ASME 1996

PROBLEMS AND SOLUTION STRATEGIES IN RELIABILITY UPDATING

Roger Sindel
Institut für Tragwerksbau
Technische Universität München
München, Germany

Stephan Gollwitzer
RCP GmbH
München, Germany

Rüdiger Rackwitz
Institut für Tragwerksbau
Technische Universität München
München, Germany

ABSTRACT

Reliability updating by inspections has become widely used in the offshore industry. The theoretical concepts and basic numerical techniques are well established. Practical applications indicated that simple FORM estimates based on individual linearizations of the failure surface and the observation equations or observation inequalities are rather inaccurate. Full FORM/SORM techniques implying a search for the joint β-point need to be applied. However, certain algorithmic difficulties can occur. This is especially true for reliability updating in the presence of structural fatigue (crack propagation and instability) but the problem appears to exist whenever conditional probabilities are to be calculated. Most frequently the algorithms will fail because they iterate into physically or mathematically inadmissible domains. New and widely successful strategies to select appropriate starting points for the β-point search as well as strategies partly based on heuristic arguments to avoid physically inadmissible domains are proposed. These strategies are designed to work independently of the special search algorithm used. They are illustrated at a typical example in which it is also demonstrated that the simpler crude FORM approaches are inadequate.

INTRODUCTION

One of the most important applications of FORM/SORM methods in offshore technology is reliability updating using observations on variables or states of the system. The following equation states the formula for conditional probabilities or Bayes' rule

$$P(F \mid B) = \frac{P(F \cap B)}{P(B)} \tag{1}$$

in which F denotes the failure event and B the observation event, be it the event that the structure has survived up to the observation time, a crack length observation, wave height observations etc. or combinations and multiplicities thereof. Numerator and denominator are calculated separately. Especially for more complex events FORM/SORM methods have been successfully applied (see, for example, Madsen, 1985, and Fujita et al., 1989, or more recently Goyet et al., 1994). The computational details are described in Madsen et al. (1989), Gollwitzer and Rackwitz (1987), and Gollwitzer and Rackwitz (1988), where special emphasis is given to the case when observation events are given as equality constraints. The computational tasks in reliability updating by the classical methods of FORM and SORM consist of the search for the so-called joint β-point(s) both for the numerator and the denominator in (1) and the subsequent probability evaluation. While the latter only requires some simple algebraic manipulations and the computation of the multinormal integral (Gollwitzer and Rackwitz, 1988), the search of the joint β-point can be quite involved and can fail under conditions which are not at all unusual. For convenience, those computational tasks are all formulated in the transformed standard space (Hohenbichler et al., 1987). Given a set of constraints to be fulfilled simultaneously, the β-point is defined to be the one with the minimal distance from the coordinate origin to the limiting curve of the constraining domain, thus leading to the optimization problem

$$\begin{aligned} \|u^*\| &= min\{\|u\|\} && (2) \\ g_j(u) &= 0 \quad , \quad j = 1, ..., m_e && (3) \\ g_j(u) &\leq 0 \quad ; \quad j = m_{e+1}, ..., m && (4) \\ u_{low} \leq u &\leq u_{up} && (5) \end{aligned}$$

where u is the vector of basic variables, limited by its lower and upper bounds u_{low} and u_{up} respectively, u^* is the location vector of the joint β-point, and the $g_j(u)$ are equality and inequality constraints, respectively, which are assumed to be at least twice differentiable. $g_j(u)$ is denoted by state function in the sequel. Additionally, it is assumed that the optimization problem is unique, i.e. there exists only one such joint β-point. For simple component reliability it can be reduced to a single inequality constraint.

A variety of programs applying different concepts and algorithms are available to handle the optimization problem. One of the most sophisticated and efficient among them appears to be the NLPQL code (Schittkowski, 1985-1994) which is based on successive solutions of quadratic programming subproblems with subsequent one-dimensional line searches to determine the iteration step length with the help of a suitable merit function. During iteration the Hessian matrix of an augmented Lagrangian function will automatically be updated. However, many alternative algorithms exist which serve the purpose equally well in most cases.

