

## HORSERACING SIMULATION ALGORITHM FOR EVALUATION OF SMALL FAILURE PROBABILITIES

Lambros S. Katafygiotis<sup>1</sup>, Konstantin M. Zuev<sup>2</sup>

<sup>1</sup>The Hong Kong University of Science and Technology  
Hong Kong  
e-mail: lambros@ust.hk

<sup>2</sup>The Hong Kong University of Science and Technology  
Hong Kong  
e-mail: zuev@ust.hk

**Keywords:** Reliability, High dimensions, Markov chain Monte Carlo, Subset simulation

**Abstract.** *Over the past decade, the civil engineering community has ever more realized the importance and perspective of the reliability-based design optimization (RBDO), the selection of a design that satisfies performance constraints with a specified probability. Since then several stochastic simulation algorithms for computing small failure probabilities encountered in reliability analysis of engineering systems have been developed: Subset Simulation [1], Line Sampling [2], The Auxiliary Domain Method [3], ALIS [4], etc.*

*In this paper we propose a novel advanced stochastic simulation algorithm for solving high-dimensional reliability problems, called Horseracing Simulation (HRS). The key idea behind HS is as follows. Although the reliability problem itself is high-dimensional, the limit-state function maps this high-dimensional parameter space into a one-dimensional real line. This mapping transforms a high-dimensional random parameter vector, which may represent the input load as well as the uncertain structural parameters, into a random variable with unknown distribution, that represents the uncertain structural response. It turns out, that the corresponding cumulative distribution function (CDF) of this random variable of interest can be accurately approximated by empirical CDFs constructed from specially designed samples. The accuracy and efficiency of the new method is demonstrated with a real-life wind engineering example.*

## 1 INTRODUCTION

In reliability engineering our task is to calculate the reliability or equivalently the probability of failure of a given structure under uncertain loading conditions. The mathematical models of the uncertain input load  $x$  and the structure response  $g(x)$  are a random vector  $x \in \mathbb{R}^N$  with joint probability density function (PDF)  $\pi_0$  and a function  $g : \mathbb{R}^N \rightarrow \mathbb{R}_+$ , correspondingly. For example, if the structure is a tall building, the stochastic input may represent wind velocities along the building height and the response may represent the maximum roof displacement or the maximum interstory drift (absolute value) under the given wind load.

Define the failure domain  $\Omega \subset \mathbb{R}^N$  as the set of inputs that lead to the exceedance of some prescribed critical threshold  $z^* \in \mathbb{R}_+$ :

$$\Omega = \{x \in \mathbb{R}^N | g(x) > z^*\} \quad (1)$$

In the above example the critical threshold  $z^*$  represents the maximum permissible roof displacement or maximum permissible interstory drift and the failure domain  $\Omega$  represents the set of all wind loads that lead to the exceedance of this threshold.

The structural reliability problem is to compute the probability of failure, that is given by the following expression:

$$p_F = P(x \in \Omega) = \int_{\Omega} \pi_0(x) dx = \int_{\mathbb{R}^N} I_{\Omega}(x) \pi_0(x) dx = E_{\pi_0}[I_{\Omega}] \quad (2)$$

where  $I_{\Omega}$  is the indicator function ( $= 1$  if  $x \in \Omega$ ,  $= 0$  otherwise) and  $E_{\pi_0}$  denotes expectation with respect to the distribution  $\pi_0$ .

Throughout this work we assume we are dealing with probability integrals (2) under the following context:

1. The computation of probability integral (2) is extremely challenging for real-world structures and can be done only in approximate ways. A well established methodology (see, for example, [5]) consists of introducing a one-to-one transformation  $\Upsilon$  between the physical space of variables  $x$  and the standard Gaussian space of variables  $y$  and then computing the probability of failure as  $p_{\Omega} = \int_{\Upsilon(\Omega)} \mathcal{N}(y) dy$ , where  $\mathcal{N}$  denotes the standard Gaussian joint PDF and  $\Upsilon(\Omega)$  is the image of the failure domain in the standard Gaussian space. Abusing the notation, we shall assume that the PDF  $\pi_0$  is the  $N$ -dimensional standard Gaussian distribution, in particular we can evaluate  $\pi_0(x)$  for any given  $x$  and we can generate random samples from  $\pi_0$  efficiently.
2. The function  $I_{\Omega}(x)$  is not explicitly known. Although for any  $x$  we can check whether it is a failure point or not, i.e. calculate the value  $I_{\Omega}(x)$  for a given  $x$ , we cannot obtain an explicit formula.
3. The computational effort for evaluating  $I_{\Omega}(x)$  for each value of  $x$  is assumed to be significant so that it is essential to minimize the number of such function evaluations. In the context of the tall building example the last two statements mean the following. Since the structure is too complex, we cannot directly predict whether a given wind load will cause a damage or not. The only way to do so is by performing a dynamic analysis which by itself is computationally demanding.

4. *The parameter space  $\mathbb{R}^N$  is assumed to be high-dimensional.* As already mentioned, for real-world structures the reliability problem has no exact analytical solution and one has to rely on computer to obtain an approximate value of the failure probability. Any continuous problem must be discretized before it can be treated computationally, and thus the stochastic input load  $x$  is modeled as a random vector in  $\mathbb{R}^N$ . The larger  $N$  the more precisely this discrete model describes the continuous random input. Thus, the assumption of high dimensionality of parameter space comes from the aspiration to use a good discrete model of a real continuous problem. Also, computational algorithms that are applicable in high dimensions are generally more robust. In our examples we shall consider  $N \sim 10^4$ .

Among all procedures developed for estimation of  $p_\Omega$ , a prominent position is held by stochastic simulation methods [1, 2, 3, 4, 6, 8]. In this paper we propose a novel advanced stochastic simulation algorithm, called Horseracing Simulation, and demonstrate its accuracy and efficiency with a real-life example.

## 2 BASIC IDEA OF HORSERACING SIMULATION

Let us start with the discussion of the following auxiliary problem. Let  $z$  be a continuous random variable with PDF  $f$  and CDF  $F$ , which are unknown. Suppose, that we can draw samples from the distribution  $f$ . Our goal is, trying to use as few samples as possible, to approximate  $F$  in some neighbourhood of a given point  $z^* \in \mathbb{R}$ . If the point  $z^*$  is not very far from the median  $\tilde{z}$ , then we can just draw Monte Carlo samples from  $f$  and use the empirical CDF  $F^{(0)}$ , constructed based on these samples, as an approximation of  $F$ , see Fig. 1. However, if the probability  $p = 1 - F(z^*)$  is very small  $p \ll 1$ , then the Monte Carlo method will require a lot of samples in order to get some information about  $F$  in the neighbourhood of  $z^*$ . Therefore, since it is essential to minimize the number of samples, the direct Monte Carlo method is not applicable.

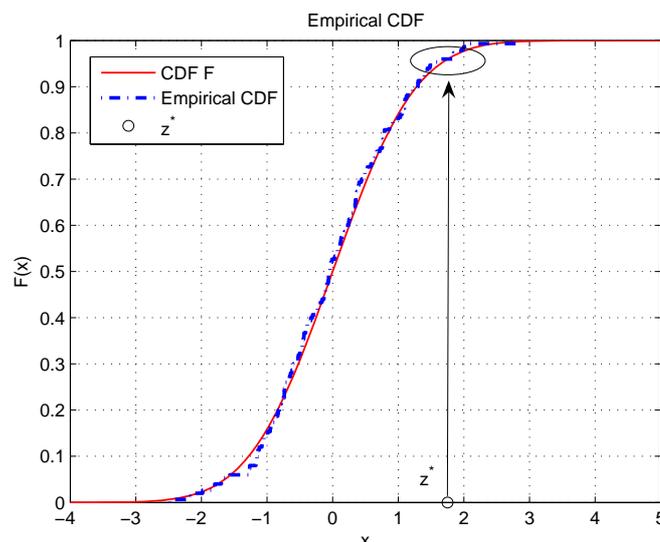


Figure 1: Approximation of the standard Gaussian CDF by empirical CDF, constructed based on Monte Carlo samples.

