Application of Markov chain Monte Carlo method for structural reliability

Krzysztof Kolanek*

Institute of Fundamental Technological Research, Polish Academy of Sciences, Swietokrzyska 21 Str., 00–049 Warsaw, Poland

Abstract

Markov chain Monte Carlo is a method for simulation from a probability density function which is known up to a normalizing constant. Application of this method for evaluation of a probability of failure is presented. Several ideas which facilitate the proposed approach are provided. An illustrative numerical example is given.

Keywords: Reliability analysis; Markov chain Monte Carlo; Normalizing constant; Exponential tilting; Importance sampling

1. Introduction

Evaluation of the probability of failure is essential in a structural reliability analysis. Probability of failure is defined as the integral of probability density function over the region in random variable space, for which failure occurs [1]. Due to the usually high number of random variables in real life applications, numerical integration is inefficient for this problem. In practice, first- or second-order approximation methods (FORM/SORM [1]) are often used to evaluate probability of failure. However, applicability of these methods is limited to problems satisfying certain conditions. An alternative is the Monte Carlo integration. Since a failure event is usually rare, it is common to apply importance sampling in order to facilitate calculations. Some well-developed algorithms for structural reliability are available; however, all of them have certain limitations. Thus, the author found it interesting to investigate application of the so-called Markov chain Monte Carlo method for evaluation of the probability of failure. Markov chain Monte Carlo is any simulation method producing an ergodic Markov chain with a given stationary distribution [2,3]. Concerning the number of samples required for an estimation of expectations these methods are less efficient than importance sampling. However, Markov chain Monte Carlo does not need an additional optimization algorithm, because it explores the random variable space by itself. It is also worth mentioning that during recent years Markov chain Monte Carlo has attracted a great deal of attention from researchers. This has resulted in the fast development of the method and many successful applications in various scientific areas. Yet, the only paper known to the author that presents a slightly different approach to the application of Markov chain Monte Carlo to the evaluation of probability of failure is [4].

2. Structural reliability analysis by importance sampling

The time invariant structural reliability problem is usually defined as follows [1]. Uncertain structural parameters are represented by a real-valued random vector $\mathbf{X} = (X_1, X_2, \ldots, X_n)$, with probability density function $f(\mathbf{x})$. Structural performance with respect to random parameters is reflected by a limit state function $g(\mathbf{x})$. The limit state function is defined to take negative values for parameters for which failure occurs. Thus, the limit state function defines a subset in the random variable space called the failure domain $\Omega_F = \{\mathbf{x} : g(\mathbf{x} \leq 0)\}$. Finally, the probability of failure is defined as

$$P_F = \int_{\Omega_F} f(\mathbf{x}) d\mathbf{x} \tag{1}$$

P_F can be evaluated by means of Monte Carlo integration. Since for engineering structures a small probability of failure is desired, the crude Monte Carlo is inefficient for such problems. Therefore, application of variance

* Tel.: +48 (22) 827 46 92; E-mail: kkolanek@ippt.gov.pl

Computational Fluid and Solid Mechanics 2005
K.J. Bathe (Editor)

reduction techniques like importance sampling is usually attempted. The importance sampling for evaluation of P_F is formulated based on Eq. (1), rewritten as follows:

$$P_F = \int_{\Omega_F} \frac{f(\mathbf{x})}{h(\mathbf{x})} h(\mathbf{x}) d\mathbf{x} = \mathbb{E}_h \left[I(g(\mathbf{x}) \le 0) \frac{f(\mathbf{x})}{h(\mathbf{x})} \right] \tag{2}$$

where $h(\mathbf{x})$ is an importance sampling density, $I(\cdot)$ is the indicator function of the failure domain, and $\mathbb{E}_h$ denotes an expectation operation with respect to the density $h(\mathbf{x})$. Having m independent sample points $\mathbf{x}^{(k)}$, ($k = 1, \ldots, m$) from distribution $h(\mathbf{x})$, the expectation in Eq. (2) can be estimated from

$$\hat{P}_F = \frac{1}{m} \sum_{k=1}^{m} I(g(\mathbf{x}^{(k)}) \le 0) \frac{f(\mathbf{x}^{(k)})}{h(\mathbf{x}^{(k)})} \tag{3}$$

The optimal density function $h(\mathbf{x})$ that minimizes variance of this estimator has the following form:

$$h^*(\mathbf{x}) = \begin{cases} \frac{f(\mathbf{x})}{P_F}, & \text{if } g(\mathbf{x}^{(k)} \le 0) \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

However, this formula is rather formal, since generation of independent random variables requires the knowledge of the term of interest P_F. In practice the distribution, from which samples are produced, is usually chosen to resemble the distribution with density $f(\mathbf{x})$ truncated to the failure domain. On the other hand, if the sample independence requirement is omitted Markov chain Monte Carlo can be used to generate a sample from the density h^*. Obviously, an estimator based on a dependent sample will no longer be optimal. Nevertheless, it seems that for some problems an algorithm based on Markov chain Monte Carlo should be more efficient than the crude Monte Carlo and easier to apply than the importance sampling.

