



# Stochastic modeling of structural fatigue damage in High Strength Steel structures

Jiri Brozovsky, Martin Krejsa, Petr Lehner, Premysl Parenica

VSB - Technical University of Ostrava, Czech Republic

[jiri.brozovsky@vsb.cz](mailto:jiri.brozovsky@vsb.cz), <https://orcid.org/0000-0001-6866-8718>

[martin.krejsa@vsb.cz](mailto:martin.krejsa@vsb.cz), <https://orcid.org/0000-0003-0571-2616>

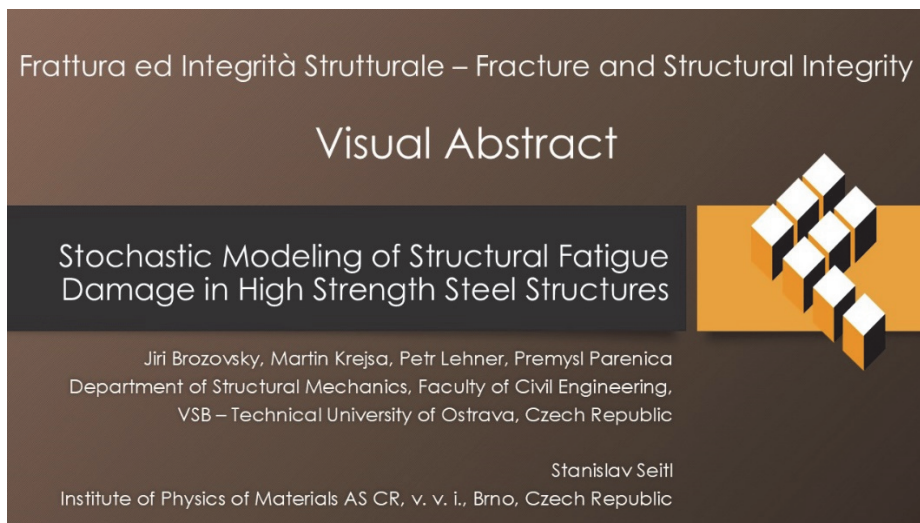
[petr.lehner@vsb.cz](mailto:petr.lehner@vsb.cz), <http://orcid.org/0000-0002-1478-5027>

[premysl.parenica@vsb.cz](mailto:premysl.parenica@vsb.cz), <https://orcid.org/0000-0002-5491-1542>

Stanislav Seitl

Institute of Physics of Materials AS CR, v. v. i., Czech Republic

[seitl@ipm.cz](mailto:seitl@ipm.cz), <https://orcid.org/0000-0002-4953-4324>



**Citation:** Brozovsky, J., Krejsa, M., Lehner, P., Parenica, P., Seitl, S., Stochastic Modeling of Structural Fatigue Damage in High Strength Steel Structures, *Fracture and Structural Integrity*, 71 (2025) 273-284.

**Received:** 25.09.2024

**Accepted:** 24.10.2024

**Published:** 28.11.2024

**Issue:** 01.2025

**Copyright:** © 2024 This is an open access article under the terms of the CC-BY 4.0, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

**KEYWORDS.** Probability of Failure, Fatigue, High Strength Steel, Direct Optimized Probability Computation Method, Parallel Processing.

## INTRODUCTION

One of the most common failure mechanisms of steel structures is associated with material fatigue. A frequent task of engineers is to determine the design or residual service life of a steel structure, which is often associated with evaluating its current condition based on inspections focused on fatigue damage.



Failure primarily occurs due to the initiation of cracks through stress cracking followed by fatigue crack growth requiring a certain stress range and a sufficiently large number of cycles until final failure ensued through sudden and unstable fracture after fatigue growth to a critical crack size appears. The state-of-the-art review of metal fatigue has been published, e.g., in [1]. Numerous methods have been proposed for the evaluation of the remaining fatigue life of load-carrying steel structures and steel bridges [2], some of them are based on probabilistic approaches [3], as a number of factors affecting the fatigue behavior of structures, e.g. operating loads, material properties and environmental conditions, are random in nature. Numerous bridges that use high-strength steel (HSS) for the main structural components have been built recently, mainly owing to its material durability, architectural qualities, and strength requirements. The use of high-strength steel for civil engineering structures is relatively recent. Numerous bridges that use high-strength steel for the main structural components have been built recently, mainly owing to its material durability, architectural qualities, and strength requirements [4]. A significant advantage of high-strength steel is its favorable ratio of strength/mass. High-strength steel has no pronounced yield plateau and exhibits an early deflection from linear elastic behavior with strong strain hardening, which is a significant difference compared to low or medium-carbon steel (S235, S355). The shape of the stress-strain curve of the base metal in the plastic range ensures higher plastic moment resistance than low-medium steels. It is interesting to note that fatigue crack propagation rates versus stress intensity range exhibit with increasing yield strength of steels, similar to report by [5]. This implies that there is important place for experimental study with crack initiation and propagation in HSS.