Assume now, that we can propagate our Monte Carlo samples towards the important region

(neighbourhood of  $z^*$ ). Namely, for any sample  $z^{(0)} \sim f(z)$  we are able to draw samples from the conditional distribution  $f(z|z \geq z^{(0)})$ , for any sample  $z^{(1)} \sim f(z|z \geq z^{(0)})$  we are able to draw samples from the conditional distribution  $f(z|z \geq z^{(1)})$ , etc. It can be proven (see Appendix), that the  $k^{\text{th}}$  random variable  $z^{(k)}$ , defined by this process, has PDF

$$f_k(z) = \frac{(-1)^k}{k!} f(z) [\log(1 - F(z))]^k. \quad (3)$$

It is well-known from the importance sampling theory (e.g. see [7]), that if  $x_1, \dots, x_n$  are independently drawn from a trial distribution  $h$  and the weight  $w_i / \sum_{i=1}^n w_i$ , where  $w_i = f(x_i)/h(x_i)$ , is assigned to each  $x_i$ , then as  $n \rightarrow \infty$  this approach produces a sample that is approximately distributed according to  $f$ . In standard terminology, the trial distribution  $h$ , the weight  $w_i$  and  $w_i / \sum_{i=1}^n w_i$  are called “the importance distribution”, “the importance weight” and “normalized importance weight”, correspondingly.

So, if  $z_1^{(k)}, \dots, z_n^{(k)}$  are independently distributed according to the distribution  $f_k$ , then the weighted samples  $(z_1^{(k)}, w_1^{(k)}), \dots, (z_n^{(k)}, w_n^{(k)})$ , where

$$w_i^{(k)} \propto \frac{f(z_i^{(k)})}{f_k(z_i^{(k)})} \propto \frac{1}{[\log(1 - F(z_i^{(k)}))]^k}, \quad (4)$$

are approximately distributed according to  $f$ . Therefore, we can construct the empirical CDF  $F^{(k)}$  based on  $\{(z_i^{(k)}, w_i^{(k)})\}_{i=1}^n$  and use it for the updating of the empirical CDF  $F^{(0)}$ , which was constructed based on Monte Carlo samples  $\{z_i^{(0)}\}_{i=1}^n$ . Note, that the importance weights in (4) depend explicitly only on the CDF  $F$  that we want to approximate and do not depend on the unknown PDF  $f$ .

The above discussion suggests the following scheme of an algorithm (which is a prototype of the Horseracing Simulation algorithm) for the approximation of  $F$ .

#### HORSERACING SIMULATION SCHEME

- I. Sample  $z_1^{(0)}, \dots, z_n^{(0)}$  from  $f_0 = f$ ,  
Set  $k = 0$ .
- II. Construct the empirical CDF  $F^{(k)}$  based on  $\{z_i^{(k)}\}_{i=1}^n$ .

While the stopping criterion  $C(z^*)$  is not fulfilled do:

- III. Sample  $z_i^{(k+1)}$  from  $f_0(z|z \geq z_i^{(k)})$  for each  $i = 1 \dots n$ .
- IV. Construct the empirical CDF  $G^{(k+1)}$  based on  $\{(z_i^{(k+1)}, w_i^{(k+1)})\}_{i=1}^n$ .
- V. Update CDF  $F^{(k)}$  to  $F^{(k+1)}$ ,  $(F^{(k)}, G^{(k+1)}) \rightsquigarrow F^{(k+1)}$ ,  
Set  $k=k+1$ . Go to III.

Of course the steps of this algorithm should be specified and the stopping criterion  $C(z^*)$  should be properly chosen. Then we can naturally expect that in some neighbourhood of  $z^*$

$$F^{(k)} \approx F. \quad (5)$$

Before we explain how this scheme can help to solve the reliability problem, let us strike some life into notation and explain the name origin for this algorithm. One can think of  $z_1, \dots, z_n$  as about horses participating in a race, where  $z_i^{(k)}$  denotes the position of the  $i$ th horse at time instant  $k$ . The race is over when the finishing rule given by  $C(z^*)$  is fulfilled, for example, when one of the horses ( $z_i^{(k)}$ ) reaches the finish line ( $z_i^{(k)} \geq z^*$ ).

Let us now relate the Horseracing Simulation scheme with the reliability problem. Recall, that the structural reliability problem is to compute the probability of failure, that is given by (2). Limit-state function  $g : \mathbb{R}^N \rightarrow \mathbb{R}_+$  maps the high-dimensional parameter space into a one-dimensional real line. This mapping transforms the high-dimensional random parameter vector  $x$  into a random variable  $z = g(x)$ , which represents the structural response. This is schematically shown in Fig. 2.

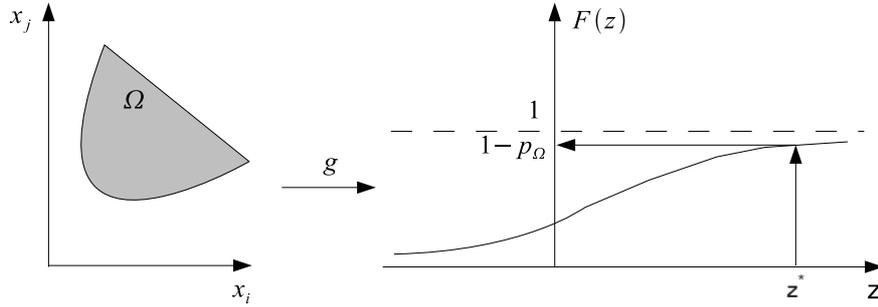


Figure 2: Dimension reduction induced by the limit-state function.

Let  $f$  and  $F$  be the PDF and CDF of  $z$  respectively, then the probability of failure in (2) can be rewritten as follows:

$$p_{\Omega} = \int_{z^*}^{\infty} f(z) dz = 1 - F(z^*). \quad (6)$$

If the limit-state function  $g$  is continuously differentiable, then

$$f(z) = \int_{\{x: z=g(x)\}} \frac{\pi_0(x)}{\|\nabla g(x)\|} dV, \quad (7)$$

where  $\nabla g(x) = (\partial g/\partial x_1, \dots, \partial g/\partial x_n)$  is the gradient of the limit-state function, integration is taking over the  $(N - 1)$ -dimensional surface  $\{x : z = g(x)\}$  and the differential volume  $dV$  must be replaced by a parametrization of this surface for a particular calculation. Although for any given  $x$  we can calculate the value  $g(x)$ , we cannot obtain any other information such as explicit formula, gradient, and so on. As a consequence, neither  $f$  nor  $F$  is known. Hence, the limit-state function allows us to shift the difficulty of the reliability problem from geometry (in (2) we have to calculate the high-dimensional integral over a complex domain that is only known implicitly) to probability ((6) is just a one-dimensional integral, yet of an unknown function), see Fig 2.

Thus, in order to use (6) for failure probability estimation one should find an approximation of the CDF  $F$  of the random variable  $z$ . In the rest part of this paper we shall show how the Horseracing Simulation scheme can be successfully used for this purpose.

### 3 IMPLEMENTATION ISSUES

In this section we discuss the details of the proposed Horseracing Simulation scheme in the context of the reliability problem.

#### 3.1 Sampling

According to step I of the scheme, we have to sample from the distribution  $f_0 = f$ . In other words, we have to define the initial positions of the horses participating in a race. Although the distribution  $f_0$  is unknown, it is very simple to get a sample from it. Namely, Monte Carlo samples  $x_1^{(0)}, \dots, x_n^{(n)} \sim \pi_0$ , being transformed by the limit-state function, will automatically provide independent samples from  $f_0$ :

$$z_1^{(0)} = g(x_1^{(0)}), \dots, z_n^{(0)} = g(x_n^{(n)}) \sim f_0. \quad (8)$$

Based on these samples  $F^0$  is constructed, according to the step II.

Next, according to step III of the Horseracing Simulation scheme, we have to sample from  $f_0(z|z \geq z_i^{(k)})$ , where  $z_i^{(k)} \sim f_k$ , i.e., we need to find the position of the  $i$ th horse at time  $k + 1$ . The main idea is the same as in step 1: to sample in the high-dimensional parameter space and then apply transformation, generated by the limit-state function.