There are certain well-known circumstances under which they can fail. In most cases this is due to taking numerical derivatives of the non-linear constraints. This problem can usually be solved by using more involved differentiation formulae. If the code still does not find a solution many computer codes then provide usually time consuming devices to enforce convergence if the underlying problem has a solution at all. These devices include modification of the numerical constants of the algorithm, presetting of the Hessian with values different from the unit matrix, different methods for the line searches but, first of all, appropriate starting points need to be selected. In some cases it may even be necessary to switch to another algorithm which is better suited to the particular optimization problem. If only inequality constraints are present convergence is usually achieved more easily than in case of equality constraints constituting a more difficult problem for most algorithms. All in all, if there is a solution to the optimization problem under the conditions mentioned above, it can be found. Here we wish only to emphasize that sometimes optimization even in the specialized form required by FORM/SORM can require some skill, experience and insider knowledge. Those convergence problems which are somehow inherent in any optimization are not dealt with herein.

Unfortunately and almost independent of the particular algorithm used, several other serious difficulties are met in practical applications. In the following some of these other sources of algorithmic instabilities will be discussed and appropriate solution strategies will be proposed. A typical example from the field of inspection and maintenance planning for ships will illustrate the effect of these solution strategies.

SOURCES OF NON-ALGORITHMICALLY BASED CONVERGENCE PROBLEMS

The domain defined by formulae (3) to (4) is denoted by the (algorithmically) feasible domain. It is assumed that the functions $g_j(u)$ also exist in the algorithmically infeasible domain but physically admissible domain, i.e. the safe domain. Usually the algorithm is started in this domain at some central value for which it is known that all constraint functions exist.

It is generally true that one constraint problems can be solved much more reliably. Therefore, a number of authors have proposed a simplified version of the rigorous scheme described before. The proposal is to individually linearize all constraints in their respective β-point. Then, the original problem of working in n-dimensional spaces with m non-linear constraints $(n \geq m)$ can be reduced to a problem in m dimensions with m linear constraints. The remaining problem then is to find the joint β-point of the linearizations and the solution of the m-dimensional multinormal integral. This method will be denoted by crude FORM in the following. However, depending on the shape of the state functions the final probability results for the numerator and/or the denominator in equation (1) can differ considerably from the more exact values obtained by the FORM and SORM methods. Moreover, the conditional probabilities can be in error by several orders of magnitude as can be demonstrated easily (see also the example later on). It is therefore mandatory to apply the rigorous theory. But this fails in some cases. The reason is primarily due to the fact that the state functions are not or cannot be properly formulated.

In fact, the most serious failure of any search algorithm occurs if during iteration a point is found which is mathematically and/or physically inadmissible and this can occur in spite of defining admissibility windows individually for the variables. For example, at some distance from the origin mathematical singularities may exist, root finding implemented in the state function results in complex solutions, determinants may become negative which are admitted only positive from physical considerations, etc.. Very disturbing is the fact, that the domain which is mathematically and/or physically admissible, as depicted in figure 1, can hardly be defined in advance in terms of additional constraints. Even if this is possible in special cases it can require enormous intellectual and numerical effort. If no provision is taken, the algorithm will either stop with a mathematical error or iterate in wrong directions and convergence to a valid solution cannot be achieved.

In the following we will discuss mainly three points to cope with the situation:

1. Constraint formulation: The performance of the optimization algorithm strongly depends on the mathematical formulation of the state functions. Often several constraint representations are possible, each influencing the β-point search in a different way.
2. Starting points: In many cases the selection of suitable starting points is vital for successful optimization.
3. Admissible domain: The shape of the admissible domain itself can be a reason for convergence problems. Depending on the state functions the optimization algorithm can have a more or less strong tendency to leave the admissible domain which usually leads to a fatal error during a computational run.

SOLUTION STRATEGIES

Constraint Formulation

The specific formulations of the state functions play an important role for the stability and efficiency of the algorithm. In most cases large gradients of the state functions are beneficial and some care must be taken to scale these functions appropriately. Also, the operation of differentiation in a state function can make function and gradient evaluation unstable whereas the operation of integration or summation has a smoothing effect. Considering for example crack propagation in a structural component several possibilities of representation are conceivable. One of these would be formulating the state functions in the so-called event space referring to the crack sizes involved. A second possibility is to perform reliability investigations in the time space by comparing expected observation or service times with certain target values. As a third alternative stresses related to the propagation process could be considered which leads to formulations in the stress space. Evidently, additional alternatives can be formulated each resulting in a different algorithm performance. For general reliability problems it is found that only tests can determine which way of constraint representation appears to be most advantageous.