### COMPUTATIONAL MODELS FOR FATIGUE FAILURE PREDICTION

The reliability of the load-bearing structure, stressed by variable loads, is significantly affected by the degradation effects, caused mainly by the fatigue of the base material. The design process of these structures is based on the concept of the so-called Wöhler curves (or  $S-N$  curves), in which a limited-service life until failure is allowed, which is very problematically determined based on a constant oscillation and an assumed number of load cycles. The methodology was gradually developed into procedures describing real conditions and facilitating the work of designers, e.g. [6].

Randomly appearing fatigue cracks on existing structures - crane tracks and bridges, indicate a certain imperfection of this design methodology. Methods are being developed considering possible defects and defects in the form of initialization cracks, which significantly accelerate the propagation of fatigue cracks. One of the alternatives is linear fracture mechanics, which has been the subject of research for many years, especially in the field of engineering, and is gradually being taken over and adapted to the issue of the design of load-bearing building structures. It is mainly used to determine inspection times and to analyze their results, which, if cracks are not detected, lead to a conditional probability of their occurrence.

The problem solved in this paper is focused on fatigue damage of a selected bearing element of the building steel structures or bridges, in which the characterization of the acceptable size of the fatigue crack from the edge is evaluated. This dimension has a decisive role in the degradation of an element designed for an extreme combination of loads but burdened by operational variable effects. This is a possible traceable degradation of the designed element to the limit state of bearing capacity.

Three dimensions are important for the prediction of the propagation of fatigue cracks. The first is the initiation size, the second is the measurable size, and the third significant dimension is the acceptable fatigue crack length size, determined with the reliability criterion for the limit state of the load-bearing capacity of the element under investigation. Fatigue crack damage is dependent on the number of cycles of stress oscillation, which represents a time factor in the course of reliability over the entire design life. The probability of failure increases with time and reliability decreases.

A fatigue crack, which weakens a structural element by a certain area, is described by only one overall dimension  $a$  when growth is monitored. To describe crack growth, the method of linear elastic fracture mechanics defined by the Paris-Erdogan equation (also known as the Paris–Erdogan law, [7]) is most often used:

$$\frac{da}{dN} = C \cdot (\Delta K)^m \quad (1)$$

where  $C$ ,  $m$  are material constants, obtained experimentally,  $a$  is the crack dimension,  $N$  is the number of loading cycles and  $\Delta K$  is stress intensity factor range.



## PROBABILISTIC RELIABILITY APPROACHES

In general, probabilistic analysis of structures or parts thereof can be performed using many different methods. These can be sorted into e.g. *simulation-based techniques* and *approximation techniques*, which further branch into subgroups involving advanced approaches and methods.

The basis of *simulation-based techniques* is the Monte Carlo (MC) method [8], which depends on the generation of a large number of randomly variable input parameters and the subsequent numerical solution. An example of the use of MC in civil engineering is the *Simulation-based Reliability Assessment* method (SBRA) [9], which uses histograms as input data, MC as a tool for generating random variables, and the reliability of the structure is determined by comparing the probability of failure and the normative design probability of failure. The disadvantage of using MC in this way is the need for a large number of random inputs for sufficient accuracy, and hence the high computational complexity. Therefore, more advanced methods have been developed, falling into the categories of *stratified sampling techniques* and *advanced simulation techniques*.

An example of *stratified sampling techniques* is *Latin Hypercube Sampling* (LHS) [10], which is an evolution of the MC method and significantly speeds up the computational process itself by using distribution functions of equal probability.

For the sake of completeness, we must also mention a representative of *advanced simulation technique*, where the computational procedure focuses on specific areas of structural failure and their primary goal is to efficiently obtain the result of the analysis. We include here, for example, *Importance Sampling* [11], *Adaptive Sampling* [12], *Directional Simulation* [13], or *Slice Sampling* [14]. The second group, *approximation techniques*, use, for example, analytical approximations to determine the outcome of a chosen phenomenon. The most important methods are *First Order Reliability Method* (FORM) [15] and *Second Order Reliability Method* (SORM) [16]. Both FORM and SORM use just analytic approximations to estimate the reliability index as the distance from the origin to the limit state surface. FORM uses a linear approximation, while SORM uses an improved quadratic or higher-order approximation to approximate the result more accurately. Alternatively, we can then mention *Response Surface Method*, *Perturbation techniques* and *Artificial neural networks*, which differ in principle only in the approximation method used.

As an alternative to all of these methods, the *Direct Optimized Probability Computation method* (DOProC) [17], which belongs to a separate category of numerical approaches, was selected for the presented analysis. In this regard, the Point Estimate Method (PE) [18] should be mentioned here for DOProC might be compared to it or at least put into the category of PE methods. Since MC is the most used, the results of the DOProC calculation are partially compared with those using MC. There are several advantages offered by an alternative approach:

- ✓ The DOProC method does not depend on any type of randomly generated numbers. Thus, the results are repeatable and there is no need for a random numbers generator. There is also no requirement to consider the quality of such generators or their usability and effectiveness in parallel environments.
- ✓ Use of an alternative approach can provide another set of results which can be compared with more common approaches. Such comparison can help to detect possible issues and problems.
- ✓ The DOProC in its basic form can be parallelized in a way similar to the basic MC procedure.