Let  $x_i^{(k)}$  be one of the previously generated samples, that corresponds to  $z_i^{(k)}$ , i.e.,  $g(x_i^{(k)}) = z_i^{(k)}$ . Define the subset  $\Omega_{z_i^{(k)}} \subset \mathbb{R}^N$  as follows:

$$\Omega_{z_i^{(k)}} = \{x \in \mathbb{R}^N \mid g(x) \geq z_i^{(k)}\}. \quad (9)$$

Note, that  $x_i^{(k)}$  belongs to the boundary of  $\Omega_{z_i^{(k)}}$ , i.e.  $x_i^{(k)} \in \partial\Omega_{z_i^{(k)}}$ . It is clear, that if  $x$  is sampled from the conditional distribution  $\pi_0(x|x \in \Omega_{z_i^{(k)}})$ , then  $z = g(x)$  is automatically distributed according to  $f_0(z|z \geq z_i^{(k)})$ . So, the problem of sampling from  $f_0(z|z \geq z_i^{(k)})$  reduces to the sampling from  $\pi_0(x|x \in \Omega_{z_i^{(k)}})$ . It turns out, that the latter task can be done by using the modified Metropolis-Hastings (MMH) algorithm [1].

Let  $\mathbb{R}_{z_i^{(k)}}$  denote the half-line in front of  $z_i^{(k)}$ ,

$$\mathbb{R}_{z_i^{(k)}} = g(\Omega_{z_i^{(k)}}) = \{z \in \mathbb{R} \mid z \geq z_i^{(k)}\}, \quad (10)$$

and  $Z_{z_i^{(k)}}^{(t)}$  denote the set of all horse positions at time  $t$ , which are in front of  $z_i^{(k)}$ :

$$Z_{z_i^{(k)}}^{(t)} = \{z_j^{(t)}\}_{j=1}^n \cap \mathbb{R}_{z_i^{(k)}} = \{z_j^{(t)} \mid z_j^{(t)} \geq z_i^{(k)}, j = 1, \dots, n\}, \quad t = 1, \dots, k. \quad (11)$$

All samples from  $Z_{z_i^{(k)}}^{(t)}$  are distributed according to the conditional distribution  $f_t(z|z \geq z_i^{(k)})$ .

Therefore, the weighted samples  $\{(z_j^{(t)}, w_j^{(t)}) \mid z_j^{(t)} \in Z_{z_i^{(k)}}^{(t)}\}$ , where

$$w_j^{(t)} \propto \frac{f_0(z_j^{(t)})}{f_t(z_j^{(t)})} \propto \frac{1}{[\log(1 - F(z_j^{(t)}))]^t}, \quad (12)$$

are approximately distributed according to the conditional distribution  $f_0(z|z \geq z_i^{(k)})$ . Note, that instead of the unknown CDF  $F$  in (12) we can use its approximation  $F^{(k)}$ , obtained in step V

(or in step II) of the scheme. So, at time  $k$  we have  $k+1$  sets of weighted samples approximately drawn from the conditional distribution  $f_0(z|z \geq z_i^{(k)})$ . These sets are schematically shown in Fig. 3.

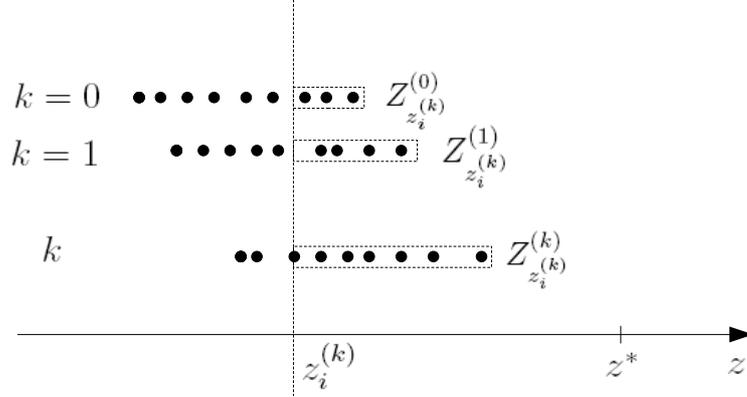


Figure 3: Conditionally distributed samples.

The following algorithm can be used for sampling from  $f_0(z|z \geq z_i^{(k)})$ :

#### SAMPLING ALGORITHM

1. Select  $Z_{z_i^{(k)}}^{(t_0)}$  from  $Z_{z_i^{(k)}}^{(t)}$ ,  $t = 1, \dots, k$  according to their sample sizes  $\chi_{z_i^{(k)}}^{(t)} = \#(Z_{z_i^{(k)}}^{(t)})$ .
2. Select  $z_{j_0}^{(t_0)}$  from  $Z_{z_i^{(k)}}^{(t_0)}$  according to the weights given by (12).
3. Take previously generated sample  $x_{j_0}^{(t_0)}$ , that corresponds to  $z_{j_0}^{(t_0)}$ , i.e.  $g(x_{j_0}^{(t_0)}) = z_{j_0}^{(t_0)}$ , and perform the MMH update  $x_{j_0}^{(t_0)} \rightarrow \hat{x}$  with invariant distribution  $\pi_0(x|x \in \Omega_{z_i^{(k)}})$ .
4. Set  $z_i^{(k+1)} = g(\hat{x})$ .

The way we select  $z_{j_0}^{(t_0)}$  guarantees that it is distributed according to  $f_0(z|z > z_i^{(k)})$ . Therefore, the corresponding sample  $x_{j_0}^{(t_0)}$  has conditional distribution  $\pi_0(x|x \in \Omega_{z_i^{(k)}})$ . Since the MMH update preserves the invariant distribution,  $\hat{x}$  has distribution  $\pi_0(x|x \in \Omega_{z_i^{(k)}})$  as well. From the latter in turn, it follows, that  $z_i^{(k+1)} = g(\hat{x})$  is distributed according to  $f_0(z|z \geq z_i^{(k)})$ . The proposed sampling algorithm is schematically shown in Fig. 4.

### 3.2 Construction of the empirical CDF and its updating

In step II of the Horseracing Simulation scheme, we have to construct a zero<sup>th</sup> approximation  $F^{(0)}$  of the CDF of interest  $F$ , based on the Monte Carlo samples  $z_1^{(0)}, \dots, z_n^{(0)} \sim f_0$ . For this purpose we use the following piecewise linear approximation:

$$F^{(0)}(z) = \frac{z}{n(z_{i+1}^{(0)} - z_i^{(0)})} + \frac{(2i-1)z_{i+1}^{(0)} - (2i+1)z_i^{(0)}}{2n(z_{i+1}^{(0)} - z_i^{(0)})}, \quad \text{for } z \in [z_i^{(0)}, z_{i+1}^{(0)}], \quad (13)$$

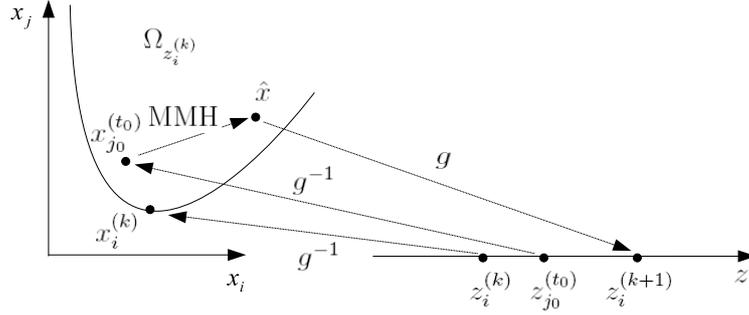


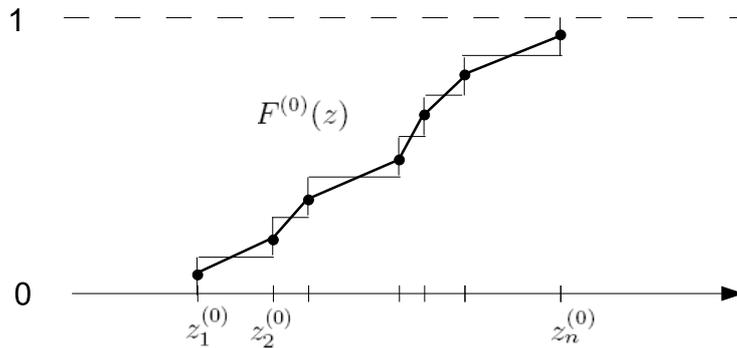
Figure 4: Sampling algorithm.

where  $i = 1, \dots, n - 1$ . Samples  $z_1^{(0)}, \dots, z_n^{(0)}$  are assumed to be ordered in (13), so that  $z_1^{(0)} < z_2^{(0)}, \dots, < z_n^{(0)}$ . The zeroth approximation  $F^{(0)}$  is shown in Fig. 5. Note, that  $F^{(0)}$  is not defined for  $z \in (-\infty, z_1^{(0)}) \cup (z_n^{(0)}, \infty)$ .