3. Markov chain Monte Carlo for evaluation of the failure probability

This section outlines the Metropolis-Hastings algorithm, which is the most general form of the Markov chain Monte Carlo. However functions specific for reliability analysis are used in the presented formulas. Comprehensive coverage of Markov chain Monte Carlo can be found in the book by Robert et al. [2]. The Metropolis-Hastings algorithm starts with the target density from which is to be simulated h^* in the considered case. The so-called proposal (instrumental) distribution, with a conditional density $q(\mathbf{y}|\mathbf{x})$, is then chosen. To facilitate implementation of the algorithm it should be easy to simulate from $q(\cdot|\mathbf{x})$ and it must be symmetric or (either) explicitly known up to normalizing constant. The Metropolis-Hastings algorithm generates a Markov chain $\mathbf{X}^{(t)}$ ($t = 0, \ldots, T$) with stationary distribution h^* by iterating two steps. In the first step a potential consecutive state of the Markov chain is generated from the proposal distribution

$$\mathbf{Y}_t \sim q(y|\mathbf{x}^{(t)}) \tag{5}$$

where $\mathbf{x}^{(t)}$ is the current state of the chain. In the second step the proposed state is accepted or the current state is repeated according to the following formula:

$$\mathbf{X}^{(t+1)} = \begin{cases} \mathbf{Y}_t & \text{with probability } \alpha(\mathbf{x}^{(t)}, \mathbf{Y}_t) \\ \mathbf{x}^{(t)} & \text{with probability } 1 - \alpha(\mathbf{x}^{(t)}, \mathbf{Y}_t) \end{cases} \tag{6}$$

where

$$\alpha(\mathbf{x},\mathbf{y}) = \min \left\{ \frac{h^*(\mathbf{y})\, q(\mathbf{x}|\mathbf{y})}{h^*(\mathbf{x})\, q(\mathbf{y}|\mathbf{x})}, 1 \right\} \tag{7}$$

The so-called acceptance probability $\alpha(\mathbf{x},\mathbf{y})$ is obviously defined only when $h^*(\mathbf{x}) > 0$. Hence a starting point $\mathbf{x}^{(0)}$ must be located in the failure domain if the target density is given by Eq. (4). Finding such a point might be difficult when the probability of failure is small. It is possible to overcome this problem by modifying the density function $f(\mathbf{x})$. The proposed modification of the target density function, similar to exponential tilting [5], is defined as

$$h_t(\mathbf{x}) = \frac{1}{C_t} \exp(-r(g(\mathbf{x}))) f(\mathbf{x}) \tag{8}$$

where C_t is a normalizing constant and $r(\cdot)$ is a function with properties similar to a penalty function used in constrained optimization [6]

$$r(y) = \begin{cases} 0 & \text{for } y \le 0 \\ > 0 & \text{for } y > 0 \end{cases} \tag{9}$$

The function $r(\cdot)$ should be selected to quickly depress $h_t(\mathbf{x})$ with increasing value of the limit state function g. However, if the selected starting point is located outside the failure domain $h_t(\mathbf{x}^{(0)})$ should have a positive value.

Performance of the algorithm obviously depends on the choice of a proposal density for a given target distribution. Specifying the good proposal distribution can be a difficult task, especially for high dimensional problems. There are several papers on optimality of Markov chain algorithms for normal target density [7,8,9]. Recently, several methods were also developed allowing optimization of a parametric proposal distribution during a chain run [10,11,12]. One of the basic and often used proposal distributions is normal distribution with mean in the most recent state of the chain. The scaling parameter of this distribution can be adjusted or optimized during the run of the chain in order to maximize performance of the algorithm.

