

## Research Article

# Spline-Interpolation-Based FFT Approach to Fast Simulation of Multivariate Stochastic Processes

Jinhua Li,<sup>1,2</sup> Chunxiang Li,<sup>1</sup> and Shuisheng Chen<sup>2</sup>

<sup>1</sup> Department of Civil Engineering, Shanghai University, No. 149 Yanchang Road, Shanghai 200072, China

<sup>2</sup> Department of Civil Engineering, East China Jiaotong University, No. 808 Shuanggang Street, Nanchang 330013, China

Correspondence should be addressed to Chunxiang Li, li.chunxiang@sina.com

Received 2 June 2011; Revised 17 August 2011; Accepted 19 August 2011

Academic Editor: Wei-Chiang Hong

Copyright © 2011 Jinhua Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The spline-interpolation-based fast Fourier transform (FFT) algorithm, designated as the SFFT algorithm, is proposed in the present paper to further enhance the computational speed of simulating the multivariate stochastic processes. The proposed SFFT algorithm first introduces the spline interpolation technique to reduce the number of the Cholesky decomposition of a spectral density matrix and subsequently uses the FFT algorithm to further enhance the computational speed. In order to highlight the superiority of the SFFT algorithm, the simulations of the multivariate stationary longitudinal wind velocity fluctuations have been carried out, respectively, with resorting to the SFFT-based and FFT-based spectral representation SR methods, taking into consideration that the elements of cross-power spectral density matrix are the complex values. The numerical simulation results show that though introducing the spline interpolation approximation in decomposing the cross-power spectral density matrix, the SFFT algorithm can achieve the results without a loss of precision with reference to the FFT algorithm. In comparison with the FFT algorithm, the SFFT algorithm provides much higher computational efficiency. Likewise, the superiority of the SFFT algorithm is becoming more remarkable with the dividing number of frequency, the number of samples, and the time length of samples going up.

## 1. Introduction

Monte Carlo technique has widely been employed for simulating the stochastic processes which are either one-dimensional or multidimensional, univariate or multivariate, homogeneous or nonhomogeneous, stationary or nonstationary, and Gaussian or non-Gaussian. The Monte Carlo simulation methods are able to generate the sample functions that accurately provide the probabilistic characteristics of the corresponding stochastic processes. For the simulation of stochastic processes, the following approaches are now available: (1)

autoregressive (AR) method, such as Mignolet and Spanos [1, 2], Iannuzzi and Spinelli [3], Deodatis and Shinozuka [4], and Novak et al. [5]; (2) autoregressive moving average (ARMA) method, for example, Gersch and Yonemoto [6], Kozin [7], Kareem and Li [8], and Rossi et al. [9]; (3) spectral representation (SR) method, for instance, Shinozuka and Jan [10], Grigoriu [11], Deodatis [12], Grigoriu [13], and Chen and Letchford [14].

It is known that among foregoing simulation methods, the SR method has very high demands on both the computer memory and speed. Notwithstanding this, the SR does not have the problem of model selection, as does the AR and ARMA methods. Likewise, it is easy to implement and has high accuracy. Hence, the SR method has yet been receiving increasing attention in simulating the multivariate stochastic processes with the target cross power spectral density (CPSD) matrices. The primary concept of the SR method dates back to Rice [15], Goto and Toki [16], and Borgman [17]. For the multidimensional, multivariate, and nonstationary cases, Shinozuka [18, 19] first established the SR approach. Subsequently, Deodatis and Shinozuka [20] extended the application of the SR method to the simulation of stochastic waves. Li and Kareem [21] developed a hybrid discrete Fourier transform and digital filtering approach to simulate the multivariate random process. Likewise, Grigoriu [11] compared two different SR models. But, it is worth pointing out that the generated sample functions with resorting to the early algorithm of the SR method by Shinozuka and Jan [10] are not ergodic. In order to generate the ergodic sample functions, several attempts have been made via modifying the existing algorithms. Shinozuka et al. [22] introduced the idea of double-indexing frequencies. But, the obtained sample functions based on the proposed formula are still not ergodic. Deodatis [12] further extended the SR method to simulate the multivariate ergodic stochastic processes. Likewise, the capabilities and efficiency of the proposed algorithm were demonstrated in detail using a one-dimensional trivariate process as an example. On the other hand, the computational efficiency of the early algorithm of the SR method is yet low. In order to cope with this issue, the fast Fourier transform (FFT) algorithm was introduced into the SR method. Yang [23, 24] showed that the FFT algorithm can remarkably enhance the computational efficiency of the SR method and proposed a formula to simulate the random envelop processes. Shinozuka [25] extended the application of the FFT algorithm to the multidimensional cases. Wittig and Sinha [26] showed that the SR method combined with the FFT algorithm seems to be more than an order of magnitude faster than other simulation methods. Further, Li and Kareem [27] used the FFT-based approach to simulate the multivariate nonstationary Gaussian random processes with the prescribed evolutionary spectral description.

Likewise, the multicorrelated stationary random processes, such as the wind velocity or pressure fluctuations, on structures can be transformed into a set of subprocesses through diagonalizing their covariance or CPSD matrices with resorting to either the Cholesky (lower or upper triangular) or eigenvector decomposition. The eigenvector decomposition offers physically meaningful insight into the process as each eigenvector (eigenmode) may be characterized on the basis of its spatial distributions. Theoretically, the eigenvector decomposition is based on the Karhunen-Loeve expansion, consequently referred also to as the proper orthogonal decomposition (POD) [28]. The POD approach has been widely utilized to reduce the dimensions or variables in the large scale systems to enhance the computational speed, such as Holmes et al. [29], Di Paola and Gullo [30], Rathinam and Petzold [31], Chen and Kareem [32], and Solari and Carassale [33]. In [34], Carassale and Solari proposed some strategies such as interpolation of the factorizations of the CPSD matrix, cutoff of the spectral representation based on POD and parallelization of the code, which provide significant computational advantages enabling the simulation of large wind fields. Likewise, Ding et al.

[35] also introduced interpolation approximation of the factorizations of the CPSD matrix and cutoff of the spectral representation based on POD to the SR algorithm for the simulation of wind velocity fields on large scale structures. The numerical advantage of the POD technique, akin to the modal analysis of structural dynamics, relies mainly on the reduced-order representation via the truncation of the higher eigenmodes associated with smaller eigenvalues. Of course, this reduced-order representation must warrant that the important characteristics of the stochastic processes and related quantities remain unchanged, or the modification resulting from the approximate representation is acceptable. However, the truncation of higher modes may not necessarily work effectively in the case of local responses, implying that there exists a possibility of underestimating the local wind loads and corresponding effects [36]. In the case of using the CPSD matrix-based POD technique, the similar observations made by Chen and Kareem [37] once again underscore the foregoing phenomenon.

In order to illuminate solely the effectiveness of interpolation in fast simulation of multivariate stochastic processes without POD technique, the spline function employing the nature of both the smoothness and continuance is utilized to reduce the number of the Cholesky decomposition of power spectral matrix by means of interpolation. The spline-interpolation-based FFT algorithm, designated as the SFFT algorithm, then is proposed in the present paper. Therefore, different from [34, 35], the main purpose of the present study is to evaluate the computational efficiency and accuracy of the SFFT-based SR method with reference to the FFT-based SR method through simulating the multivariate stationary longitudinal wind velocity fluctuations with the phase angles.