## DIRECT OPTIMIZED PROBABILITY COMPUTATION

The main overview of the method is available in [17]. The Direct Optimized Probability Computation Method (DOProC) uses input variables provided in form of tables - they are usually graphically represented as so-called bounded histograms. The main principle of the DOProC method is that all possible combinations of random variable values are tested. Because of the representation of variables, there is always a finite number of combinations that have to be investigated.

This is the main advantage of the method as it allows to have a repeatable solution that is only affected by the quality of the tabular representation of data. But it is also the main disadvantage as for every single combination the full solution (usually called the realization in Monte Carlo content) of the studied problem must be done. It is obvious that the numbers of such realizations are quite high (typically  $10^9 \dots 10^{12}$ ) and much higher than for a typical Monte Carlo - based solution ( $10^6$  for structural reliability problems).

Another issue is the fact that the DOProC cannot work with variables represented by functions. Every such function has to be approximated by a table. This can be sub-optimal for many uses but it is often not a problem for engineering applications where much input data came from *in situ* or laboratory measurements. Measured data like numbers of cycles, load values, and material parameters are available in discrete forms. From the practical point of view, it is then easier to directly use such measured data for numerical analysis (of course after their verification and possible cleaning).



The high number of computations required is of course an issue for many possible applications. There have been developed techniques to greatly reduce the number of combinations that have to be investigated. They are called *optimization techniques* in [17]. It is important to note that many of these optimization techniques depend on the properties of studied input variables thus their effectiveness is not guaranteed for a general problem. However, they were used with great effect to solve well-defined practical problems. In these cases, a very slow run time environment was used - the BASIC interpreter of an office package spreadsheet application - but feasible computational speeds were reached thanks to the correct use of above-mentioned optimization techniques.

## PARALLEL APPROACH TO COMPUTATIONS

### *Parallelization of Monte Carlo method*

MC-based approaches are often used for practical applications like the determination of fatigue life of various structures or components in many areas [19]. These approaches may one finite element analysis (or other time-consuming computation) per simulation. Reduction in computational time is often accomplished by parallelization of the procedure. Parallel execution of Monte Carlo computation is relatively straightforward because in the most basic form of the MC every simulation can be considered independent of others and thus executed in parallel. Usually, there is only a need to communicate between processes when the initial data are distributed and then when the results are collected. This allows effective use of parallel computing using, for example, single program multiple data paradigm, often in form of the Message Passing Interface. However, there are several challenges. For example, the more advanced forms of the MC (like the LHS) usually add some additional complexity which usually adds the necessity of additional inter-process communications. In any case, there is an unavoidable need for a reliable random numbers generator which can also run in parallel [20]. Several efforts exist. One of commonly used is the SPRNG (the Scalable Parallel Random Numbers Generator) [21]. There are several efforts to improve this generator by adding hardware support for random numbers generation based, for example, on the GPU hardware [22] or on the custom hardware implementation on the field-programmable gate array (FPGA) boards [23]. These approaches, however, require such specialized hardware to be available on the systems where computations are running.

### *Serial and parallel Monte Carlo implementations*

In the presented work the Monte Carlo (MC) - based approach is used mainly for comparison with the DOProC solution. The basic program algorithm is the serial one. It uses input data in form of bounded histograms which is usually the form of data which are available from measurements and laboratory tests are often done in civil engineering area. Results of computations are available in the same form.

There are several approaches to preparing bounded histograms of results - their size can be pre-defined before start of computation, or these parameters can be determined at run time based on the first  $m$  simulations. Then these histograms can be used to store results without further changes of their boundaries and numbers of intervals. Alternatively, histograms can be expanded if needed but interval size cannot be further changed. Use of bounded histogram representation of both input and output data makes the MC - based code more compatible with the DOProC code in terms of inputs and outputs as the DOProC principle is based on use of bounded histogram.

Random numbers are generated with use of an external library. The SPRNG library [21] is used. The code can use a less sophisticated build-in generator, too but it was not utilized for the discussed works.

The code execution is relatively fast (a single time step of the problem studied below can be analyzed in tens of seconds if  $10^6$  simulations are executed) but for more complex problems a parallel execution of the MC is beneficial.

For the studied area of problems statistical dependencies between input variables are not used as these dependencies cannot be determined so far on basis of in situ tests and laboratory measurements. From the point of view of the computational procedure it simplifies parallelization because communication between parallel processes can be limited. Thus, the code uses the *single program* - multiple data approach which is implemented with the use of the Message Passing Interface (MPI) in the computer code.

The initial data (input variables represented by bounded histograms) are distributed to all  $k$  processes (with use of `MPI_Bcast()`) and then every process run  $1/k$  of total number of simulations. At the end the data are collected from all processes with use of `MPI_Allreduce()`.