Next, in step IV we have to construct the empirical CDF  $G^{(k+1)}$  based on the weighted samples  $\{(z_i^{(k+1)}, w_i^{(k+1)})\}_{i=1}^n$ . If the weights  $\{w_i^{(k+1)}\}_{i=1}^n$  are given, then we can define the empirical CDF  $G^{(k+1)}$  in a similar way as the zeroth approximation:

$$G^{(k+1)}(z) = \frac{w_i^{(k+1)} + w_{i+1}^{(k+1)}}{2(z_{i+1}^{(k+1)} - z_i^{(k+1)})} z + \frac{\left(2 \sum_{j=1}^i w_j^{(k+1)} - w_i^{(k+1)}\right) z_{i+1}^{(k+1)} - \left(2 \sum_{j=1}^i w_j^{(k+1)} + w_{i+1}^{(k+1)}\right) z_i^{(k+1)}}{2(z_{i+1}^{(k+1)} - z_i^{(k+1)})}, \quad (14)$$

for  $z \in [z_i^{(k+1)}, z_{i+1}^{(k+1)}]$  and  $i = 1, \dots, n - 1$ . Samples  $z_1^{(k+1)}, \dots, z_n^{(k+1)}$  are assumed to be ordered in (14), so that  $z_1^{(k+1)} < z_2^{(k+1)}, \dots, < z_n^{(k+1)}$ . Note, that if all weights are equal, i.e.  $w_i^{(k+1)} = 1/n$ , then (14) becomes (13).


 Figure 5: The zero<sup>th</sup> approximation.

In order to use (14), the weights  $w_1^{(k+1)}, \dots, w_n^{(k+1)}$  should be calculated. Since the samples  $z_1^{(k+1)}, \dots, z_n^{(k+1)}$  are approximately distributed according to  $f_{k+1}$ , their weights are given by (12), where  $t = k + 1$  and  $F$  is replaced by its approximation  $F^{(k)}$ , obtained in step V (or in step II). However, the approximation  $F^{(k)}$ , which was constructed using samples

$\{(z_1^{(t)}, \dots, z_n^{(t)})\}_{t=1}^k$ , is defined only for  $z \in [\min_i z_i^{(0)}, \max_i z_i^{(k)}]$ . So, in order to calculate the weights of the “fastest” horses, i.e. of  $z_i^{(k+1)}$  such that  $z_i^{(k+1)} > \max_i z_i^{(k)}$ , we have to interpolate  $F^{(k)}$  on the interval  $[\max_i z_i^{(k)}, \max_i z_i^{(k+1)}]$ . Thus, the weights  $w_1^{(k+1)}, \dots, w_n^{(k+1)}$  are defined as follows:

$$w_i^{(k+1)} \propto \begin{cases} \frac{1}{[\log(1-F^{(k)}(z_i^{(k+1)}))]^{k+1}}, & \text{if } z_i^{(k+1)} \leq \max_i z_i^{(k)}; \\ \frac{1}{[\log(1-F_{int}^{(k)}(z_i^{(k+1)}))]^{k+1}}, & \text{if } z_i^{(k+1)} > \max_i z_i^{(k)}, \end{cases} \quad (15)$$

where  $F_{int}^{(k)}$  is the interpolation of the CDF  $F^{(k)}$ . According to the notation introduced in the sampling algorithm,  $\chi_{\max_i z_i^{(k)}}^{(k+1)}$  is the number of samples  $\{z_i^{(k+1)}\}_{i=1}^n$  that are larger than  $\max_i z_i^{(k)}$ . Let  $z_{j_0}^{(k+1)}$  be the smallest from such samples,  $z_{j_0}^{(k+1)} = \min_i \{z \in Z_{\max_i z_i^{(k)}}^{(k+1)}\}$ . Then, the interpolation of the CDF  $F^{(k)}$  is defined as the piecewise linear function shown in Fig. 6. Note, that  $\chi_{\max_i z_i^{(k)}}^{(k+1)} = n - j_0 + 1$ .

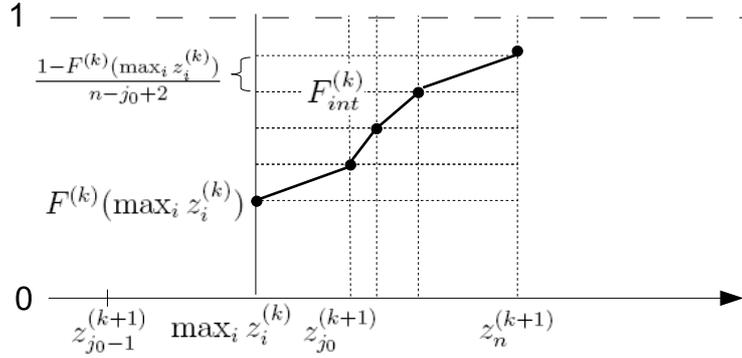


Figure 6: The interpolation of  $F^{(k)}$ .

Finally, in step V we need to update the CDF  $F^{(k)}$ , using new information provided by  $G^{(k+1)}$ , and construct a new approximation  $F^{(k+1)}$  of the CDF  $F$ . Suppose for convenience, that samples  $z_1^{(t)}, \dots, z_n^{(t)}$  are ordered for each  $t = 1, \dots, k+1$  as follows:

$$\begin{aligned} z_1^{(0)} &< z_2^{(0)}, \dots, < z_n^{(0)}, \\ z_1^{(1)} &< z_2^{(1)}, \dots, < z_n^{(1)}, \\ &\dots \\ z_1^{(k+1)} &< z_2^{(k+1)}, \dots, < z_n^{(k+1)}. \end{aligned} \quad (16)$$

The CDF  $G^{(k+1)}$  is not defined on the interval  $[z_1^{(0)}, z_1^{(k+1)}]$ . This means, that at time  $t = k+1$  we do not get any new information about CDF  $F$  on this interval relative to the information available at time  $t = k$ , so  $F^{(k+1)}(z) = F^{(k)}(z)$  for  $z \in [z_1^{(0)}, z_1^{(k+1)}]$ . On the interval  $[z_1^{(k+1)}, z_n^{(0)}]$ , the approximation  $F^{(k)}$  is constructed using  $k+1$  sets of samples  $\{z_i^{(0)}\}_{i=1}^n, \dots, \{z_i^{(k)}\}_{i=1}^n$ , while the approximation  $G^{(k+1)}$  is based only on one set  $\{z_i^{(k+1)}\}_{i=1}^n$ . Therefore, it is natural to define a new approximation  $F^{(k+1)}(z) = ((k+1)F^{(k)}(z) + G^{(k+1)}(z))/(k+2)$ , for  $z \in [z_1^{(k+1)}, z_n^{(0)}]$ .