## 2. Simulation of Multivariate Stochastic Processes

According to Deodatis [12] and Shinozuka [19], the multivariate stationary stochastic process  $f_j^0(t)$  ( $j = 1, 2, \dots, n$ ) with the mean value equal to zero can be simulated by the following series:

$$f_j(t) = 2\sqrt{\Delta\omega} \sum_{m=1}^n \sum_{l=1}^N |H_{jm}(\omega_{ml})| \cos[\omega_{ml}t - \vartheta_{jm}(\omega_{ml}) + \Phi_{ml}] \quad (j = 1, 2, \dots, n), \quad (2.1)$$

where  $\mathbf{H}(\omega)$  is the Cholesky decomposition of CPSD matrix  $\mathbf{S}^0(\omega)$  (see the appendix) and a lower triangular matrix;  $N$  is the sufficiently large dividing number of frequency;  $\Delta\omega = \omega_{\text{up}}/N$  is the circular frequency increment;  $\omega_{\text{up}}$  refers to the upper cutoff circular frequency, with the condition that, when  $\omega > \omega_{\text{up}}$ , the value of  $\mathbf{S}^0(\omega)$  is trivial and negligible;  $\Phi_{ml}$  is the sequences of independent random phase angles, distributed uniformly over the interval  $[0, 2\pi]$ ; the double-indexing frequency  $\omega_{ml} = l\Delta\omega - ((N-m)/N)\Delta\omega = (l-1)\Delta\omega + (m/N)\Delta\omega$ .

It is noted that the simulated stochastic process using the SR method is asymptotically Gaussian as  $N \rightarrow \infty$  (Deodatis [12]). Shinozuka and Deodatis [38] provided rigorous derivations and elaborations about asymptotic Gaussian of the simulated stochastic process according to the central limit theorem. It can be considered approximately Gaussian for most practical applications if  $N$  is greater than approximately 100 (Deodatis and Micaletti [39]).

In order to avoid aliasing, the time interval  $\Delta t$  has to obey the condition as follows:

$$\Delta t \leq \frac{2\pi}{2\omega_{\text{up}}}. \quad (2.2)$$

Likewise, the entire period of the sample functions in (2.1) can be calculated as follows:

$$T_0 = \frac{2\pi n}{\Delta\omega} = \frac{2\pi nN}{\omega_{\text{up}}}. \quad (2.3)$$

It has been proved by Deodatis [12] that the obtained results based on (2.1) possesses the ergodicity. Apparently, once the CPSD matrix is determined and the associated parameters, such as  $N$ ,  $\omega_{\text{up}}$ , and  $\Delta\omega$ , are properly chosen, the stationary one-dimensional multivariate Gaussian stochastic process can then be simulated quite well with resorting to (2.1).

### 3. Spline-Interpolation-Based FFT (SFFT) Algorithm

For the simulation of the multivariate stationary stochastic processes, the spline-interpolation-based FFT algorithm, referred to as the SFFT algorithm, is proposed in the present paper.

#### 3.1. Spline Interpolation Technique

Since the Cholesky decomposition has to be conducted separately for each frequency  $\omega_{ml}$ , the number of performing the Cholesky decomposition with respect to (2.1) then is  $n \times N$ . The computational effort of the SR method is tremendous in the large scale system, though the FFT algorithm can significantly enhance the computational speed.

For the multivariate stationary stochastic processes with the phase angles, the CPSD matrix  $\mathbf{S}^0(\omega)$  is the complex-valued matrix. Then, the element  $H_{jm}(\omega) = |H_{jm}(\omega)|e^{i\vartheta_{jm}}$  of the obtained lower triangular matrix  $\mathbf{H}(\omega)$  based on the Cholesky decomposition is the complex value. It is noted that  $|H_{jm}(\omega)|$  varies continuously with regard to the circular frequency. Therefore, as long as  $|H_{jm}(\omega)|$  at some appropriate circular frequency points are calculated,  $|H_{jm}(\omega)|$  at other circular frequency points can then be obtained by the cubic spline interpolation. Since the spline function has the nature of both the smoothness and continuance, it has widely been utilized to interpolate and fit data in engineering.

The circular frequency interval  $[0, \omega_u]$  is evenly divided into  $r$  subintervals by  $r - 1$  frequencies  $\omega_1, \omega_2, \dots, \omega_{r-1}$  ( $0 = \omega_0 < \omega_1 < \omega_2 < \dots < \omega_{r-1} < \omega_r = \omega_u$ ). The corresponding  $|H_{jm}(\omega_i)|$  ( $i = 0, 1, 2, \dots, r$ ) can then be calculated. It is assumed that  $\widetilde{H}_{jm}(\omega_i) = |H_{jm}(\omega_i)|$  ( $i = 0, 1, 2, \dots, r$ ) and second continuous derivative  $\widetilde{H}_{jm}''(\omega_i) = P_i$  ( $i = 0, 1, 2, \dots, r$ ), then  $\widetilde{H}_{jm}(\omega)$  at other frequency points can be obtained by the cubic spline interpolation. The flow scheme of building spline interpolation  $\widetilde{H}_{jm}(\omega)$  is shown in Figure 1.

In the circular frequency interval  $[\omega_{i-1}, \omega_i]$ ,  $\widetilde{H}_{jm}(\omega)$  is expressed as cubic polynomial, and then  $\widetilde{H}_{jm}''(\omega)$  is a linear function:

$$\widetilde{H}_{jm}''(\omega) = P_{i-1} \frac{\omega_i - \omega}{\omega_i - \omega_{i-1}} + P_i \frac{\omega - \omega_{i-1}}{\omega_i - \omega_{i-1}} = \frac{r}{\omega_u} (P_{i-1}(\omega_i - \omega) + P_i(\omega - \omega_{i-1})). \quad (3.1)$$

Then, the continuous double integral of  $\widetilde{H}_{jm}''(\omega)$  can be expressed as

$$\widetilde{H}_{jm}(\omega) = \frac{r}{6\omega_u} \left[ (\omega_i - \omega)^3 P_{i-1} + (\omega - \omega_{i-1})^3 P_i \right] + A_i(\omega_i - \omega) + B_i(\omega - \omega_{i-1}), \quad (3.2)$$

