STRUCTURAL RELIABILITY ANALYSIS USING OBJECT ORIENTED ENVIRONMENT STAND

J. Knabel, K. Kolanek, V. Nguyen Hoang, R. Stocki, and P. Tauzowski
Institute of Fundamental Technological Research, W arsaw, Poland

1. Introduction

Dealing with uncertainty is an essential part of the structural design process. Traditionally uncertainties are taken into account by means of safety factors specified by appropriate design codes. For most of the typical problems this approach allows to obtain safe structures, however the actual safety level of such designs is hard to estimate. On the other hand, a progressive increase of computational capabilities and development of reliability analysis methods and specialized software allows for more rational treatment of inherent uncertainties of material parameters, structural geometry and the applied loads. The contemporary "computer aided" reliability analysis provides a valuable tool for designers and decision makers by enabling them for more realistic estimation of the probability of structural failure, identification of the weakest elements and evaluation of their influence on the safety of entire structure.

The structural reliability analysis can be performed by a number of software packages, such as ANSYS PDS and DesignXplorer, CalREL/FERUM/OpenSees, COSSAN, NESSUS, PERMAS-RA/STRUREL, PHIMECA-SOFT, PROBAN, PROFES, UNIPASS, all of them reported in the special issue of Structural Safety journal [1]. The structural reliability analysis usually requires many evaluations of the so-called failure function, which is defined by means of selected structural responses. Since these responses are most often computed using the finite element (FE) method therefore, a reliability analysis system must interact with the FE analysis program. This is usually realized in two alternative ways: either the reliability and FE codes are closely integrated (linked) or reliability analysis code have interface enabling it to modify and read input and output data files, respectively, of a third-party FE analysis system or other structural analysis program.

Although the first approach has unquestionable advantages such as easy access to FE model parameters or a common graphical user interface etc., it restricts application of reliability analysis to problems supported by the integrated FE analysis code. The second solution gives possibility of performing the reliability analysis for any problem, provided that the corresponding FE computations can be executed from the command line. This approach is often preferred and used in most of the systems listed above.

The key to success in developing any large software package is a proper organization of its code facilitating future modifications and allowing it to be simultaneously developed by many programmers. The code architecture should also facilitate integration of new algorithms into the existing environment. Such a flexibility is offered by object oriented programming and therefore it was C++ that was chosen as a programming language for creating reliability analysis software STAND (Stochastic Analysis and Design), developed in the Institute of Fundamental Technological Research (IFTR) of Polish Academy of Sciences. By analyzing reports in [1] it can be concluded that only few from the above listed programs benefit form using object oriented paradigm.

The graphical user interface implemented in STAND is easy to use and provides the interface to the external computational programs by the concept of data files parsing. Parameters of, say, FE model can be easy identified and linked to random variables in the stochastic model by simply highlighting appropriate fields in a template input file. Similar approach was employed in COSSAN [2] and PROFES [3], for instance. Analogous method is employed for collecting the FE analysis results. As it was mentioned, this type of interface is suitable only for the FE programs that can be run from the command line and read text input data files as well as produce results in the text format. This, however, is the case for most of the commercial and research oriented FE codes.

2. STAND environment description

So far, these are mainly time invariant component reliability problems that can be addressed with STAND. In addition to standard reliability analysis methods like FORM, SORM, crude Monte Carlo, unimodal importance sampling or mean value first order method STAND offers multimodal adaptive importance sampling method (MAISM), developed recently in IFTR. MAISM proved to be very efficient in dealing with noisy nonlinear limit state functions (LSFs), see [4].

The basic statistical analysis is available in STAND as well. It is based on two sampling methods: crude Monte Carlo and a very efficient descriptive sampling design - optimal Latin hypercube (OLH), see [5]. OLHs can be either created during the problem execution or, if available, loaded from the attached large database of pre-generated OLH designs.

Some of reliability analysis algorithms implemented in STAND take advantage of parallel computing, which now becomes the standard in commercial codes. Thus the tasks performed with STAND can be submitted on a single PC as well as on parallel computers, which should be a default choice for real life problems involving computationally expensive FE simulations.

STAND has been successfully employed in crashworthiness reliability analysis of the welded sheet metal car components. Due to manufacturing imperfections and fatigue deterioration significant number of spot welds may missing in operational vehicle. This effect may significantly reduce strength of an important car component such as thin-walled s-rail. A great number of joints makes precise stochastic modeling of spot welds unreasonable. Thus, it was proposed to model the uncertainty of element connections by adding a random noise to LSF. However reliability analysis problem defined in this way is very difficult to solve. This is not only due to difficulties with assessing failure probability for noisy LSFs, but also due to time consuming FE analysis that is involved. An effective method for solving this class of problems was implemented in STAND, see [4]. In the considered example there were assumed 8 random variables corresponding to metal sheet thicknesses, the initial velocity of impacting mass and material parameters. The failure event was defined as insufficient energy absorption.

The second example demonstrates capabilities of STAND integration with third party FE analysis programs. Influence of geometrical imperfections on the buckling behavior of a cylindrical shell presented in the ABAQUS example problems manual (example 1.2.6.) has been studied. In the original example imperfections are modeled using a linear combination of the eigenvectors of the linear buckling problem. For the purpose of stochastic analysis coefficients of the linear combination are assumed to be random variables. Then it was possible to estimate probability that the buckling load of the investigated cylindrical shell is lower than an assumed value. Application of STAND to the modified example problem of widely used FE code clearly illustrates its integration possibilities.

6. References

[1] M.F. Pellissetti, G.I. Schueller (2006). On general purpose software in structural reliability – an overview. *Structural Safety*, **28**, 3–16.

[2] G.I. Schueller and H.J. Pradlwarter (2006). Computational stochastic structural analysis COSSAN – a software tool, *Structural Safety*, **28**, 68–82.

[3] Y.-T. Wu, Y. Shin, R.H. Sues, A. Cesare~Mark (2006). Probabilistic function evaluation system (PROFES) for reliability-based design, *Structural Safety*, **28**, 164–195.

[4] R. Stocki, P. Tauzowski, J. Knabel. Reliability analysis of a crushed thin-walled s-rail accounting for random spot weld failures, *International Journal of Crashworthiness*, in print

[5] M. Liefvendahl, R. Stocki (2003). A study on algorithms for optimization of Latin hypercubes. *Journal of Statistical Planning and Inference*, **136**, 3231-3247.