If the size of histograms has to be determined during run-time as it was described above then there is another round of inter-process communication (`MPI_Reduce()`, ... `MPI_Bcast()`) to collect the data computed so far and then to

propagate computed parameters of histograms for results. All further changes (if any) are stored locally, and they are synchronized after the end of computations.

Use of the SPRNG [21] random number generator introduces some more inter\*-process communication but its influence on computational time is very low.

### Parallelization of DOProC procedure

The DOProC approach in the proposed form can be parallelized in a relatively straightforward way. The input data (in histogram form) can be distributed to all processes and then every process only computes a part of the result histogram(s) data. These results are then collected and combined. The computer code has been made with use of the MATLAB environment where the SPMD functionality was used. The SMPD functionality is principally similar to the MPI approach. That is, computations are executed on  $N$  independent processes with relatively expensive communication. Thus, the same philosophy as for the Monte Carlo - based algorithm was adopted here: the only necessary communication is input data distribution at the beginning and collection of results after the solution.

To simplify the implementation a decision was made to use the same interval for all input and output variables. Such approach allows eliminating of any communication during the solution because parameters of result data histograms can be determined in the beginning.

As it was mentioned before DOProC approach uses bounded histograms for input data. The most basic approach investigates all possible combinations (see Fig. 1) of input histogram intervals to compute results. Every possible combination of input data constitutes an equivalent to one *simulation* in the MC-based approach. The obvious issue is of course number of these combinations which depends on number of intervals in input data histograms. Thus unlike the MC number of such combinations is independent on target reliability level or on other parameters.

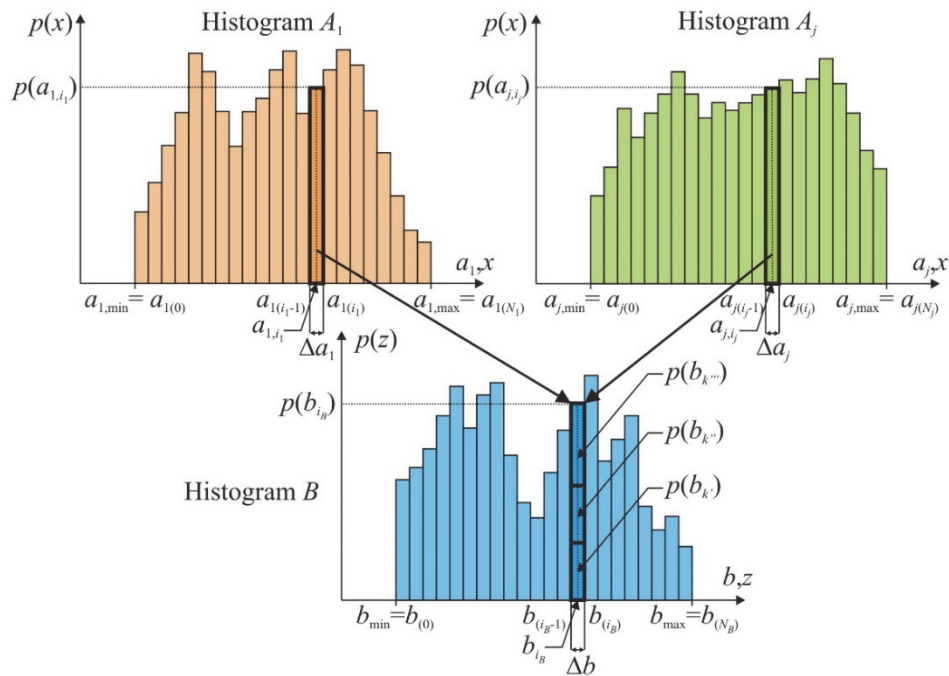


Figure 1: Basic scheme of DOProC operation: The principle of performing numerical operations with histograms of two independent random variables, modified from [17].

Such a solution was implemented as the ObjectPASCAL code and successfully used in several applications, including practical engineering tasks of mining anchor's reliability or the assessment of mining reinforcements. However, in this case it was possible and feasible to introduce several DOProC *optimizations* to considerably reduce number of necessary operations. In a general case, it can be very complicated such procedural *optimizations* thus it other ways of speeding up computations have to be used. The problem of fatigue damage prediction [24] has been so far studied as a general case without any considerations about random variables parameters that can be used to reduce the number of computations.



The DOProC procedure is similar to the MC one in terms of the requirement for data operations. The input data have to be represented by bounded histograms. The results of computations are obviously obtained in the same form as it is illustrated in Fig. 1. The number of intervals on results data histograms has to be determined before the computation starts. It is possible to use a similar procedure as the Monte algorithm used to determine histogram parameters at run-time but there was no obvious reason for it.

The solution can be divided between  $k$  processes in the analogy of the MC-based approach (see Fig. 2). The single program - multiple data was thus used in the form of the MATLAB SPMD functionality. The actual implementation utilized by the MATLAB is based on the MPI approach and libraries, so it is very similar in terms of strength and weak points to the MC-based procedure described above.