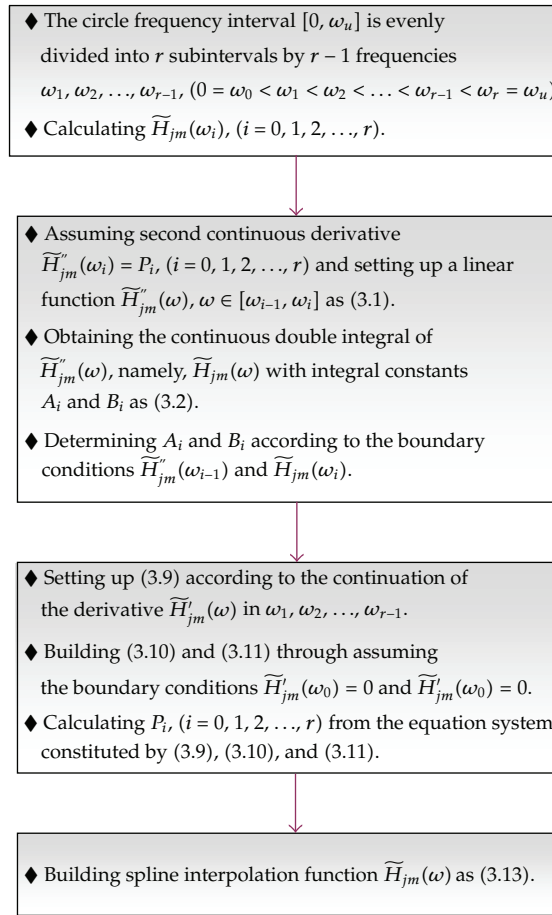


Figure 1: Flow scheme of building spline interpolation function  $\tilde{H}_{jm}(\omega)$ .

in which  $A_i$  and  $B_i$  are integral constants. According to the boundary conditions

$$\begin{aligned} \tilde{H}_{jm}(\omega_{i-1}) &= \frac{\omega_u^2}{6r^2} P_{i-1} + A_i \frac{\omega_u}{r}, \\ \tilde{H}_{jm}(\omega_i) &= \frac{\omega_u^2}{6r^2} P_i + B_i \frac{\omega_u}{r}, \end{aligned} \tag{3.3}$$

the integral constants  $A_i$  and  $B_i$  can be expressed as

$$\begin{aligned} A_i &= \left( \tilde{H}_{jm}(\omega_{i-1}) - \frac{\omega_u^2}{6r^2} P_{i-1} \right) \frac{r}{\omega_u}, \\ B_i &= \left( \tilde{H}_{jm}(\omega_i) - \frac{\omega_u^2}{6r^2} P_i \right) \frac{r}{\omega_u}. \end{aligned} \tag{3.4}$$

Substituting (3.4) into (3.2),

$$\begin{aligned}\widetilde{H}_{jm}(\omega) &= \frac{r}{6\omega_u} \left[ (\omega_i - \omega)^3 P_{i-1} + (\omega - \omega_{i-1})^3 P_i \right] \\ &+ \left( \widetilde{H}_{jm}(\omega_{i-1}) - \frac{\omega_u^2}{6r^2} P_{i-1} \right) \frac{r}{\omega_u} (\omega_i - \omega) \\ &+ \left( \widetilde{H}_{jm}(\omega_i) - \frac{\omega_u^2}{6r^2} P_i \right) \frac{r}{\omega_u} (\omega - \omega_{i-1}) \quad (\omega_{i-1} \leq \omega \leq \omega_i; i = 1, 2, 3, \dots, r).\end{aligned}\quad (3.5)$$

Then, the derivative of  $\widetilde{H}_{jm}(\omega)$  can be written as

$$\begin{aligned}\widetilde{H}'_{jm}(\omega) &= \frac{r}{2\omega_u} (\omega - \omega_{i-1})^2 P_i - \frac{r}{2\omega_u} (\omega_i - \omega)^2 P_{i-1} + \frac{\omega_u}{6r} (P_{i-1} - P_i) \\ &+ \frac{r}{\omega_u} \left( \widetilde{H}_{jm}(\omega_i) - \widetilde{H}_{jm}(\omega_{i-1}) \right) \quad (\omega_{i-1} \leq \omega \leq \omega_i; i = 1, 2, 3, \dots, r).\end{aligned}\quad (3.6)$$

In the interval  $[\omega_{i-1}, \omega_i]$ ,

$$\widetilde{H}'_{jm}(\omega_i) = \frac{\omega_u}{3r} P_i + \frac{\omega_u}{6r} P_{i-1} + \frac{r}{\omega_u} \left( \widetilde{H}_{jm}(\omega_i) - \widetilde{H}_{jm}(\omega_{i-1}) \right), \quad (3.7)$$

and, in next interval  $[\omega_i, \omega_{i+1}]$ ,

$$\widetilde{H}'_{jm}(\omega_i) = -\frac{\omega_u}{3r} P_i - \frac{\omega_u}{6r} P_{i+1} + \frac{r}{\omega_u} \left( \widetilde{H}_{jm}(\omega_{i+1}) - \widetilde{H}_{jm}(\omega_i) \right). \quad (3.8)$$

According to (3.7) and (3.8), we can obtain

$$P_{i+1} + 4P_i + P_{i-1} = 6 \frac{r^2}{\omega_u^2} \left( \widetilde{H}_{jm}(\omega_{i+1}) - 2\widetilde{H}_{jm}(\omega_i) + \widetilde{H}_{jm}(\omega_{i-1}) \right) \quad (i = 1, 2, 3, \dots, r-1). \quad (3.9)$$

In order to determine  $P_i$  ( $i = 0, 1, 2, \dots, r$ ), the boundary conditions are assumed as  $\widetilde{H}'_{jm}(\omega_0) = 0$  and  $\widetilde{H}'_{jm}(\omega_r) = 0$ . Then, we can obtain the following functions:

$$2P_0 + P_1 = 6 \frac{r^2}{\omega_u^2} \left( \widetilde{H}_{jm}(\omega_1) - \widetilde{H}_{jm}(\omega_0) \right), \quad (3.10)$$

$$2P_r + P_{r-1} = 6 \frac{r^2}{\omega_u^2} \left( \widetilde{H}_{jm}(\omega_{r-1}) - \widetilde{H}_{jm}(\omega_r) \right). \quad (3.11)$$

According to (3.9), (3.10) and (3.11),  $P_i$  ( $i = 0, 1, 2, \dots, r$ ) can be determined through the following equation system

$$\begin{aligned} 2P_0 + P_1 &= 6 \frac{r^2}{\omega_u^2} \left( \widetilde{H}_{jm}(\omega_1) - \widetilde{H}_{jm}(\omega_0) \right), \\ P_2 + 4P_1 + P_0 &= 6 \frac{r^2}{\omega_u^2} \left( \widetilde{H}_{jm}(\omega_2) - 2\widetilde{H}_{jm}(\omega_1) + \widetilde{H}_{jm}(\omega_0) \right), \\ P_3 + 4P_2 + P_1 &= 6 \frac{r^2}{\omega_u^2} \left( \widetilde{H}_{jm}(\omega_3) - 2\widetilde{H}_{jm}(\omega_2) + \widetilde{H}_{jm}(\omega_1) \right), \\ &\vdots \\ P_r + 4P_{r-1} + P_{r-2} &= 6 \frac{r^2}{\omega_u^2} \left( \widetilde{H}_{jm}(\omega_r) - 2\widetilde{H}_{jm}(\omega_{r-1}) + \widetilde{H}_{jm}(\omega_{r-2}) \right), \\ 2P_r + P_{r-1} &= 6 \frac{r^2}{\omega_u^2} \left( \widetilde{H}_{jm}(\omega_{r-1}) - \widetilde{H}_{jm}(\omega_r) \right). \end{aligned} \quad (3.12)$$