Using this line of reasoning, we define the new approximation  $F^{(k+1)}$  as follows:

$$F^{(k+1)}(z) = \begin{cases} F^{(k)}(z), & \text{for } z \in [z_1^{(0)}, z_1^{(k+1)}]; \\ \frac{(k+1)F^{(k)}(z) + G^{(k+1)}(z)}{k+2}, & \text{for } z \in [z_1^{(k+1)}, z_n^{(0)}]; \\ \frac{kF^{(k)}(z) + G^{(k+1)}(z)}{k+1}, & \text{for } z \in (z_n^{(0)}, z_n^{(1)}]; \\ \dots, & \dots \\ \frac{F^{(k)}(z) + G^{(k+1)}(z)}{2}, & \text{for } z \in (z_n^{(k-1)}, z_n^{(k)}]; \\ G^{(k+1)}(z), & \text{for } z \in (z_n^{(k)}, z_n^{(k+1)}]. \end{cases} \quad (17)$$

However, since some of the weights in (15) are calculated using the interpolation function  $F_{int}^{(k)}$ , the CDF  $G^{(k+1)}$  approximates not exactly  $F$ . More precisely,  $G^{(k+1)}$  can be decomposed as follows:

$$G^{(k+1)} = G_F^{(k+1)} + G_{int}^{(k+1)}, \quad (18)$$

where  $G_F^{(k+1)}$  is an approximation of  $F$  and  $G_{int}^{(k+1)}$  is a perturbation due to the slightly incorrect weights. Therefore,  $F^{(k+1)}$ , given by (17), also approximates not exactly  $F$  due to the perturbation term  $G_{int}^{(k+1)}$ . Note, that the norm  $\|G_{int}^{(k+1)}\|$  is a decreasing function of  $\chi_{\max_i z_i^{(k)}}^{(k+1)}$ : the less number of samples with weights calculated with  $F_{int}^{(k)}$  in (15), the less the norm  $\|G_{int}^{(k+1)}\|$ . So, if the number of such samples is relatively small, which is the case in applications, we can assume, that the norm  $\|G_{int}^{(k+1)}\|$  is small enough. In order to completely eliminate the influence of  $G_{int}^{(k+1)}$ , we propose the following iterative updating algorithm:

#### UPDATING ALGORITHM

1. Set  $s = 1$ ,  $\varepsilon = 1$ ,  
Define  $H_s$  according to (17).

While the error  $\varepsilon > \varepsilon_0$  do:

2. Recalculate the weights  $\{w_i^{(k+1)}\}_{i=1}^n$  using (15) with  $H_s$  instead of  $F^{(k)}$  and  $F_{int}^{(k)}$ .
3. Recalculate  $G^{(k+1)}$  using (14) with new weights.
4. Define  $H_{s+1}$  according to (17).
5. Recalculate the error:

$$\varepsilon = \max_{z \in \{z_i^{(t)}\}_{i=1, t=1}^{n, k+1}} \left\{ \left| \frac{H_{s+1}(z) - H_s(z)}{1 - H_s(z)} \right| \right\} \quad (19)$$

Set  $s = s + 1$ .

End while

6. Set  $F^{(k+1)} = H_s$ .

The error  $\varepsilon$ , defined in (19), describes the relative change in two successive iterations  $H_s$  and  $H_{s+1}$ . Note, that  $\varepsilon$  is more sensitive to the changes in the important region, where  $H_s$  is close to 1. When this error is smaller than some prescribed threshold  $\varepsilon_0$  (in the further example the value  $\varepsilon_0 = 0.01$  is used) we take the last  $H_s$  as the new approximation  $F^{(k+1)}$  of the CDF  $F$ .

### 3.3 Stopping criterion

The stopping criterion  $C(z^*)$  plays a very important role in the Horseracing Simulation algorithm. As it was already mentioned, one of the possible choices for  $C(z^*)$  is the following rule: the race is over when at least one of the horses ( $z_i^{(k)}$ ) reaches the finish line ( $z_i^{(k)} \geq z^*$ ). The main advantage of this rule is that it allows to obtain the estimate of the failure probability  $p_\Omega = 1 - F(z^*)$  with the minimum possible computation effort (as soon as we reach the threshold  $z^*$  we stop the algorithm). However, this rule has a serious drawback: the estimate may be very inaccurate. Indeed, if, for instance,  $p_\Omega = 0.01$  and we use  $n = 100$  samples, then in average 1 out of 100 Monte Carlo samples  $z_1^{(0)}, \dots, z_n^{(0)}$  will be a failure sample. In this case, the estimate for the failure probability will have coefficient of variation  $\delta = \sqrt{(1 - p_\Omega)/np_\Omega} \approx 1$ .

Another natural candidate for the stopping criterion is the following rule: the race is over when  $r\%$  of horses reach  $z^*$ . For a target probability level of  $10^{-2}$  to  $10^{-5}$ , choosing  $r = 10\%$  is found to yield good efficiency.

The Fig. 7 summarizes the Horseracing Simulation Scheme and the discussed implementation issues into the Horseracing Simulation Algorithm.

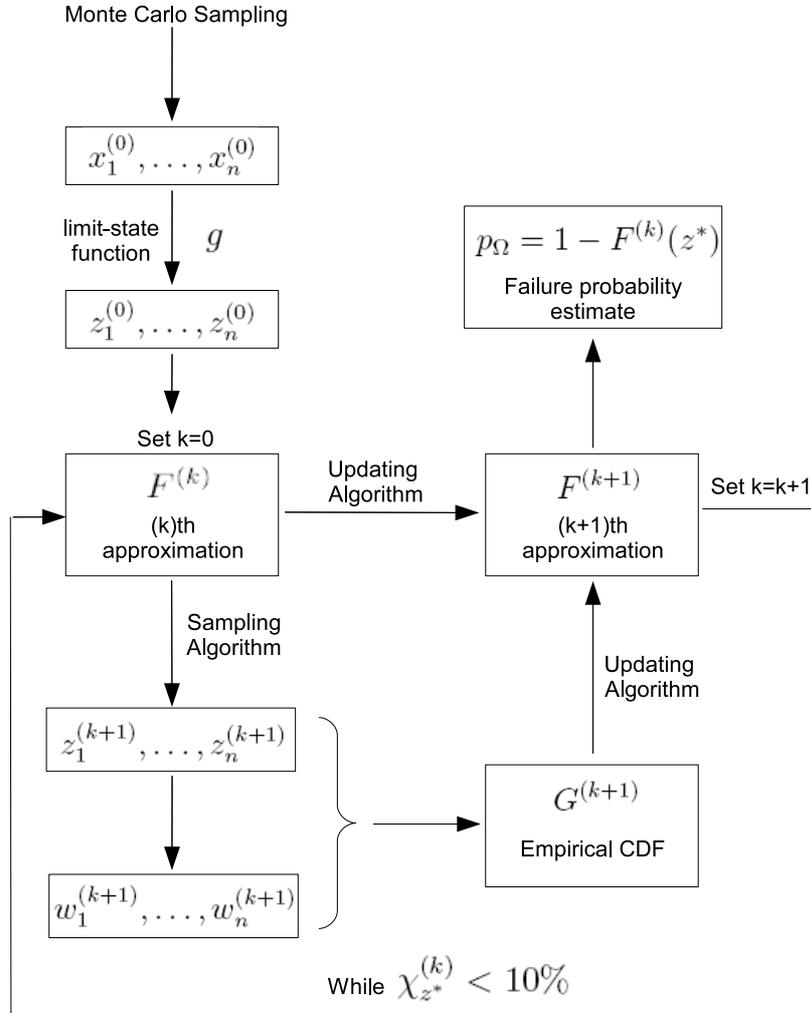


Figure 7: Horseracing Simulation Algorithm.

## 4 EXAMPLE

In this section we demonstrate the efficiency and accuracy of the Horseracing Simulation algorithm with a real-life example which is taken from [8].

### 4.1 CAARC standard tall building model

We consider an along-wind excited steel building as showing in Fig. 8, which has the same geometric shape as the Commonwealth Advisory Aeronautical Research Council (CAARC) standard tall building model [9]. A 45-story, 10-bay by 15-bay rectangular tubular framework is used to model this building. With story height of 4 m and bay width of 3 m, the building has a total height of 180 m and a rectangular floor with dimension 30 m by 45 m. Each floor is assumed to be rigid and has lumped swaying mass of  $6.75 \times 10^5$  kg and rotational mass moment of inertia of  $1.645 \times 10^8$  kg.m<sup>2</sup> at the geometric center of the floor. The members of beams and columns have standard AISC steel sections, and the details of the design are presented in Table 1. With the above configurations, the established building model has the following first three modal frequencies: 0.197 Hz, 0.251 Hz and 0.422 Hz.