Improved Starting Solutions

Each optimization algorithm requires a suitable initial value. The natural starting point is the mean or median of the basic variables vector which usually also checks the validity of the constraint formulations. At least some point should be selected which lies in the safe (infeasible but admissible) domain so that necessary presettings for the algorithm can be performed. It certainly is desirable to start already in the neighbourhood of the final solution. This is practical for low-dimensional variable spaces only and if already sufficient knowledge about the shape and location of the state functions exists.

An often good guess of the location of the final joint β-point is obtained by calculating the joint β-point u^*_{il} of the linearizations which have previously been determined as tangent hyperplanes going through the individual β-points. The individual β-points u^*_j of all the constraints $g_j(u)$ with $j = 1, ..., m$ are generally determined rather reliably. Equality constraints are temporarily treated as inequality constraints. The final joint β-point is denoted as u^*_{12}, as illustrated in figure 2 which depicts an example with two constraints in a two-dimensional variables space.

It is possible to use more information especially if the gradients of the true constraints are evaluated in this new point. Note also that the true equality constraints will usually not be fulfilled. The choice of this starting point makes the algorithms considerably better convergent. Besides it can provide the so-called crude FORM solution at very little additional computational effort.

Linear Approximation of the Admissible Domain

First of all informative error messages must be provided if mathematically and/or physically inadmissible domains are entered. Restart of the algorithm at another admissible (and, possibly, feasible) point then sometimes will lead to convergence. The question is how to choose an appropriate restarting point. An efficient strategy to stabilize the β-point search is based on a successive approximation of the limiting curve of the admissible domain by enveloping tangent hyperplanes.

For this purpose additional linear constraints are activated whenever the optimization algorithm leaves the admissible domain. These hyperplanes are perpendicular to their direction cosines. Thus they make use of all information available at that time. They are also referred to as dummy constraints in the sequel as they are not taken into account if the algorithm remains in the admissible domain. Consider for example an iteration point, as denoted by number 1 in figure 3. In this case a check in the program routine evaluating the state functions would report an error as point 1 is in the inadmissible domain. Instead of interrupting the reliability analysis completely, only the optimization is stopped and restarted with additional information after the following two steps.

Step 1 includes a bisection procedure on the line connecting point 1 with the coordinate origin. The stretch between the origin and the very last bisection point is repeatedly halved until the current bisection point falls into the admissible domain (point 4 in figure 3). Then a dummy constraint, given as a linear hyperplane through the last inadmissible point (point 3 in figure 3) and perpendicular to the bisection line, is activated.

Step 2 is characterized by another bisection procedure, taking the final result and the last inadmissible point of step 1 (points 4 and 3 respectively in figure 3) as a basis. Now the stretch between the base point of the activated dummy constraint (point 3) and the last bisection point is repeatedly halved until the current iterate lies in the inadmissible domain (point 7 in figure 3). Finally the activated dummy constraint can be improved by moving it parallelly into that bisection point.

Taking account of the additional constraint and utilizing the admissible bisection point of step 1 (point 4 in figure 3) as an initial solution, the optimization algorithm for the β-point search is restarted. The described procedure is repeated until the admissible domain is enveloped by a polyhedron formed by the hyperplanes whose individual β-points all lie in the inadmissible domain. The optimization algorithm is stopped either when convergence is accomplished or until an arbitrary maximum number of dummy constraints is reached.

It is clear that this strategy is purely heuristic. There is no proof of convergence which may be difficult to assess unless for special cases. It can only work in the rotationally symmetric standard normal space. The strategy may cut out domains which are still admissible and, in the extreme, are infeasible thus changing and reducing the effective safe domain. For example, this is rather likely if the state functions are only defined in the infeasible (safe) domain. Failure probabilities will thus be overestimated. And there is still a possibility that the algorithm will not converge. Several alternatives for the chosen bisection strategy and different algorithmic details appear possible and are under study. Their discussion is beyond the scope of this paper. Nevertheless, the proposed strategy is an attempt to overcome certain mathematical and/or physical problems which can show up during optimization and which are usually overlooked when formulating the state functions or, in the worst case, cannot be taken care of at all.