Then, substituting  $P_i$  ( $i = 0, 1, 2, \dots, r$ ) into (3.5), we can obtain

$$\widetilde{H}_{jm}(\omega) = \begin{cases} \frac{r}{6\omega_u} \left[ (\omega_1 - \omega)^3 P_0 + (\omega - \omega_0)^3 P_1 \right] + \left( \widetilde{H}_{jm}(\omega_0) - \frac{\omega_u^2}{6r^2} P_0 \right) \frac{r}{\omega_u} (\omega_1 - \omega) \\ \quad + \left( \widetilde{H}_{jm}(\omega_1) - \frac{\omega_u^2}{6r^2} P_1 \right) \frac{r}{\omega_u} (\omega - \omega_0) \quad (\omega_0 \leq \omega \leq \omega_1), \\ \frac{r}{6\omega_u} \left[ (\omega_2 - \omega)^3 P_1 + (\omega - \omega_1)^3 P_2 \right] + \left( \widetilde{H}_{jm}(\omega_1) - \frac{\omega_u^2}{6r^2} P_1 \right) \frac{r}{\omega_u} (\omega_2 - \omega) \\ \quad + \left( \widetilde{H}_{jm}(\omega_2) - \frac{\omega_u^2}{6r^2} P_2 \right) \frac{r}{\omega_u} (\omega - \omega_1) \quad (\omega_1 \leq \omega \leq \omega_2), \\ \vdots \\ \frac{r}{6\omega_u} \left[ (\omega_i - \omega)^3 P_{i-1} + (\omega - \omega_{i-1})^3 P_i \right] + \left( \widetilde{H}_{jm}(\omega_{i-1}) - \frac{\omega_u^2}{6r^2} P_{i-1} \right) \frac{r}{\omega_u} (\omega_i - \omega) \\ \quad + \left( \widetilde{H}_{jm}(\omega_i) - \frac{\omega_u^2}{6r^2} P_i \right) \frac{r}{\omega_u} (\omega - \omega_{i-1}) \quad (\omega_{r-1} \leq \omega \leq \omega_r). \end{cases} \quad (3.13)$$

Introducing the spline interpolation into  $|H_{jm}(\omega)|$ , (2.1) for the simulation of the multivariate stationary stochastic processes can then be rewritten as follows:

$$f_j(t) = 2\sqrt{\Delta\omega} \sum_{m=1}^n \sum_{l=1}^N \widetilde{H}_{jm}(\omega_{ml}) \cos[\omega_{ml}t - \vartheta_{jm}(\omega_{ml}) + \Phi_{ml}] \quad (j = 1, 2, \dots, n). \quad (3.14)$$

### 3.2. Fast Fourier Transform (FFT) Algorithm

Taking into account that introducing the FFT algorithm renders very high computation efficiency in simulating the stationary processes (Li and Kareem [27]), with resorting to the FFT algorithm, (3.14) can then be derived as follows:

$$f_j(p\Delta t) = \operatorname{Re} \left\{ \sum_{m=1}^j \widetilde{h}_{jm}(p\Delta t) \exp \left[ i \left( \frac{m\Delta\omega}{n} \right) (p\Delta t) \right] \right\}, \quad (3.15)$$

where

$$p = 0, 1, \dots, n \times M - 1;$$

$$M = \frac{T_0}{n\Delta t};$$

$$T_0 = \frac{2\pi n}{\Delta\omega} = \frac{2\pi nN}{\omega_{\text{up}}};$$

$$\widetilde{h}_{jm}(p\Delta t) = \begin{cases} \widetilde{g}_{jm}(p\Delta t), & p = 0, 1, \dots, M-1, \\ \widetilde{g}_{jm}[(p-M)\Delta t], & p = M, M+1, \dots, 2M-1, \\ \vdots & \vdots \\ \widetilde{g}_{jm}[(p-(n-1)M)\Delta t], & p = (n-1)M, (n-1)M+1, \dots, nM-1, \end{cases}$$

$$\widetilde{g}_{jm}(p\Delta t) = \sum_{l=1}^{M-1} \widetilde{B}_{jml} \exp \left[ ilp \frac{2\pi}{M} \right] \quad (p = 0, 1, \dots, M-1),$$

$$\widetilde{B}_{jml} = \begin{cases} \sqrt{2\Delta\omega} \widetilde{H}_{jm} \left( l\Delta\omega + \frac{m\Delta\omega}{n} \right) \exp \left[ -i\vartheta_{jm} \left( l\Delta\omega + \frac{m\Delta\omega}{n} \right) \right] \exp(i\Phi_{ml}), & 0 \leq l \leq N, \\ 0, & N < l \leq M-1. \end{cases} \quad (3.16)$$

It is noteworthy that  $\widetilde{g}_{jm}(p\Delta t)$  can be obtained in terms of the inverse FFT of  $\widetilde{B}_{jml}$ .



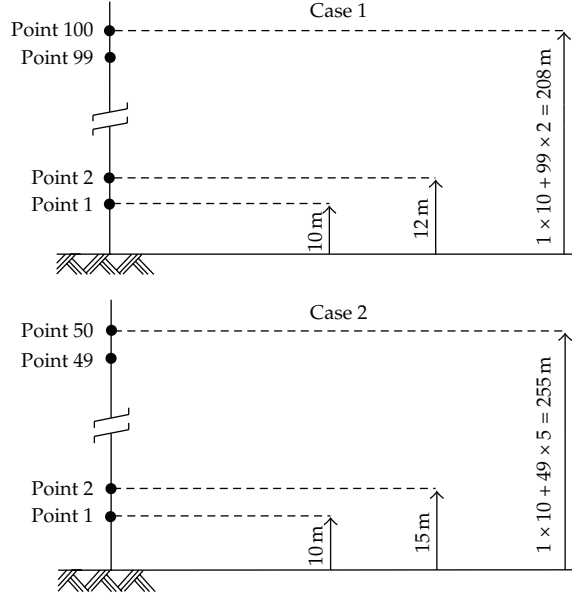


Figure 2: Configuration of points along a vertical line in Case 1 and Case 2.

## 4. Numerical Results

In the following, two cases of performing the simulation of multivariate stationary fluctuating wind velocity are taken into consideration in order to demonstrate the capabilities and computational efficiency of the proposed SFFT algorithm.

### 4.1. Description of Case 1

A hundred velocity points to be simulated are evenly distributed from 10 m to 208 m heights along a vertical line (see Case 1 in Figure 2). The process corresponding to these hundred components is denoted by  $f_j^0(t)$  [ $f_1^0(t), f_2^0(t), \dots, f_{100}^0(t)$ ]. It is assumed that the mean value of the process is equal to zero. Then, the elements of its CPSD matrix may be given as

$$S_{jj}^0(\omega) = S_j(\omega) \quad (j = 1, 2, \dots, 100), \quad (4.1)$$

$$S_{jk}^0(\omega) = \Gamma_{jk}(\omega) \sqrt{S_j(\omega) S_k(\omega)} e^{-i\theta_{jk}(\omega)} \quad (j, k = 1, 2, \dots, 100, j \neq k), \quad (4.2)$$

in which  $S_j(\omega)$  represents the power spectral density function of  $f_j^0(t)$ ;  $\Gamma_{jk}(\omega)$  denotes the coherence function between  $f_j^0(t)$  and  $f_k^0(t)$ ;  $\theta_{jk}(\omega)$  refers to the phase angle between  $S_j(\omega)$  and  $S_k(\omega)$ .