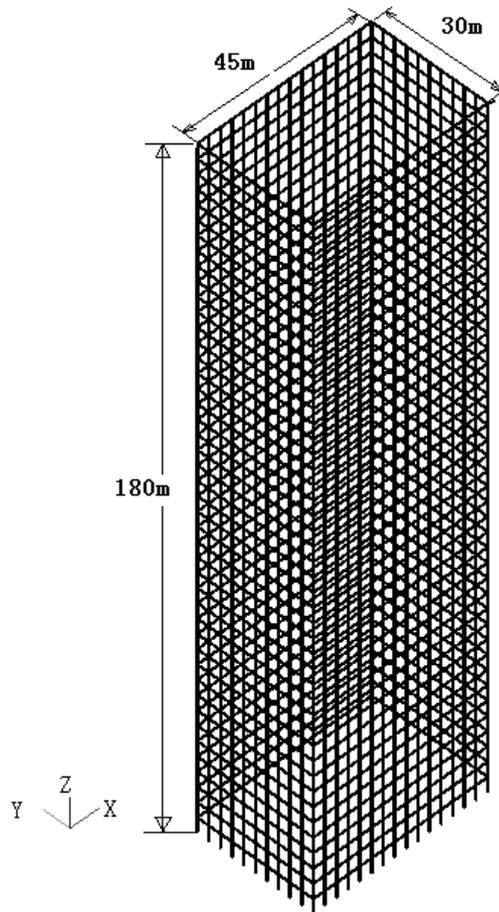


Figure 8: CAARC standard tall building model.

Floor zone	Column members	Beam members
1 ~ 9F	W14X550	W30X357
10 ~ 18F	W14X500	W30X326
19 ~ 27F	W14X370	W30X292
28 ~ 36F	W14X257	W30X261
37 ~ 45F	W14X159	W30X221

Table 1: Design of column members and beam members.

Excitation	Acting height (m)	Acting area (m <sup>2</sup> )
$U_1(t)$	24	45 × 45
$U_2(t)$	68	45 × 45
$U_3(t)$	112	33.75 × 45
$U_4(t)$	136	22.5 × 45
$U_5(t)$	156	22.5 × 45
$U_6(t)$	176	11.25 × 45

Table 2: Acting heights and acting areas of 6 excitation forces in the discretization scheme.

## 4.2 Wind excitation

The along-wind excitation in the  $Y$ -direction of the building is considered. In our example the excitation field is discretized using  $N_u = 6$  excitation forces  $U_1(t), \dots, U_{N_u}(t)$ . The acting heights and acting areas for this discretization scheme are shown in Table 2, and the discretized excitation field is schematically shown in Fig. 9.

At a given point located at height  $h_j$  from the ground, the wind velocity is

$$V_j(t) = \bar{V}_j + v_j(t), \quad (20)$$

where  $\bar{V}_j$  is the mean wind speed and  $v_j(t)$  is the fluctuating component of the wind velocity.

According to the Hong Kong wind code, the mean wind speed  $\bar{V}_j$  (m/s) is given by the power law [10]:

$$\bar{V}_j = 41 \left( \frac{h_j}{180} \right)^{0.25}, \quad j = 1, \dots, N_u. \quad (21)$$

The generation of the fluctuating components is carried out by simulation of an  $N_u$ -variate zero-mean stationary stochastic vector process  $v(t) = [v_1(t), \dots, v_{N_u}(t)]^T$  using the spectral representation method [11, 12, 13, 14]. In this method, the stochastic vector process is simulated using its cross-power spectral density matrix

$$S^0(\omega) = \begin{pmatrix} S_{11}^0(\omega) & \dots & S_{1N_u}^0(\omega) \\ \vdots & & \vdots \\ S_{N_u1}^0(\omega) & \dots & S_{N_uN_u}^0(\omega) \end{pmatrix}. \quad (22)$$

The cross spectral density matrix  $S^0(\omega)$  is modeled by formulas proposed by Davenport in [15, 16]. Namely, the power spectral density function  $S_{jj}^0(\omega)$  of  $v_j(t)$ ,  $j = 1, \dots, N_u$ , is given by

$$S_{jj}^0(\omega) = \frac{\bar{V}_j^2 K^2}{\left( \ln \frac{h_j}{h_0} \right)^2} \frac{8\pi a(\omega)^2}{\omega(1 + a(\omega)^2)^{4/3}}, \quad (23)$$

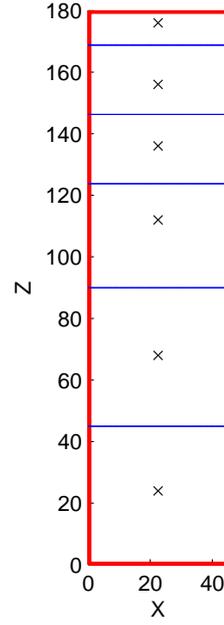


Figure 9: Discretized excitation filed.

$$a(\omega) = \frac{600\omega}{\pi \bar{V}_{10}}, \quad (24)$$

where  $\omega$  (rad/s) is the frequency,  $K = 0.4$  is Von Karman's constant,  $h_0 = 0.05$  m is the roughness length, and  $\bar{V}_{10} = 19.9$  m/s is the mean wind velocity at height of 10 m. The cross-power spectral density function  $S_{jk}^0(\omega)$  of  $v_j(t)$  and  $v_k(t)$  is given by

$$S_{jk}^0(\omega) = \sqrt{S_{jj}^0(\omega)S_{kk}^0(\omega)}\gamma_{jk}(\omega), \quad j, k = 1, \dots, N_u, j \neq k, \quad (25)$$

$$\gamma_{jk}(\omega) = \exp\left(-\frac{\omega}{2\pi} \frac{C_h |h_j - h_k|}{0.5(\bar{V}_j + \bar{V}_k)}\right), \quad (26)$$

where  $\gamma_{jk}(\omega)$  is the coherence function between  $v_j(t)$  and  $v_k(t)$ , and  $C_h$  is a constant that can be set equal to 10 for structural design purposes [10].

To perform the generation of the wind velocity fluctuations the cutoff frequency is taken as  $\omega_c = 0.8\pi$  rad/s, so that the ratio  $r_c$  of the neglected power spectrum content over the total content is less than 10% for all components  $S_{jk}^0(\omega)$ ,  $j, k = 1, \dots, N_u$ . The frequency step is set equal to  $\Delta\omega = \pi/900$ , therefore, the period  $T_v = 4\pi/\Delta\omega$  of the fluctuating wind velocity components  $v(t)$  is 3600 s.

The wind excitation forces  $U_j(t)$ ,  $j = 1, \dots, N_u$  can be expressed as follows:

$$U_j(t) = \frac{1}{2}\rho A_j V_j(t)^2 = \frac{1}{2}\rho A_j (\bar{V}_j + v_j(t))^2, \quad (27)$$

where  $\rho$  is the air density, taken to be  $1.2$  kg/m<sup>3</sup>, and  $A_j$  is the area upon which the discretized force  $U_j(t)$  is assumed to act (see Table 2).

### 4.3 Geometric description of the failure domain

From the above chosen parameters, it follows that the number of standard gaussian random variables involved in the simulation of wind excitation is

$$N = 2 \times N_u \times N_\omega = 2 \times N_u \times \omega_c / \Delta\omega = 8640. \quad (28)$$

In other words, the failure domain  $\Omega$  is a subset of a high-dimensional parameter space  $\Omega \subset \mathbb{R}^N$ , where  $N = 8640$ .