EXAMPLE

The convergence problems during the β-point search have been found particularly severe in investigations for inspection and maintenance for ship and offshore structure components subject to fatigue damage. The expected failure probability of a structural component after a service time t_s needs to be computed depending on observations of the crack size before t_s. The state functions can be expressed by

$$F = \{a_{crit} - a(t_s) \leq 0\} \quad (6)$$

$$B = \{\epsilon\, a_{obs} - a(t_{obs}) = 0\} \quad (7)$$

or

$$B = \{\epsilon\, a_{obs} - a(t_{obs}) \leq 0\} \quad (8)$$

where a_{crit} is the critical crack length and $a(t_s)$ is the expected crack length at t_s. a_{obs} denotes the observed crack size, ϵ the measurement error, and $a(t_{obs})$ the expected crack size at t_{obs}. In the following we concentrate on the simple case of only one observation of type equation (7). The above formulation of the constraints is in the so-called event space as actual crack sizes are directly compared with each other. Another possibility to represent the reliability problem is to transform the equations into the time space, thus referring to corresponding crack propagation time intervals. Crack growth is described by the Paris/Erdogan relationship (Paris and Erdogan, 1963)

$$\frac{da}{dt} = C\,\nu\,\left[\Delta s\,\sqrt{\pi a}\,Y(a)\right]^m \quad (9)$$

with the mean value upcrossing rate ν of the stress process, the stress range Δs, a geometry function $Y(a)$ depending on the crack shape, and the material constants C and m. By integrating equation (9) the crack lengths $a(t_s)$ or $a(t_{obs})$ can be obtained. One of the physical restrictions limiting the admissible domain can be determined by investigating the $a(t)$-curve. For $m > 2$ and $Y(a) = Y = const$ it can be shown that at a certain time $a(t)$ gets singular, for greater times even negative. This time t_{exp}, called crack explosion time, is given by

$$t_{exp} = -\frac{a_0^{\frac{2-m}{2}}}{\frac{2-m}{2} C \pi^{\frac{m}{2}} E\left[\Delta s^m\right] \nu Y^m} \qquad (10)$$

with the initial crack size a_0 and the m-th power of the equivalent stress range $E[\Delta s^m]$. For cases with $Y(a) \neq const$ no analytical solution for $a(t)$ can generally be obtained. Hence to guarantee fairly accurate results for t_{exp}, Y^m in the denominator is replaced by $Y(a_0)^m$ and an additional safety factor $F_{exp} \leq 1$ is introduced into the numerator. Evidently, further physical restrictions can be found for this example, but a more detailed description would go beyond the scope of this paper.

In a numerical investigation an example with the stochastic model given in table 1 was considered. The analyzed crack is a one-dimensional edge crack in a plate with a width of 1000 mm, described by its geometry function according to Broek (1986).

The effect of the different solution strategies has been studied for several observation times. Ranges where one strategy still succeeds while another fails can be determined. Whenever FORM results were obtainable the respective SORM results were calculated, too. Figure 4 shows the results of the parameter study in t_{obs}. As the differences between FORM and SORM results are negligibly small no separate curve for SORM is shown.

It can be seen that crude FORM (individual linearization) finds solutions for each observation time in the considered interval. In no case dummy constraints are activated. However, for $t_{obs} > 16.0$ notable discrepancies can be detected between the crude FORM and FORM results revealing the crude method as rather inaccurate in this range. Examples can be constructed in which these differences between crude FORM and a correct first order solution are even larger. The fact that SORM does not produce much better results than FORM for all probability levels also implies for this example that it is more important to locate the exact joint β-point than to take into account curvatures of the state functions.