It is emphasized here that the stochastic process described by (4.1) and (4.2) is nonhomogeneous in space, since the longitudinal velocity fluctuations  $f_1^0(t), f_2^0(t), \dots, f_{100}^0(t)$  have different frequency contents, namely,  $S_1(\omega) \neq S_2(\omega) \neq \dots \neq S_{100}(\omega)$ .

The following expression proposed by Kaimal et al. [40] is selected to model the two-sided power spectral density (PSD) function of the longitudinal wind velocity fluctuations at different height:

$$S(z, \omega) = \frac{1}{2} \frac{200}{2\pi} u_*^2 \frac{z}{U(z)} \frac{1}{[1 + 50(\omega z / 2\pi U(z))]^{5/3}}, \quad (4.3)$$

where  $z$  is the elevation above the ground, in meters;  $\omega$  is the circular frequency, in rad/s;  $u_*$  is the shear velocity of the flow, in m/s;  $U(z)$  is the mean wind speed at height  $z$ , in m/s.

Taking into account the following coherence function between the velocity fluctuations at two different heights  $z_j$  and  $z_k$  suggested by Davenport [41],

$$\Gamma(\Delta z, \omega) = \exp \left[ -\frac{\omega}{2\pi} \frac{C_z \Delta z}{(1/2)[U(z_j) + U(z_k)]} \right] \quad (4.4)$$

in which  $U(z_j)$  and  $U(z_k)$  are the mean wind speeds at heights  $z_j$  and  $z_k$ , respectively;  $\Delta z = |z_j - z_k|$ ;  $C_z$  is a constant that may be set equal to 10 for structural design purposes (Kristensen and Jensen [42] and Simiu and Scanlan [43]).

Furthermore, the expression of the phase angle  $\theta_{jk}(\omega)$  is taken from Di Paola [44] and Simiu and Scanlan [43],

$$\theta_{jk}(\omega) = \frac{\omega(z_j - z_k)}{v_{\text{app}}^{(j,k)}}. \quad (4.5)$$

In (4.2),  $e^{-i\theta_{jk}(\omega)}$  is a measure of the wave passage delay due to the apparent velocity of waves  $v_{\text{app}}^{(j,k)}$ .

Likewise, the apparent velocity of waves can be assumed to be the following form suggested by Simiu and Scanlan [43]:

$$v_{\text{app}}^{(j,k)} = \frac{\pi [U(z_j) + U(z_k)]}{C_\theta}, \quad (4.6)$$

in which  $C_\theta$  is an appropriate coefficient that has to be determined from experimental data.

In the present paper, the expression by Peil and Telljohann [45] is taken into consideration, which has the form  $v_{\text{app}}^{(j,k)} = U[(z_j + z_k)/2]/5.5$ . The presence of the phase angle turns into a time shift of the peak of the cross-correlation functions from the data of the measured fluctuations at different heights.

Let it be supposed that the mean wind velocity at the first point 1 ( $z = 10$  m) is  $U(10) = 26$  m/s and that the surface roughness length is  $z_0 = 0.001266$  m (corresponding shear velocity of the flow  $u_* = 1.76$  m/s). It is worth mentioning that the values  $z_0 = 0.001266$  m and  $u_* = 1.76$  m/s are taken from Simiu and Scanlan [43]. The upper cutoff frequency is  $\omega_{\text{up}} = 2\pi$  rad/s. The dividing number of frequency is  $N = 8192$ . The simulation is carried out at  $1 \times M = 1 \times 8192 \times 2 = 16384$  time instants with a time step  $\Delta t = 0.5$  s, over a time length equal

**Table 1:** Comparisons among several different schemes for wind velocity simulation.

Cases	Algorithms	Dividing number of frequency ( $N$ )	Number of points ( $n$ )	$T$ (sec)	Elapsed time (min)
Case 1	SFFT	8192	100	8192	10.2
	FFT	8192	100	8192	215.9
Case 2	SFFT	2048	50	2048	0.5
	FFT	2048	50	2048	5.9

to  $T = 8192$  s. The mean wind speeds at other points are computed in terms of the logarithmic law (i.e., the vertical profile), which has the form as follows:

$$U(z_j) = \frac{\ln(z_j/z_0)}{\ln(z_s/z_0)} U(z_s). \quad (4.7)$$

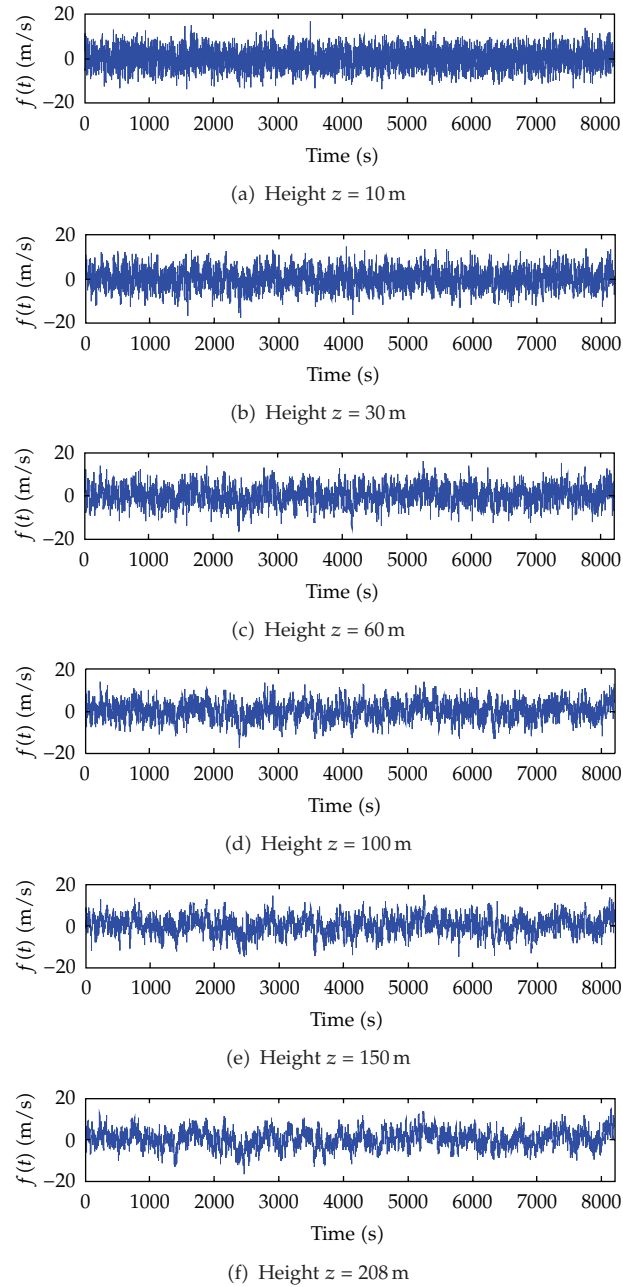
#### 4.2. Description of Case 2

Fifty velocity points to be simulated are evenly distributed from 10 m to 255 m heights along a vertical line (see Case 2 in Figure 2). The process corresponding to these fifty components is represented by  $f_j^0(t)$  [ $f_1^0(t), f_2^0(t), \dots, f_{50}^0(t)$ ]. The dividing number of frequency is  $N = 2048$ . The numerical simulation is implemented at  $1 \times M = 1 \times 2048 \times 2 = 4096$  time instants with a time step  $\Delta t = 0.5$  s, over a time length equal to  $T = 2048$  s. The other conditions are the same as Case 1.