Inter-process communication is minimal in the implemented problem: the same input data are made available to all processes at the beginning. Then every process executes its part of the computations. At the end of computation, the data are collected. Thus the procedure and its implementation are relatively straightforward and code modifications against the serial version are minimal: the serial version code has 266 lines of code in total (heavy use of build-in MATLAB functions are obvious), the parallel one is 52 lines longer.

## COMPARATIVE CALCULATION

The DOProC method has already been successfully used in the past to estimate the fatigue resistance of some structures. In [24] was performed the probabilistic assessment of a steel/reinforced concrete bridge from the highway in Slovakia using code FCProbCalc, which is possible to design the system of the inspections focused on a detail, where a longitudinal steel beam connects to a steel transversal beam, which tends to suffer from fatigue damage. The demonstration of using the methodology for the prediction of fatigue damage over time was made in [24] for the single edge-cracked steel element with rectangular cross-section under various loading based on experimentally obtained data and subsequently derived calibration functions that describe fatigue crack propagation for different types of stress. It is necessary to mention the fact that in all solved cases, the DOProC method has proven itself due to the accuracy of the solution, easy implementation, and, in the case of the use of the mentioned optimization techniques, also to the satisfactory calculation time.

### *Description of the computational model*

A three-point bending laboratory sample with pre-defined crack  $a$  was studied as a benchmark test example. The scheme of the analyzed sample is shown in Fig. 3. This particular problem was thoroughly experimentally investigated by the Institute of Physics of Materials of the Czech Academy of Sciences [25] thus necessary input data for numerical analysis have been available.

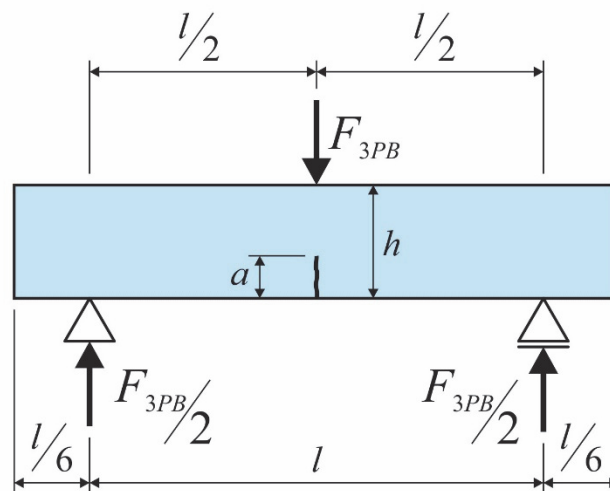


Figure 3: Testing sample scheme.



The probability of failure  $p_F$  is defined as the probability of the occurrence of a situation where the load effects  $E$  exceeds the structural resistance  $R$ , i.e. a situation where the safety margin  $G_{fail}$  is negative:

$$p_F = P(R_{a_{ac}} < E(N)) = P(G_{fail}(\mathbf{X}) < 0) \tag{2}$$

where  $\mathbf{X}$  is the vector of random properties (material properties, geometric properties of the structure, load effect parameters, and - in the studied area of problems - parameters of fatigue crack).

The analytical formulation is based on the Paris-Erdogan law, the load effects function  $E$  can then be described by the equation:

$$E(N) = C \cdot \Delta\sigma^m \cdot (N_a - N_0) \tag{3}$$

where:  $C$ ,  $m$  are material constants (fatigue parameters) from laboratory tests,  $\Delta\sigma$  is stress range,  $N_0$  is the number of cycles before the start of analysis (0 in the studied case), and  $N_a$  is the number of cycles for which the computation is executed.

The resistance function was a form of:

$$R_{a_{ac}} = \int_{a_0}^{a_{ac}} \frac{1}{(\sqrt{\pi} \cdot a \cdot f(a))^m} da \tag{4}$$

The  $a_0$  parameter is the initial fatigue crack size and the  $a$  is the actual crack size. These parameters have to be compared with the maximum acceptable fatigue crack size from the edge of the specimen  $a_{ac}$ :

$$a_{ac} = b - \sqrt{\frac{3 \cdot F_{3PB} \cdot l}{2 \cdot w \cdot f_y}} \tag{5}$$

where  $F_{3PB}$  is force load of the sample,  $f_y$  is yield stress of the material,  $w$  is width of beam,  $b$  beam height and  $l$  span of the specimen.

To define the resistance  $R$  the calibration function  $f(a)$  is necessary to derive. In the studied case (a three-point bending laboratory sample with a ratio of beam height  $b$  to beam span  $l$  of  $b/l = 4$ ) this function was obtained from [25] in the form:

$$f(a) = 1.0618 - 1.0658 \cdot \left(\frac{a}{b}\right) + 2.9787 \cdot \left(\frac{a}{b}\right)^2 + 1.0435 \cdot \left(\frac{a}{b}\right)^3 \tag{6}$$

where  $a$  is length of fatigue crack.

### *Input data*

Deterministic and random input quantities are given in Tab. 1 and Tab. 2.