A basic Markov chain Monte Carlo estimator for expectations is the usual empirical average $1/T\sum_{t=1}^{T}\tilde{g}(\mathbf{X}^{(t)})$, where $\tilde{g}$ is a function whose expectation is to be evaluated. Unfortunately this estimator is useless for estimation of expectation in Eq.(2). To calculate P_F by the outlined algorithm, it is necessary to apply methods dedicated to evaluation of the normalizing constants of the target distribution. There are several methods available for this purpose; comparison of them can be found in [13].

One of the methods applicable for evaluation of P_F is based on the Laplace approximation [13]. It uses a point comparison of density h^* and a normal approximation to a distribution of the generated Markov chain

$$\hat{P}_{FL} = \frac{h^*(\mathbf{x}^*)}{\phi(\mathbf{x}^*,\mathbf{x}^*,\hat{\mathbf{\Sigma}})} \tag{10}$$

where ϕ is the normal density function, $\mathbf{x}^*$ is the estimated point maximizing $h^*(\mathbf{x})$ and $\hat{\mathbf{\Sigma}}$ is the estimated covariance matrix of $\mathbf{X}^{(t)}$. The other method for computation of normalizing constants takes advantage of the modified importance sampling estimator. The resulting formula for the probability of failure is

$$\hat{P}_{FI} = \frac{1}{T}\sum_{t=1}^{T}\frac{h^*(\mathbf{X}^{(t)})}{\hat{p}(\mathbf{X}^{(t)})} \tag{11}$$

where $\hat{p}$ is an approximated density function of the generated sample. This approximation can be obtained by a normal approximation, a mixture of normals, or a kernel density estimation, for instance. The presented approach differs slightly from the usual importance sampling, because the sampling distribution is evaluated from the sample produced by the Markov chain Monte Carlo algorithm.

It has already been mentioned that some of the proposal points $\mathbf{Y}_t$ are rejected when the chain $\mathbf{X}^{(t)}$ is being generated. Surprisingly, for a multi-normal target distribution, optimal performance of the algorithm is attained when approximately 75% of the proposal points are rejected. Such a waste of effort is difficult to accept, especially when evaluation of the limit state function is expensive. However, some methods utilizing all generated proposals have been suggested [14,15]. The importance sampling estimator proposed in Eq. (11) can be applied to all generated proposal points.

The derivation of an analytical expression for the error of the proposed methodology seems to be a rather difficult problem. Note that it is necessary to take into account dependence of the samples and several approximations used in a sequence to obtain a final estimate. However, standard deviation and bias of the estimators obtained with the proposed methodology can be evaluated by resampling methods like the bootstrap [16]. For estimators using all proposed samples, a simple bootstrap for independent samples can be applied, while methods for resampling of the time series should be applied for estimators using the output of the chain.

4. Numerical example

The presented example is taken from Engelund et al. [17]. Its purpose is to investigate performance of the algorithm for different numbers of dimensions and different probability levels. The limit state function is an n-dimensional hyperplane $g(\mathbf{x}) = \beta n^{1/2} - \sum_{i=1}^{n} x_i$ where random variables X_i, $i = 1, 2, \ldots, n$ are independent, normal distributed variables. The Metropolis-Hastings algorithm with procedure optimizing proposal density was employed [12]. The proposed distribution was normal with a diagonal covariance matrix and a single scaling parameter. The target density was similar to Eq. (8) with linear 'penalty function' $r(y) = 30y$. The chains started from the origin point. Estimates of failure probability relative to the exact value and coefficients of variations of the estimates are shown in Fig. 1. Calculations were made for $\beta = 1$, $\beta = 5$ and $\beta = 10$ and for $n = 2$, $n = 10$ and $n = 30$. The results were obtained for 7000 simulations; the algorithm was implemented in the R environment [18]. High sensitivity of the algorithm to the number of dimensions can be seen.

5. Conclusions

In this article the author outlined how to apply Markov chain Monte Carlo to the evaluation of the probability of failure. It was shown how to take advantage of general statistical methods, rather than detail description of the algorithm. The performance of the algorithm presented in the example is not very good. However, it should be borne in mind that no additional algorithm for localization of the design point was used. Thus, the results refer to total costs of the analysis. Also, a rather basic algorithm was utilized. The application of general methods gives a possibility for easy use of future techniques which would enhance the performance. Moreover, the Metropolis-Hastings algorithm can also be applied for limit state functions with a complicated shape.

References

[1] Melchers RE. Structural Reliability Analysis and Prediction. Chichester: J Wiley & Sons, 1999.
[2] Robert CP, Casella G. Monte Carlo Statistical Methods. New York: Springer, 1999.