#### 4.3. Numerical Analyses

The comparisons among several different schemes are made and listed in Table 1. The Cholesky decomposition with the SFFT algorithm is implemented on 128 circular frequency points. It can be seen that in Case 1, the elapsed time for the simulation with the FFT algorithm is 215.9 min, while that for the simulation with the SFFT algorithm is only 10.2 min; in the Case 2, the elapsed time for the simulation with the FFT algorithm is 5.9 min, while that for the simulation with the SFFT algorithm is only 0.5 min. Apparently, in the calculation speed the SFFT algorithm has considerable advantage over the FFT algorithm. Likewise, the superiority becomes more remarkable with the increasing of  $N$ ,  $n$ , and  $T$ . This is because the number of the Cholesky decomposition with the FFT algorithm equals  $n \times N$ , while that with the SFFT algorithm keeps constant with the value equal to 128. For a determined SFFT or FFT algorithm, however, the computational efficiency becomes lower with the increasing of the parameters  $N$ ,  $n$ , and  $T$ .

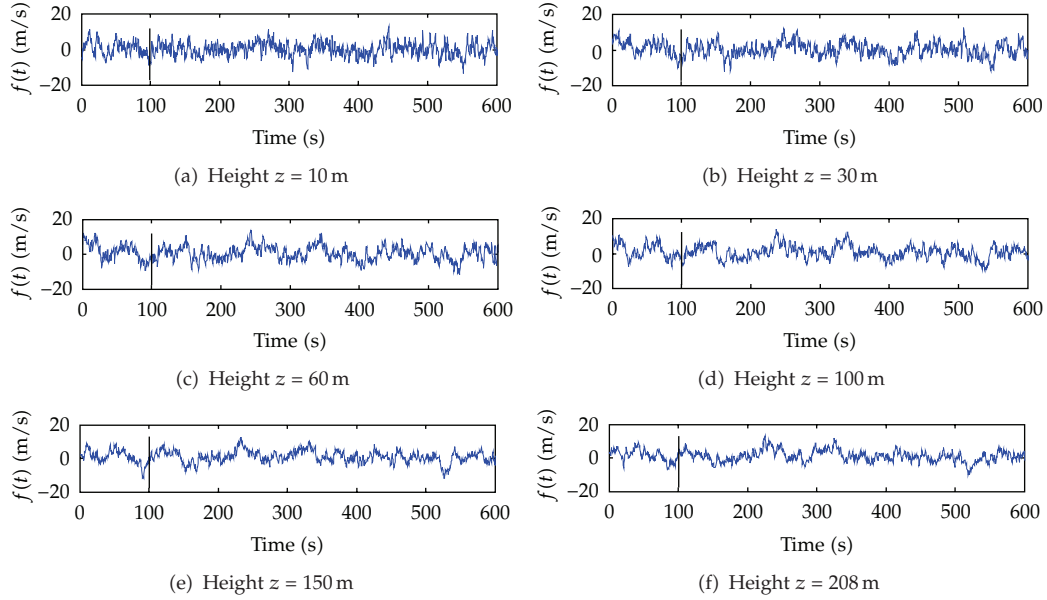
Since in Case 1 the number of the stimulated time histories is very large, without any loss of generality, arbitrary six time histories of all are chosen to demonstrate the effort of simulating the longitudinal wind velocity fluctuations by resorting to the proposed SFFT algorithm. The generated samples of longitudinal wind velocity fluctuations at six points 1, 11, 26, 46, 71, and 100 (see Case 1 in Figure 2), denoted by  $f_1(t), f_{11}(t), f_{26}(t), f_{46}(t), f_{71}(t)$ , and  $f_{100}(t)$ , respectively, are displayed in Figures 3(a), 3(b), 3(c), 3(d), 3(e), and 3(f) for a time length  $T = 8192$  s. Likewise, the parts in the first 600 s of these six time histories are further



**Figure 3:** Generated sample functions for longitudinal wind velocity fluctuations at six different heights in Case 1.

extracted out, respectively, as shown Figure 4, in order to better visualize the differences and the similarities among these six time histories.

As far as the correlation degree among these six time histories is concerned, it is clearly seen from Figure 4 that with the decreasing of the distance between arbitrary two points, the loss of coherence between two samples corresponding, respectively, to these two points will