It can further be seen that FORM without any special strategies and state functions formulated in the event space leads to solutions only for observation times in the small interval $12.0 \leq t_{obs} \leq 16.5$. Beyond these limits the optimization algorithm leaves the admissible domain and stops with a fatal error. Using state functions transformed into the time space the solvable interval increases slightly to $12.0 \leq t_{obs} \leq 17.0$. Continuing the investigation in the time space the following observations can be made. Activation of additional linear constraints to approximate the admissible domain yields two more β-values at $t_{obs} = 11.5$ and $t_{obs} = 17.5$. Additional constraints combined with an improved starting solution by taking account of the joint β-point of the individual linearizations completes the missing values for $t_{obs} > 17.5$. For $t_{obs} < 11.5$ no results can be obtained by FORM. In this case "normal" non-convergence occurs with the applied algorithm and special strategies do not help. Alternative algorithms may, however, be successful.

Table 2 contains the numerical results for five selected observation times, each characteristic for one of the ranges discussed above.

Finally, it is mentioned that the new strategies are somewhat more expensive than crude FORM. It also is worth noting that the example chosen is not an extreme one.

CONCLUSIONS

Based on the presented non-algorithmic sources of non-convergence and their proposed solution strategies in reliability updating, it can be concluded that in the development of a general program code for reliability analysis not only good optimizers have to be chosen but various extreme scenarios have to be considered to achieve maximum stability. A first possibility is to investigate convergence properties in different formulation spaces. A second somewhat laborious but in difficult cases necessary strategy is the selection of a starting value for the algorithm which is found as the joint β-point of the constraints linearized previously in their respective individual β-points. A third strategy consists of adding dummy constraining hyperplanes which sequentially envelop the admissible domain such that their individual β-points are in the inadmissible but otherwise feasible domain.

REFERENCES

Broek, D., *Elementary Engineering Fracture Mechanics*, Martinus Nijhoff, Dordrecht, 1986

Madsen, H. O., Random Fatigue Growth and Inspection, *Proceedings ICOSSAR '85*, Kobe, Japan, 1985

Madsen, H. O., Sorensen, J. D., and Olesen, R., Optimal Inspection Planning for Fatigue Damage of Offshore Structures, *Proc. 5th Intern. Conf. on Structural Safety and Reliability - ICOSSAR '89*, San Francisco, ed. A. Der Kiureghian, IASSAR, 1989

Fujita, M., Schall, G., and Rackwitz, R., Adaptive Reliability Based Inspection Strategies for Structures Subject to Fatigue, *Proc. 5th ICOSSAR '89*, (A.H.-S.

Ang, et al. ed.), San Francisco, Aug. 1989, Vol. II, pp. 1619-1626

Gollwitzer, S., and Rackwitz, R., Comparison of Numerical Schemes for the Multinormal Integral, IFIP-TC7 Conf., Aalborg, May, 1987, in: Thoft-Christensen, P. (Ed.), *Lecture Notes in Engineering*, Springer-Verlag, Berlin, 1987

Gollwitzer, S., and Rackwitz, R., An Efficient Numerical Solution to the Multinormal Integral, *Probabilistic Engineering Mechanics*, 3, 2, 1988, pp. 98-101

Goyet, J., Faber, M. H., Paygnard, J.-C., and Maroini, A., Optimal Inspection and Repair Planning: Case Studies Using IMREL Software, *Proceedings 13th OMAE '94*, Houston, 1994, Vol. II, pp. 325-333

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

Paris, P., and Erdogan, F., A Critical Analysis of Crack Propagation Laws, *Journal of Basic Engineering*, Trans. ASME Vol. 85, 1963, pp. 528-534

Schittkowski, K., *User's Guide for the Nonlinear Programming Code NLPQL*, Version 2.1 - 2.5, University of Stuttgart, Germany, 1985 - 1994