In this example we assume that the displacement response  $Y(t)$  at the top floor of the building is of interest. The relationship between the response  $Y(t)$  and the excitation forces  $U_j$ ,  $j = 1, \dots, N_u$  is given by

$$Y(t) = \sum_{j=1}^{N_u} \int_0^\infty q_j(t, \tau) U_j(\tau) d\tau, \quad (29)$$

where  $q_j(t, \tau)$  is the response function for  $Y(t)$  at time  $t$  due to a unit impulse excitation for  $U_j$  at time  $\tau$ . We assume that the system starts with zero initial conditions, is time-invariant, i.e.,  $q_j(t, \tau) = q_j(t - \tau)$ , and is causal, i.e.,  $q_j(t, \tau) \equiv 0$  for  $t < \tau$ , so that (29) can be rewritten as follows:

$$Y(t) = \sum_{j=1}^{N_u} \int_0^t q_j(t - \tau) U_j(\tau) d\tau. \quad (30)$$

The required impulse response functions  $q_1(t), \dots, q_{N_u}(t)$  are obtained through  $N_u$  dynamic analyses of the established finite element model of the building using the software SAP 2000.

Summarizing the above discussion, the simulation scheme is shown in Fig. 10.

The failure event is defined as the response  $Y(t)$  exceeding in magnitude a specified threshold  $z^*$  within one hour, i.e., the assumed duration time is  $T = 3600$  s. This duration time is conventionally used in wind engineering, for consistence with the duration of actual strong winds. Thus, in the discrete time formulation, where the sampling time interval is chosen to be  $\Delta t = 0.01$  s and the number of time instants is  $N_t = T/\Delta t = 3.6 \cdot 10^5$ , the failure domain  $\Omega \subset \mathbb{R}^N$  is defined as follows:

$$\Omega = \bigcup_{i=1}^{N_t} \{x \in \mathbb{R}^N : |Y(i)| > z^*\}. \quad (31)$$

Thus, in the space of standard normal random variables, the failure domain  $\Omega$  is a union of  $2N_t$  elementary failure domains:  $\{x \in \mathbb{R}^N : Y(i) > z^*\}$  and  $\{x \in \mathbb{R}^N : Y(i) < -z^*\}$ , for  $i = 1, \dots, N_t$ . The limit-state function is given by

$$g(x) = \max\{|Y(i)|, i = 1, \dots, N_t\}. \quad (32)$$

For each sample  $x \in \mathbb{R}^N$  a dynamic analysis is required in order to evaluate the corresponding value  $g(x)$  of the limit-state function. We refer to the total number of such dynamic analyses (or, equivalently, to the total number of limit-state function evaluations) used in a run of an algorithm as the total computation effort of the algorithm.

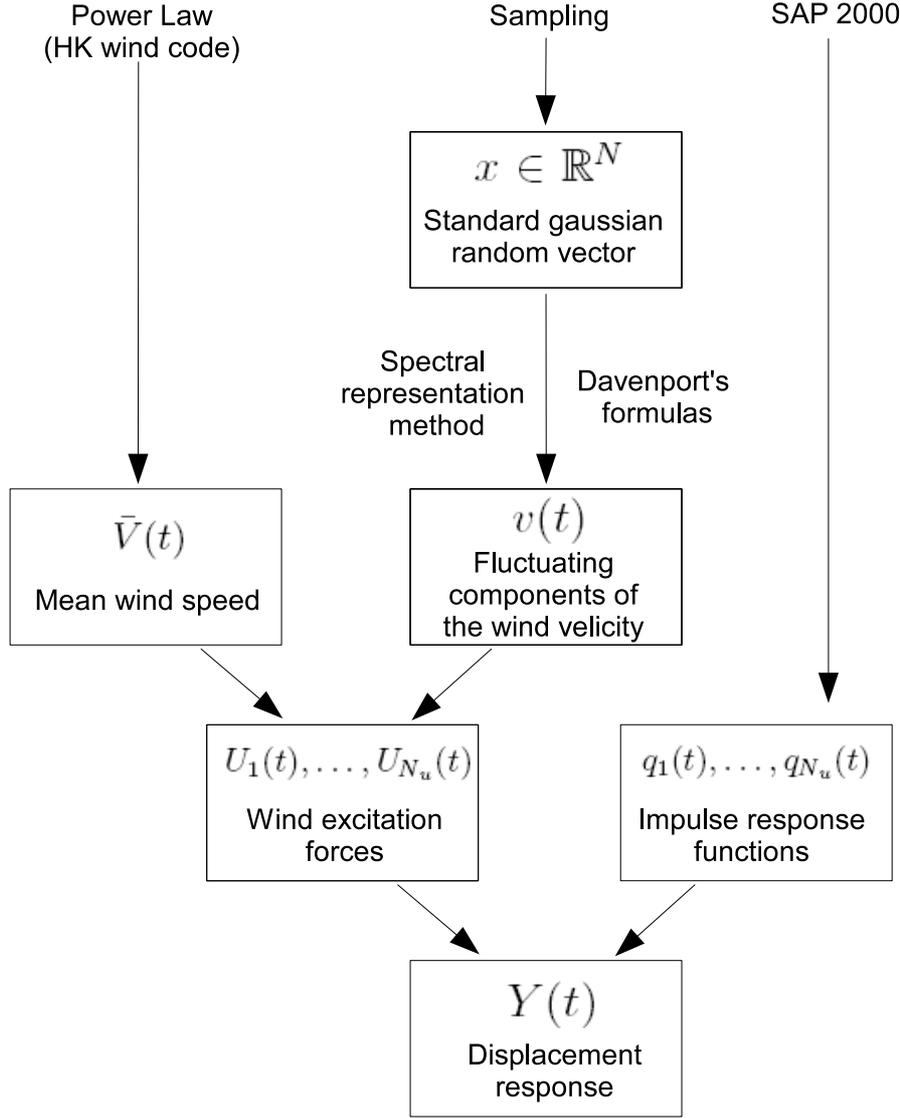


Figure 10: Simulation scheme.

#### 4.4 Simulation results

The failure events with thresholds  $z_1^* = 1.25$  m,  $z_2^* = 1.35$  m, and  $z_3^* = 1.45$  m are considered in the simulation. The corresponding Monte Carlo (MC) estimates of the failure probabilities are found to be  $p_{\Omega_1} = 3.3 \times 10^{-2}$  (with CV  $\delta_1 = 5.4\%$ ),  $p_{\Omega_2} = 6.8 \times 10^{-3}$  (with CV  $\delta_2 = 12.1\%$ ), and  $p_{\Omega_3} = 1.5 \times 10^{-3}$  (with CV  $\delta_3 = 25.8\%$ ), when using  $n_{MC} = 10^4$  samples.

The Horseracing Simulation algorithm (HRS) is applied with  $n = 500$  initial samples. The total computational efforts (CE) required by the algorithm are  $CE_1 = 1000$ ,  $CE_2 = 1500$ , and  $CE_3 = 2000$  for  $p_{\Omega_1}$ ,  $p_{\Omega_2}$ , and  $p_{\Omega_3}$  respectively. In order to get approximately the same CE, Subset Simulation is applied with  $n = 530$ ,  $n = 540$ , and  $n = 710$  initial samples respectively. The obtained mean values of the failure probability estimates and their CVs based on 25 runs of these algorithms are shown in the Fig. 11 along with the CVs of the Monte Carlo estimates (with the same CE) for comparison purposes. In the first case, when  $z_1^* = 1.25$  m, HRS and SS, showing approximately the same accuracy and efficiency, outperform MC. In the second case ( $z_2^* = 1.35$ ) as well as in the third case ( $z_3^* = 1.45$ ) the HRS outperforms SS. The reductions in