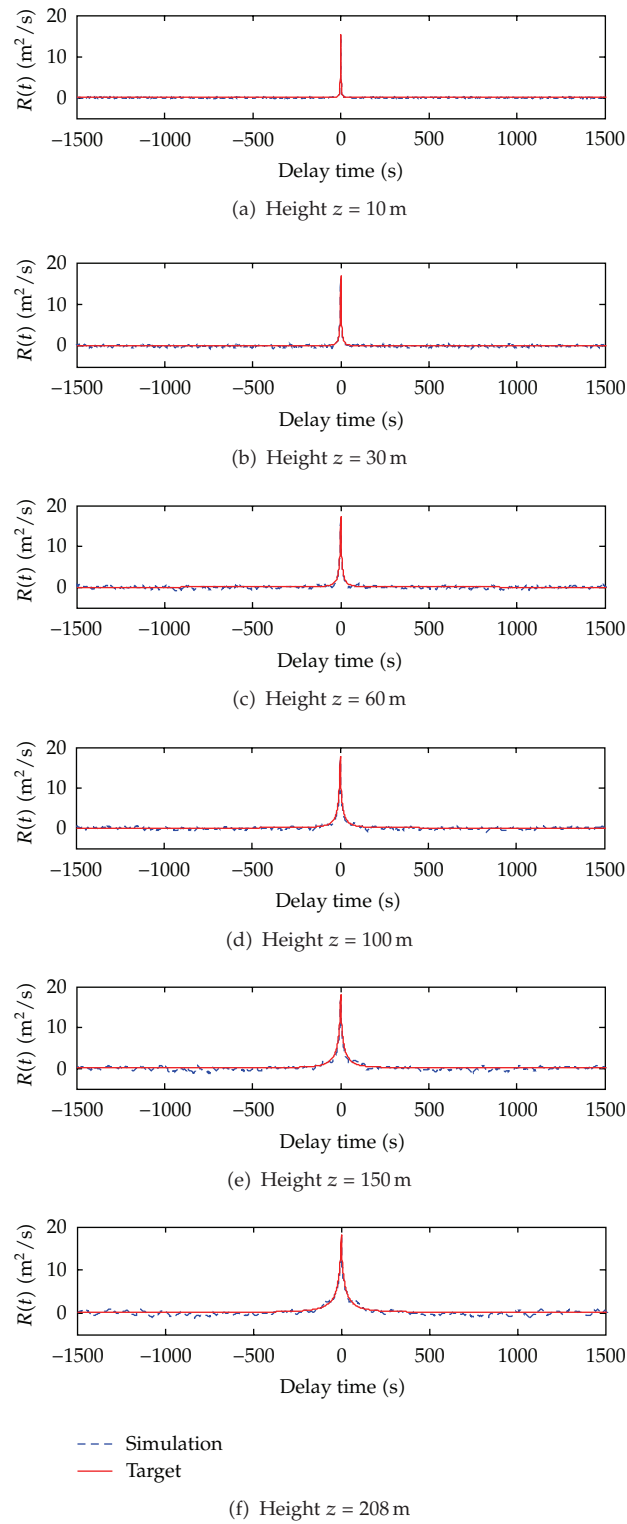


**Figure 4:** Generated sample functions for longitudinal wind velocity fluctuations at six different heights, over first 600 s of generated sample functions displayed in Figure 3.

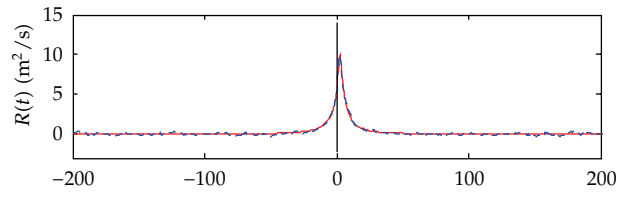
become smaller. For example, since the distance between the points 1 and 11 is the smallest in these six points, consequently, there is the smallest loss of coherence between  $f_1(t)$  and  $f_{11}(t)$ . On the other hand, there is the biggest loss of coherence between  $f_1(t)$  and  $f_{100}(t)$ , as the point 100 is located at 198 m height from the point 1 (see Case 1 in Figure 2). This behavior is controlled by the coherence functions (see (4.4)). Furthermore, the phase angle can also be detected clearly at 100 s time instant between arbitrary two samples of these six time histories. Likewise, the phase angle of these six time histories will become larger with the increasing of the distance between arbitrary two time histories. Similarly, this phenomenon can be controlled by the phase angle functions (see (4.5)).

The forgoing elucidations indicate that the proposed SFFT algorithm is able to simulate the longitudinal wind velocity fluctuations that are spatially correlated according to a prescribed coherence function and that possess the phase angles following a prescribed phase angle function.

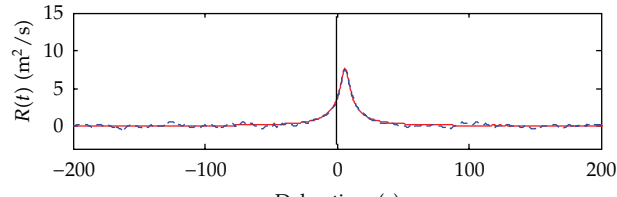
Plotted in Figure 5 is the temporal autocorrelation function  $[R_{jj}(\tau) (j = 1, 11, 26, 46, 71, 100)]$  of the generated sample function shown in Figure 3, along with the target autocorrelation function  $[R_{jj}^0(\tau) (j = 1, 11, 26, 46, 71, 100)]$ . It is worth pointing out that the target autocorrelation function  $[R_{jj}^0(\tau)]$  is computed with resorting to the Wiener-Khintchine transformation. As can be seen in Figure 5, the simulated autocorrelation function  $[R_{jj}(\tau)]$  practically coincides with the target autocorrelation function  $[R_{jj}^0(\tau)]$ . Displayed in Figure 6 is the temporal cross-correlation function  $[R_{jk}(\tau) (j = 1, k = 11, 26, 46, 71, 100)]$  of the generated sample function presented in Figure 3, along with the target cross-correlation function  $[R_{jk}^0(\tau) (j = 1, k = 11, 26, 46, 71, 100)]$ . The same, the target cross-correlation function  $[R_{jk}^0(\tau)]$  is calculated according to the Wiener-Khintchine transformation. As expected, the simulated cross-correlation function  $[R_{jk}(\tau) (j \neq k)]$  practically coincides



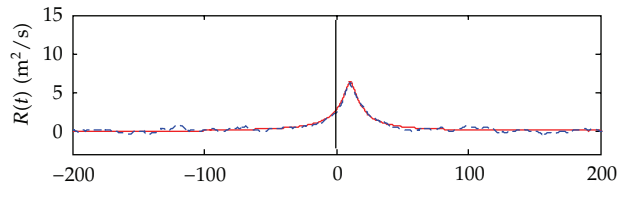
**Figure 5:** Temporal autocorrelation functions of generated sample function displayed in Figure 3 versus corresponding targets.



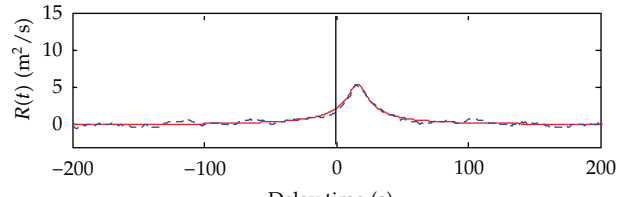
(a)  $z = 10$  m and  $z = 30$  m



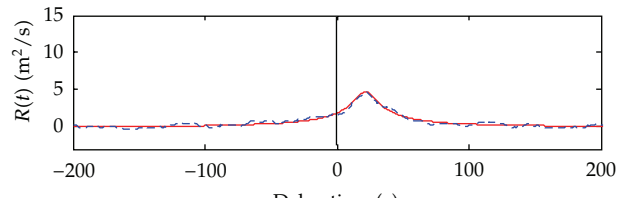
(b)  $z = 10$  m and  $z = 60$  m



(c)  $z = 10$  m and  $z = 100$  m



(d)  $z = 10$  m and  $z = 150$  m



-- Target  
 — Simulation

(e)  $z = 10$  m and  $z = 208$  m

**Figure 6:** Temporal cross-correlation functions of generated sample functions displayed in Figure 3 versus corresponding targets.

with the target cross-correlation function  $[R_{jk}^0(\tau) (j \neq k)]$ . Apparently, the temporal auto correlation and cross-correlation functions of any sample function  $[f_j(t) (j = 1, 2, \dots, 100)]$  are, respectively, identical to the target autocorrelation and cross-correlation functions based on the SFFT algorithm. Moreover, Figure 6 demonstrates that the presence of the phase angle turns into a time shift of the peak of the cross-correlation functions. Likewise, the time shift will become larger with the increasing of the distance between arbitrary two points. It is proved once again that the proposed SFFT algorithm can efficiently carry out the simulation of longitudinal wind velocity fluctuations with the phase angles.