APPENDIX: FIGURES AND TABLES

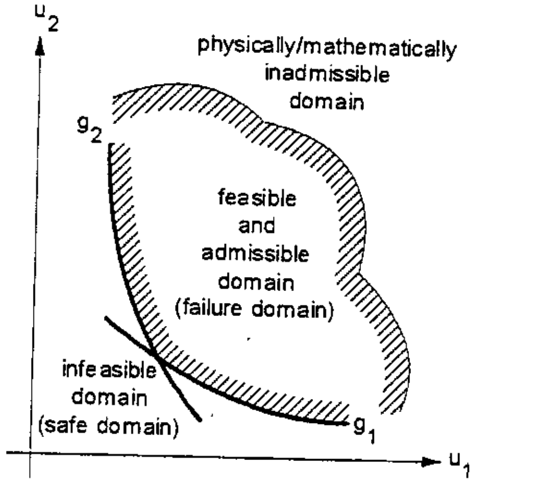

Figure 1: Definition of feasible and admissible domain

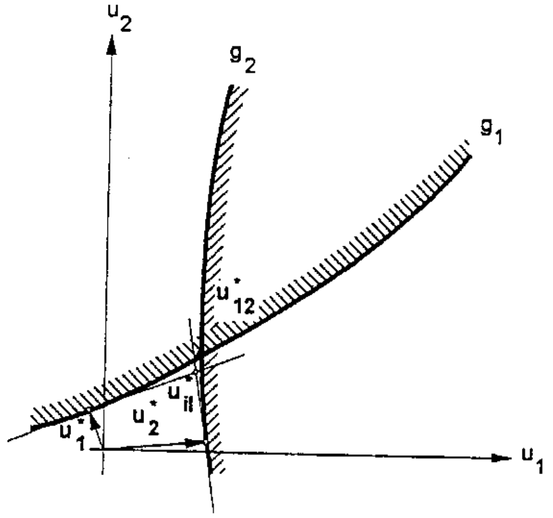

Figure 2: Improved starting solution after individual linearization of the constraints

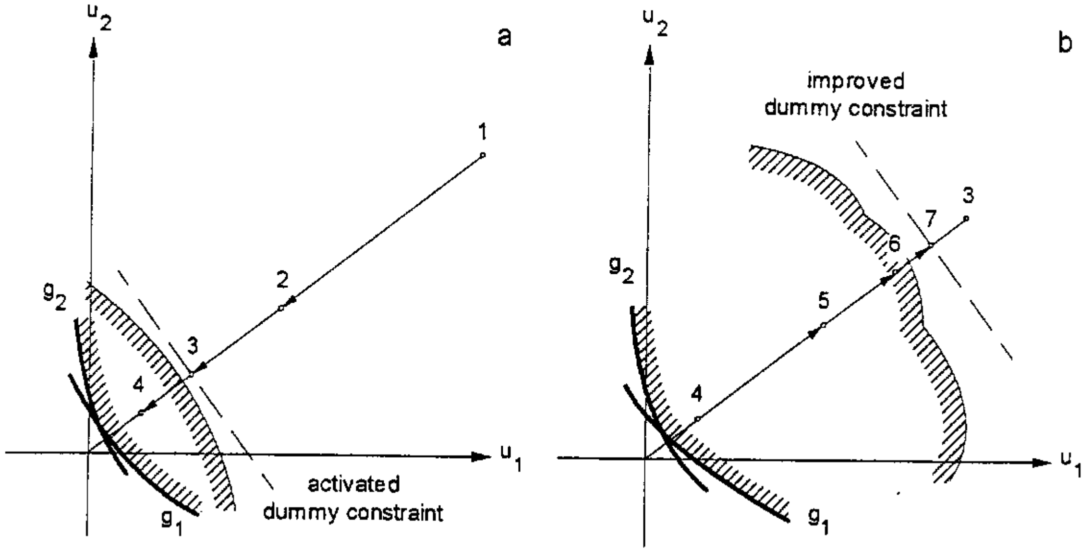

Figure 3: Activation (a) and improvement (b) of an additional constraint to approximate the admissible domain