In the model were defined four random variables: the number of cycles  $N_a$ , the yield stress of material  $f_y$ , the initial fatigue crack length  $a_0$ , and the loading force  $F = F_{3PB}$ . Random functions of variables were represented by histograms. All of these histograms have 64 bins.

Obtaining the histogram  $N_a$  was slightly more complex as only a histogram  $N$  which represented load cycles in one year was available. The probabilistic calculation is carried out in time steps where one step typically equals one year of the service



life of the construction. If there is a need to compute results at a different time than one year ( $y = 1$  year) it is necessary to compute a histogram for such time. In accordance with [24] the  $N_a$  for given number of years  $y$  can be computed as:

$$N_a = \sum_{i=1}^y N_i, \tag{7}$$

where  $N_i$  is the histogram for cycles in one year and the  $y$  is number of years.

This computation can be done with supporting tools of the ProbCalc software package or computed directly with DOProC approach (it was done in the case of MATLAB-based implementation of the DOProC).

To avoid any DOProC-related influence on the Monte Carlo-based solution the  $N_a$  was also obtained by the computation in the Monte software. For this case, another user-level routine in the C language was developed and used (the solution was executed for 100,000 simulations).

Quantity	Value
Material constant $m$	3
Material constant $C$ [MPa <sup><math>m</math></sup> m <sup><math>(m/2)+1</math></sup> ]	$2.2 \cdot 10^{-13}$
Height of the rectangular cross-section $b$ [m]	0.1
Width of the rectangular cross-section $w$ [m]	0.01
Span of the element $l$ [m]	0.4
Target probability of failure $p_d$ [-]	$0.02277 (\beta_d = 2)$

Table 1: Overview of input deterministic quantities.

Quantity	Distribution	Mean value $\mu$	Standard deviation $\sigma$
Total number of stress peaks per year $N$ [-]	Normal	$10^5$	$10^6$
Yield stress $f_y$ of S690 steel [MPa]*	Lognormal	817.1	6.15
Force in three-point bending test $F_{3PB}$ [kN]	Normal	6	0.6
Initial size of the fatigue crack $a_0$ [mm]	Lognormal	0.2	0.05

\*Note: Calculated from measured data

Table 2: Overview of input random quantities expressed by bounded histograms.

### Analysis of achieved results

The simulation in the Monte software was executed with the use of the above-mentioned custom procedure. The computation was done with the use of 100 000 simulations. The time of complete computation was under 10 seconds on the 3.8 GHz POWER9 computer. Due to the nature of a simulation-based method, the computation was repeated 5 times and average results are presented in Tab. 3.

Years	Monte Carlo	DOProC
35	0.018	0.013
65	0.043	0.047

Table 3: Resulting probabilities of failure  $P_F$  in specified time of structural operation.

Fig. 4 demonstrates the accuracy of the calculation by the DOProC method when comparing serial and parallel calculation. It can be mentioned that differences between the results of serial and parallel computations were minimal (the first five decimal points of numbers were always identical for both serial and parallel case). The graph shows the increasing value of

the failure probability for the operating time of the analyzed structure. The chart also shows that the calculated probability of failure  $p_F$  reached the specified target value between the years 53 and 54 when an inspection focusing on fatigue damage should take place on the analyzed detail in the real structure.

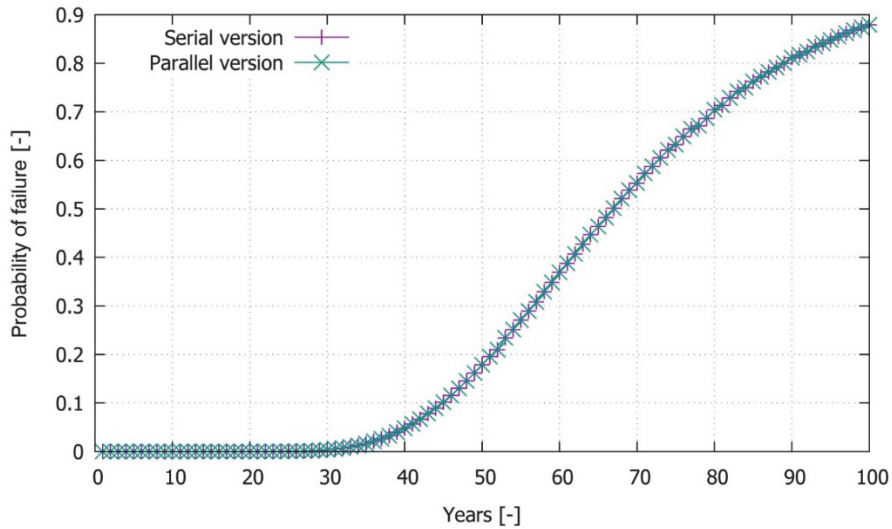


Figure 4: Comparison of results for serial and parallel DOProC computations.

Fig. 5 shows the influence of the number of computing units of parallel computing on the resulting machine time of the analysis.