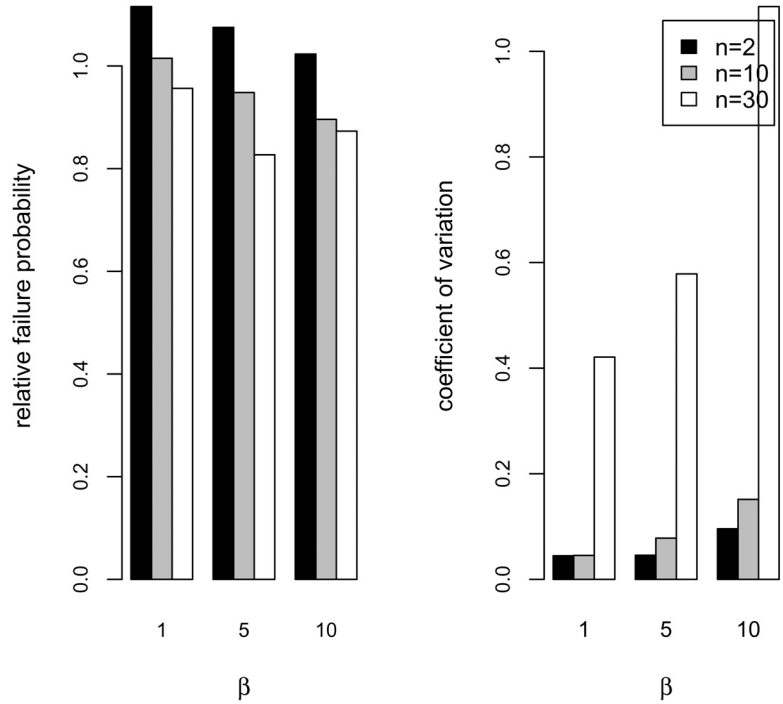

Fig. 1. Sensitivity of the algorithm to the number of dimensions and the reliability level (7000 simulations).

[3] Liu JS. Monte Carlo Strategies in Scientific Computing. New York: Springer, 2001.
[4] Au SK, Beck JL. A new adaptive importance sampling scheme. Struct Safety 1999;21(2):135–158.
[5] Hesterberg TC. Advances in importance sampling. Ph.D. dissertation, Statistics Department, Stanford University, 1988.
[6] Arora JS. Introduction to Optimum Design. New York: McGraw-Hill, 1989.
[7] Gelman A, Roberts GO, Gilks WR. Efficient Metropolis jumping rules. Bayesian Stat 1996;V:599–608.
[8] Roberts GO, Gelman A, Gilks WR. Weak convergence and optimal scaling of random walk Metropolis algorithms. Ann Appl Prob 1997;7:110–120.
[9] Roberts GO, Rosenthal JS. Optimal scaling for various Metropolis-Hastings algorithms. Stat Sci 2001;16:351–367.
[10] Andrieu C, Robert CP. Controlled MCMC for optimal sampling. Unpublished paper, 2001. http://www.ceremade.dauphine.fr/~xian/control.ps.gz
[11] Atchadé YF, Rosenthal JS. On adaptive Markov chain Monte Carlo algorithms. Unpublished paper, 2003. http://probability.ca/jeff/ftpdir/AdaptMH2.ps.Z
[12] Pasarica C, Gelman A. Adaptively scaling the Metropolis algorithm using expected squared jumped distance. Unpublished paper, 2003.
[13] DiCiccio T, Kass R, Raftery A, Wasserman L. Computing Bayes factors by combining simulation and asymptotic approximations. J Am Stat Assoc 1997;92:903–915.
[14] Casella G, Robert CP. Rao-Blackwellisation of sampling schemes. Biometrika 1996;83:81–94.
[15] Tjelmeland H. Using all Metropolis-Hastings proposals to estimate mean values. Department of Mathematical Sciences, Norwegian University of Science and Technology, Trondheim, Norway, 2004.
[16] Efron B, Tibshirani RJ. An Introducton to the Bootstrap. New York: Chapman and Hall, 1993.
[17] Engelund S, Rackwitz R. A benchmark study on importance sampling techiques in structural reliability. Struct Safety 1993;12:255–276.
[18] R Development Core Team. R: A Language and Environment for Statistical Computing. Vienna, Austria: R Foundation for Statistical Computing, 2004. http://www.R-project.org