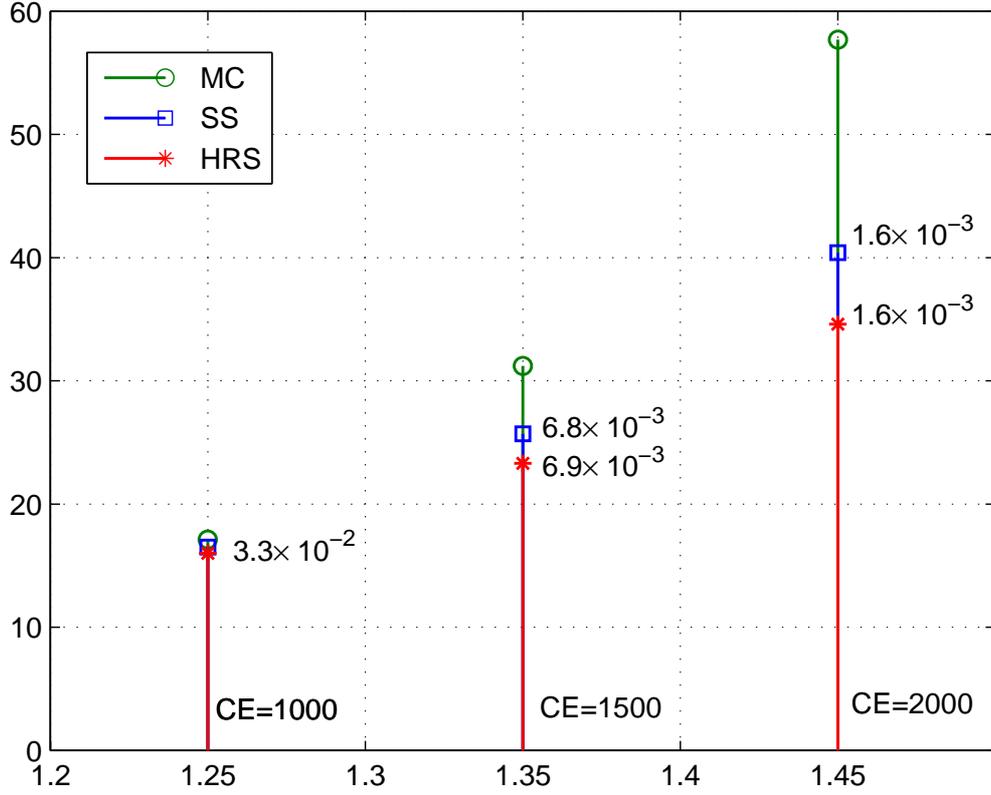


Figure 11: The CV of estimates obtained by Horseracing Simulation, Subset Simulation and Monte Carlo method.

CV achieved are  $(\delta_2^{SS} - \delta_2^{HRS})/\delta_2^{SS} = 9.3\%$  and  $(\delta_3^{SS} - \delta_3^{HRS})/\delta_3^{SS} = 14.4\%$  correspondingly.

Thus, the Horseracing Simulation algorithm clearly outperforms the standard Monte Carlo Simulation as well as Subset Simulation in the considered real-life example.

## 5 CONCLUSIONS

A new advanced stochastic simulation algorithm, called Horseracing Simulation, is proposed for solving high-dimensional reliability problems. The key idea behind HS is to approximate the cumulative distribution function (CDF) of the random variable of interest by empirical CDFs constructed from specially designed samples. The accuracy and efficiency of the new method is demonstrated with a real-life wind engineering example.

## 6 ACKNOWLEDGMENTS

This research has been supported by the Hong Kong Research Grants Council under grant 614008. This support is gratefully acknowledged.

## 7 APPENDIX

Let  $z_0$  be a random variable with PDF  $f_0$  and CDF  $F_0$ , and let  $\xi \sim f_0$  be its realization. Define  $z_1$  to be a new random variable with conditional distribution  $f_0(z|z \geq \xi)$ . In general, if  $\xi \sim f_k$  is a realization of the random variable  $z_k$ , define  $z_{k+1}$  to be a new random variable with

conditional distribution  $f_0(z|z \geq \xi)$ . This procedure defines a Markov chain  $z_0, z_1, \dots$ , which, as a matter of fact, is completely defined by the distribution of the random variable  $z_0$ .

**Theorem 1.** *The PDF  $f_k$  of the random variable  $z_k$  is*

$$f_k(z) = \frac{(-1)^k}{k!} f_0(z) [\log(1 - F_0(z))]^k \quad (33)$$

*Proof.* We prove this theorem by induction on  $k$ . For  $k = 0$  the statement of the theorem is obvious. Suppose that (33) holds for  $k$ . Then for  $k + 1$  we have:

$$\begin{aligned} f_{k+1}(z) &= \int_{-\infty}^{\infty} f_0(z|z \geq \xi) f_k(\xi) d\xi \\ &= \frac{(-1)^k}{k!} \int_{-\infty}^{\infty} \frac{f_0(z) I_F\{z \geq \xi\}}{1 - F_0(\xi)} f_0(\xi) [\log(1 - F_0(\xi))]^k d\xi \\ &= \frac{(-1)^k}{k!} f_0(z) \int_{-\infty}^z \frac{[\log(1 - F_0(\xi))]^k dF_0(\xi)}{1 - F_0(\xi)} \\ &= \frac{(-1)^{k+1}}{k!} f_0(z) \int_{-\infty}^z \frac{[\log(1 - F_0(\xi))]^k d(1 - F_0(\xi))}{1 - F_0(\xi)} \\ &= \frac{(-1)^{k+1}}{(k+1)!} f_0(z) [\log(1 - F_0(z))]^{k+1}. \end{aligned} \quad (34)$$

Which proves the theorem. □

## REFERENCES

- [1] S.K. Au, J.L. Beck, Estimation of small failure probabilities in high dimensions by subset simulation. *Probabilistic Engineering Mechanics*, **16(4)**, 263–277, 2001.
- [2] G.I. Schuëller, H.J. Pradlwarter, P.S. Koutsourelakis, A critical appraisal of reliability estimation procedures for high dimensions. *Probabilistic Engineering Mechanics*, **19(4)**, 463–474, 2004.
- [3] L.S. Katafygiotis, T. Moan, S.H. Cheung, Auxiliary domain method for solving multi-objective dynamic reliability problems for nonlinear structures. *Structural Engineering & Mechanics*, **25(3)**, 347–363, 2007.
- [4] L.S. Katafygiotis, K.M. Zuev, Estimation of small failure probabilities in high dimensions by Adaptive Linked Importance Sampling, *COMPdyn 2007*, Rethymno Crete Greece, 13–16 June 2007.
- [5] O. Ditlevsen, H.O. Madsen, *Structural Reliability Methods*, New York: Wiley, 1996.
- [6] N. Metropolis, S. Ulam, The Monte Carlo method, *J. Amer. statistical assoc.*, **44**, 247, 335–341, 1949.

- [7] D.B. Rubin, A noniterative sampling/importance resampling alternative to the data augmentation algorithm for creating a few imputations when the fraction of missing information is modest: the SIR algorithm, *Journal of the American Statistical Association*, **82**, 543-546, 1987.
- [8] L.S. Katafygiotis, J. Wang, Reliability Analysis of Wind-excited Structures using Domain Decomposition Method and Line Sampling, submitted to the *International Journal of Structural Engineering and Mechanics*.
- [9] W.H. Melbourne , Comparison of measurements on the CAARC standard tall building model in simulated model wind flows, *J. Wind. Eng. Ind. Aerodyn.*, **6**, 73-88, 1980.
- [10] E. Simiu, R.H. Scanlan, *Wind Effects on Structures*, John Wiley & Sons, Inc., New York, 1986.
- [11] M. Shinozuka, C.M. Jan, Digital simulation of random processes and its applications, *J. Sound Vib.*, **25(1)**, 111–128, 1972.
- [12] Shinozuka M., Deodatis G., (1991), *Simulation of stochastic processes by spectral representation*, *Appl. Mech. Rev.*, 44(4), 191-204.
- [13] G. Deodatis , Simulation of ergodic multivariate stochastic processes, *J. Eng. Mech.*, **122(8)**, 778–787, 1996.
- [14] M. Grigoriu, *Stochastic Calculus: Applications in Science and Engineering*, Birkhauser Boston, New York, 2002.
- [15] A.G. Davenport, The spectrum of horizontal gustiness near the ground in high winds, *J. R. Meteorol. Soc.*, **87**, 194–211, 1961.
- [16] A.G. Davenport, The dependence of wind load upon meteorological parameters, *Proceedings of the International Research Seminar on Wind Effects on Buildings and Structures*, Toronto, 1968.