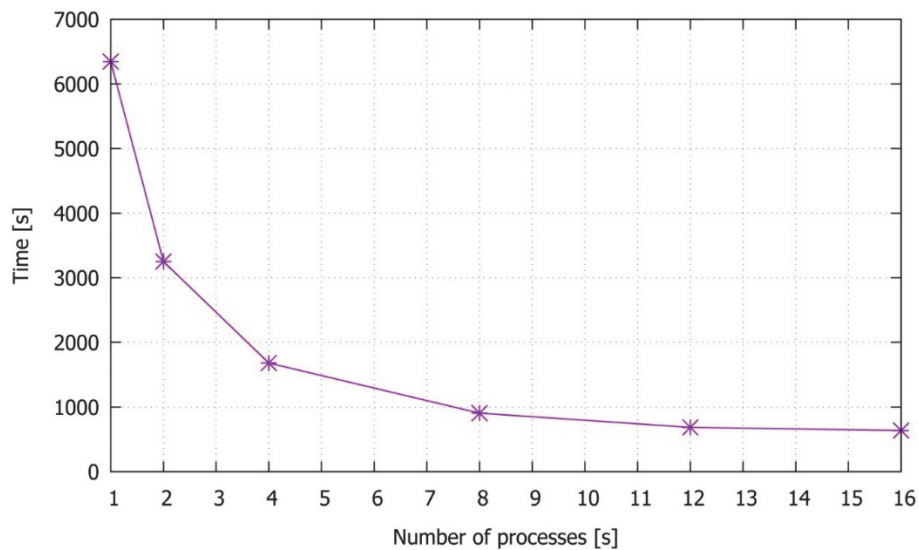


Figure 5: Parallel DOProC computational speed.

The DOProC code was running in an interpreted mode on a different computer thus the computational times are not directly comparable with the Monte Carlo - based analysis. These results were obtained from computations on the 3.7 GHz AMD CPU-based Windows 11 machine with 12 CPU cores. However, the results showed in Fig. 5 demonstrate the relative speedup of the DOProC solution related to the use of the certain number of CPU cores. For the given problem there is considerable speedup for at least 8 cores: for the single CPU core the time was 6,345 seconds and for 8 CPU cores the time was 904 seconds, for 12 cores the time was 685 seconds. It can be thus concluded that the 685 seconds (11.4 minutes) time should be an acceptable time even for civil engineering practice.



## CONCLUSIONS

The article proposed the new parallel implementation of the Direct Optimized Probabilistic Computation method (DOProC). The main objective of this work was speedup up the DOProC method computation times because one of the main disadvantages of this method is an extremely large number of repeated computations required. It leads to long computation times which may be made this method unattractive for potential practical applications. The change from serial execution to parallel one is the thus obvious approach to lessen the above-mentioned disadvantage of the DOProC.

The algorithm was demonstrated on a procedure which was applied to experimentally obtained data from a three-point test of a steel specimen with a predefined crack. The calculation was compared with the classical Monte Carlo method. The basic parameter for comparison was the value of the probability of calculation and then the speed of calculation in case of using parallel calculations.

The obtained results show that the parallel execution of computations leads to a noticeable reduction in computational times while the precision of the solution is not considerably degraded.

The created software for parallel calculation is going to be applied for fatigue damage prediction in the next period for the analysis of more complex details of real HSS structures. In this probabilistic calculation, the author's team will try to consider other factors affecting the fatigue resistance of the supporting element, for example, the effect of corrosion. It is assumed that this code will also be used on a supercomputer platform.

## ACKNOWLEDGEMENT

This contribution has been developed as a part of the research project of the Czech Science Foundation 21-14886S “Influence of material properties of high-strength steels on durability of engineering structures and bridges”.

## REFERENCES

- [1] Zerbst, U., Vormwald, M., Pippan, R., Ganser, H.P., Sarrazin-Baudoux, C. and Madia, M. (2016). About the fatigue crack propagation threshold of metals as a design criterion – A review, *Eng. Fract. Mech.*, 153, pp. 190–243. DOI: 10.1016/j.engfracmech.2015.12.002.
- [2] Cremona, C., Eichler, B., Johansson, B. and Larsson, T. (2013). Improved assessment methods for static and fatigue resistance of old metallic railway bridges, *J. Bridge Eng.*, 18(1), pp. 1164–1173. DOI: 10.1061/(ASCE)BE.1943-5592.0000466.
- [3] Guo, T., Frangopol, D. and Chen, Y. (2012). Fatigue reliability assessment of steel bridge details integrating weigh-in-motion data and probabilistic finite element analysis, *Comput. Struct.*, 112–113, pp. 245–257. DOI: 10.1016/j.compstruc.2012.09.002.
- [4] Miki, C., Homma, K. and Tominaga, T. (2002). High strength and high-performance steels and their use in bridge structures, *J. Const. Steel Res.*, 58(1), pp. 3–20. DOI: 10.1016/S0143-974X(01)00028-1.
- [5] de Jesus, A.M.P., Matos, R., Fontoura, B.F.C., Rebelo, C., da Silva, L.S. and Veljkovic, M. (2012). A comparison of the fatigue behavior between S355 and S690 steel grades, *J. Const. Steel Res.*, 79, pp. 140–150. DOI: 10.1016/j.jcsr.2012.07.021.
- [6] Murakami, Y., Takagi, T., Wada, K. and Matsunaga, H. (2021). Essential structure of S-N curve: Prediction of fatigue life and fatigue limit of defective materials and nature of scatter, *Int. J. Fatigue*, 146, 106138. DOI: 10.1016/j.ijfatigue.2020.106138.
- [7] Paris, P. and Erdogan, F. (1963). A critical analysis of crack propagation laws, *J. Basic Eng.*, 85(4), pp. 528–533. DOI: 10.1115/1.3656900.
- [8] Metropolis, N. and Ulam, S. (1949). The Monte Carlo method, *J. Am. Stat. Assoc.*, 44(247), pp. 335-341. DOI: 10.2307/2280232.
- [9] Marek, P., Gustar, M. and Permaul, K. (1999). Transition from partial factors method to Simulation-based Reliability Assessment in structural design. *Probabilistic Eng. Mech.*, 14(1-2), pp. 105-118. DOI: 10.1016/S0266-8920(98)00021-6.
- [10] McKay, M.D., Beckman, R.J. and Conover, W.J. (2000). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2), pp. 239-245. DOI: 10.2307/1268522.