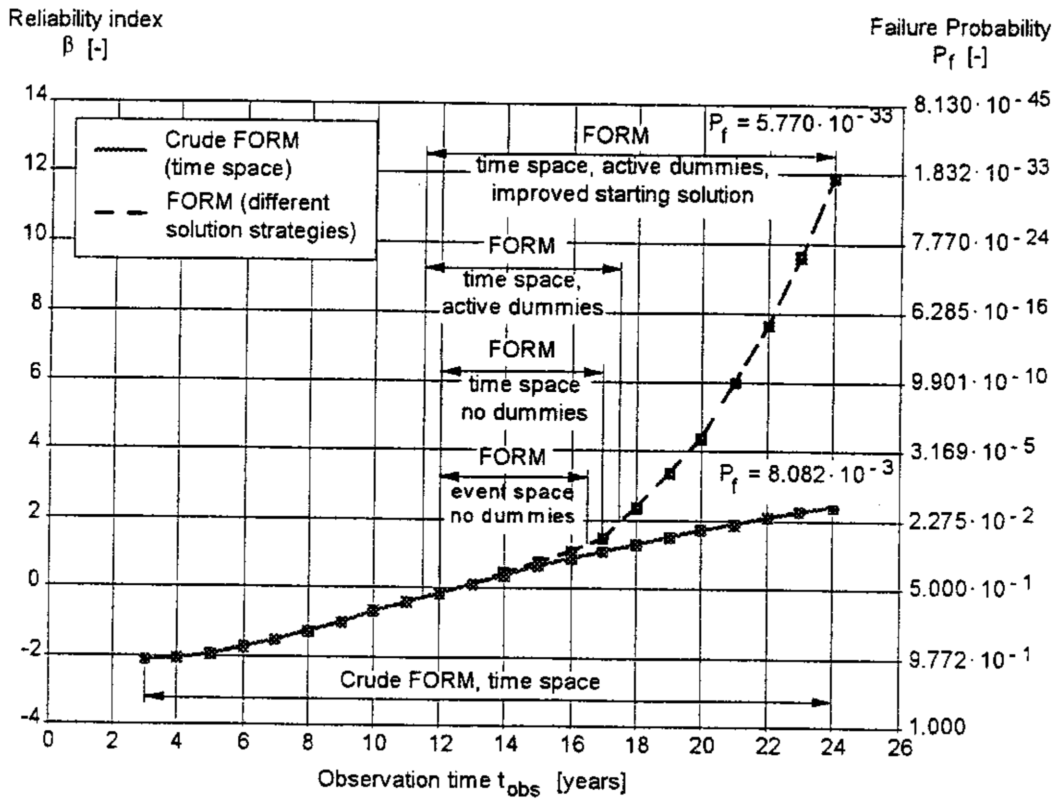

Figure 4: β-curves for different solution strategies

Variable	Distribution function	Mean	Coefficient of variation
a_0	Rayleigh	1 [mm]	0.5
C	Lognormal	$1.13 \cdot 10^{-13}$ [-]	0.4
m	Constant	3 [-]	-
ϵ	Lognormal	1 [-]	0.2
a_{obs}	Constant	3 [mm]	-
$E[\Delta s]$	Constant	30 [MPa]	-
ν	Constant	0.1 [1/s]	-
a_{crit}	Constant	50 [mm]	-
t_s	Constant	25 [years]	-
F_{exp}	Constant	0.9 [-]	-

Table 1: Stochastic model of the example

t_{obs}	Method	Solution strategy			Number of			β	P_f
		const. form.	impr. start	dum. const.	active dum.	iter. num.	func. calls		
6.0	crude FORM	ts	no	no	-	8	49	-1.756	0.9605
12.0	crude FORM	ts	no	no	-	6	36	-0.1194	0.5475
	FORM	es	no	no	-	4	73	-0.1167	0.5465
	FORM	ts	no	no	-	4	73	-0.1167	0.5465
	SORM	ts	no	no	-	4	115	-0.1167	0.5465
17.5	crude FORM	ts	no	no	-	8	48	1.218	0.1116
	FORM	ts	no	yes	1	6	106	1.857	0.0361
	SORM	ts	no	yes	1	6	157	1.833	0.0334
20.0	crude FORM	ts	no	no	-	8	48	1.720	0.0427
	FORM	ts	yes	yes	1	24	221	4.400	5.0E-05
	SORM	ts	yes	yes	1	24	272	4.343	7.0E-05
22.0	crude FORM	ts	no	no	-	8	48	2.079	0.0188
	FORM	ts	yes	yes	2	17	142	7.627	1.0E-14
	SORM	ts	yes	yes	2	17	193	7.520	1.0E-14

Table 2: Results with different solution strategies for five characteristic observation times (const. form = constraint formulation, impr. start = improved starting solution, dum. const. = dummy constraints, active dum. = active dummy constraints, iter num. = iterations for numerator probability, func. calls = state function calls, ts = time space, es = event space)