- [11] Melchers, R.E. (1989). Importance sampling in structural systems. *Struct. Saf.*, 6(1), pp. 3-10. DOI: 10.1016/0167-4730(89)90003-9.
- [12] Bucher, C.G. (1988). Adaptive sampling - an iterative fast Monte Carlo procedure. *Struct. Saf.*, 5(2), pp. 119-126. DOI: 10.1016/0167-4730(88)90020-3.
- [13] Saliby, E. (1990). Descriptive sampling: A better approach to Monte Carlo simulation. *J. Opl. Res. Soc.*, 41(12), pp. 1133-1142. DOI: 10.2307/2583110.
- [14] Neal, R.M. (2003). Slice sampling. *Ann. Statist.*, 31(3), pp. 705-767. DOI: 10.1214/aos/1056562461.
- [15] Hohenbichler, M. and Rackwitz, R. (1982). First-order concepts in system reliability. *Struct. Saf.*, 1(3), pp. 177-188. DOI: 10.1016/0167-4730(82)90024-8.
- [16] Fiessler, B., Neumann, H.J. and Rackwitz, R. (1979). Quadratic limit states in structural reliability. *J. Eng. Mech. Div.*, 105(4), pp. 661-676. DOI: 10.1061/JMCEA3.0002512.
- [17] Janas, P., Krejsa, M., Sejnoha, J. and Krejsa, V. (2017). DOProC based reliability analysis of structures. *Struct. Eng. Mech.*, 64(4), pp. 413-426. DOI: 10.12989/sem.2017.64.4.413.
- [18] He, J. and Sallfors, G. (1994). An optimal point estimate method for uncertainty studies. *Appl. Math. Model.*, 18(9), pp. 494-499. DOI: 10.1016/0307-904X(94)90327-1.
- [19] Zhang, H., Li, L., Ma, W., Luo, Y., Li, Z. and Kuai, H. (2022). Effects of welding residual stresses on fatigue reliability assessment of a PC beam bridge with corrugated steel webs under dynamic vehicle loading. *Structures*, 45, pp. 1561-1572. DOI: 10.1016/j.istruc.2022.09.094.
- [20] Srinivasan, A., Mascagni, M. and Ceperley, D. (2003). Testing parallel random number generators. *Parallel Comput.*, 29(1), pp. 69-94. DOI: 10.1016/S0167-8191(02)00163-1.
- [21] Mascagni, M. and Srinivasan, A. (2000). Algorithm 806: SPRNG: A scalable library for pseudorandom number generation. *ACM Trans. Math. Softw.*, 26(3), pp. 436-461, DOI: 10.1145/358407.358427.
- [22] Gao, S. and Peterson, G.D. (2013). GASPRNG: GPU accelerated scalable parallel random number generator library. *Comput. Phys. Commun.*, 184(4), pp. 1241-1249. DOI: 10.1016/j.cpc.2012.12.001.
- [23] Lin, Y., Wang, F. and Liu, B. (2018). Random number generators for large-scale parallel Monte Carlo simulations on FPGA. *J. Comput. Phys.*, 360, pp. 93-103. DOI: 10.1016/j.jcp.2018.01.029.
- [24] Krejsa, M., Koubova, L., Flodr, J., Protivinsky, J. and Nguyen, Q.T. (2017). Probabilistic prediction of fatigue damage based on linear fracture mechanics. *Frat. Integrita Strutt.*, 11(39), pp. 143-159. DOI: 10.3221/IGF-ESIS.39.15.
- [25] Krejsa, M., Brozovsky, J., Lehner, P., Seidl, S. and Kala, Z. (2018). Stochastic analysis for short edge cracks under selected loads. *AIP Conf. Proc.*, 1978, 150006. DOI: 10.1063/1.5043797.