From Figure 7, it is clear that the estimated spectra derived from these sample time histories and the target spectra are in remarkably good agreement with each other. Likewise, it is seen from Figures 8 and 9 that the mean and variance values of these 100 time histories generated with the SFFT algorithm approach those with the FFT algorithm, which implies that the SFFT and FFT algorithms practically achieve the same computation accuracy. More specifically, Figure 8 shows that the variance values of the generated time histories with resorting to both the SFFT and FFT are identical to the target values, respectively. It is observed in Figure 9 that the mean values of the generated time histories with resorting to both the SFFT and FFT are not more than 0.2 m/s, effectively meaning that these mean values are considerably small with respect to the zero target value.

In order to demonstrate Gaussian of the simulated wind velocity fluctuations, Figure 10 shows the probability density functions of generated samples displayed in Figure 3 versus corresponding target (Gaussian). It is clear that the probability density functions estimated from these samples match closely Gaussian probability density function.

#### 4.4. Accuracy and Time Expense of Various Interpolations

Four kinds of interpolation techniques, such as SFFT, the cubic Lagrangian interpolation based FFT (CL-FFT), square Lagrangian interpolation based FFT (SL-FFT), and neural network interpolation based FFT (NN-FFT), are taken into consideration to simulate the wind velocity field for the evaluation of accuracy and time expense of various interpolations [46]. The root mean square error (RMSE) and error factor (EF) [47] are introduced to evaluate the accuracy of these four interpolation-based FFT techniques:

$$\begin{aligned} \text{RMSE} &= \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - Y_i)^2}, \\ \text{EF} &= 1 - \left( \frac{\sum_{i=1}^N |y_i - Y_i|}{\sum_{i=1}^N |Y_i|} \right), \end{aligned} \quad (4.8)$$

in which,  $N$  is the total number of time points in the generated sample;  $Y_i$  and  $y_i$  signify generated wind velocities at a certain instant employing the FFT and the interpolation-based FFT, respectively. It is noted that, the interpolation based FFT is more precise if the value of RMSE is closer to zero; the interpolation-based FFT is more precise if the value of EF is closer to one.

The data coming from [46] are listed in Tables 2 and 3. Table 2 shows the elapsed time in the simulation using the SL-FFT, CL-FFT, SFFT, NN-FFT, and FFT, respectively, and Table 3 records both the RMSE and EF values. It is from Table 2 seen that the efficiency of the SFFT



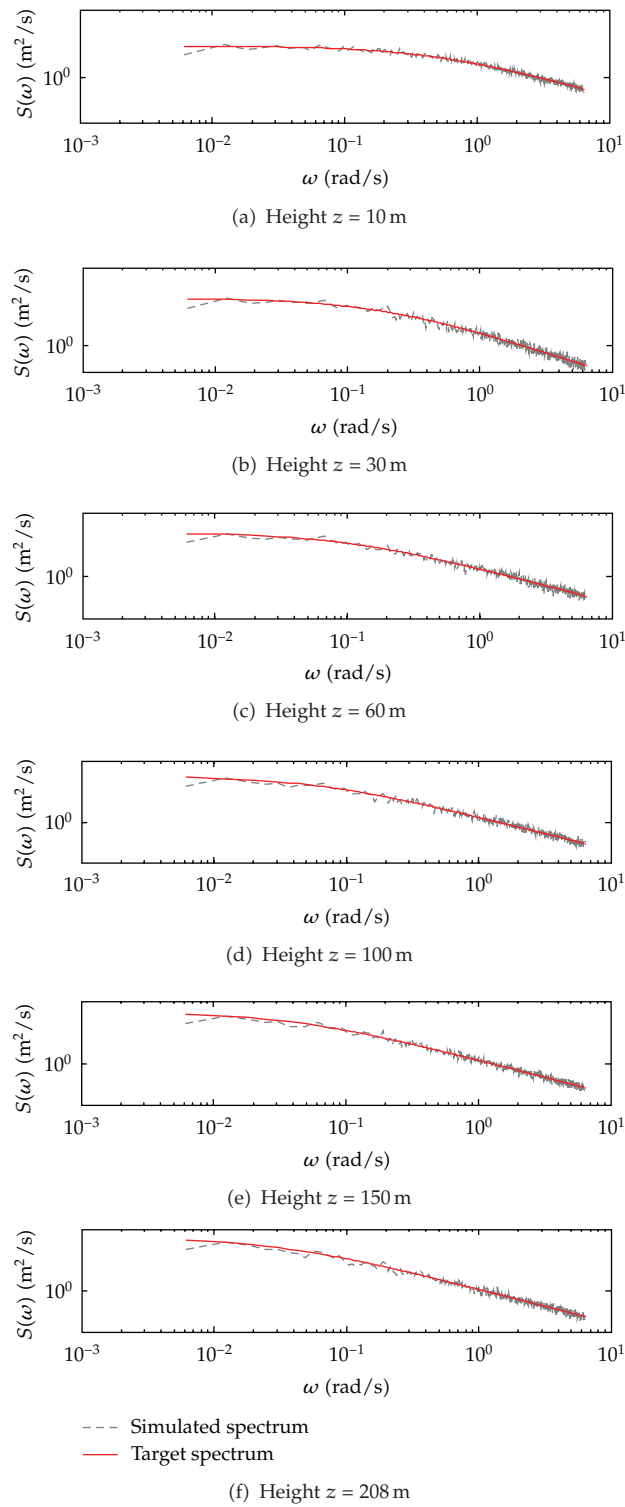
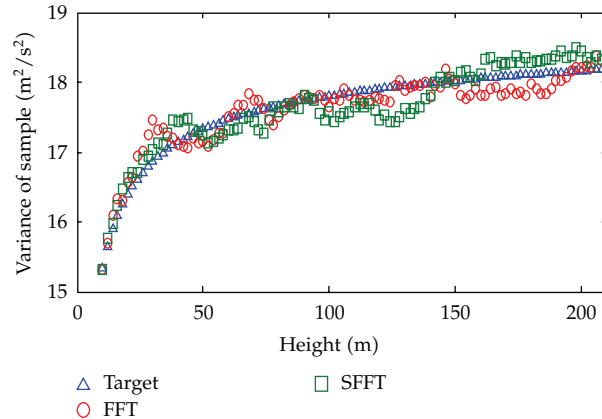
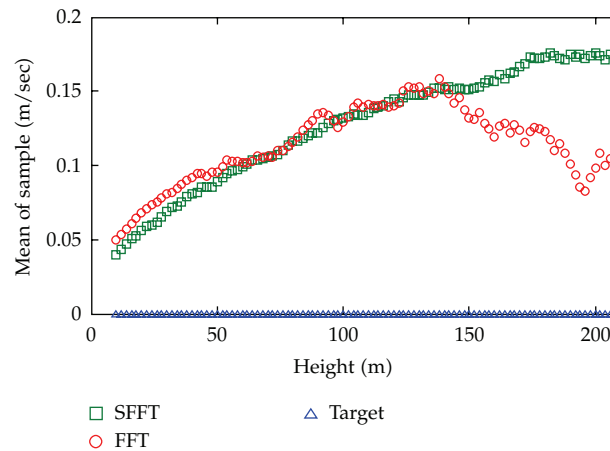


Figure 7: Power spectrum of generated sample functions displayed in Figure 3 versus corresponding targets.



**Figure 8:** Variance values of sample generated by resorting to the SFFT or the FFT with respect to corresponding targets in Case 1.



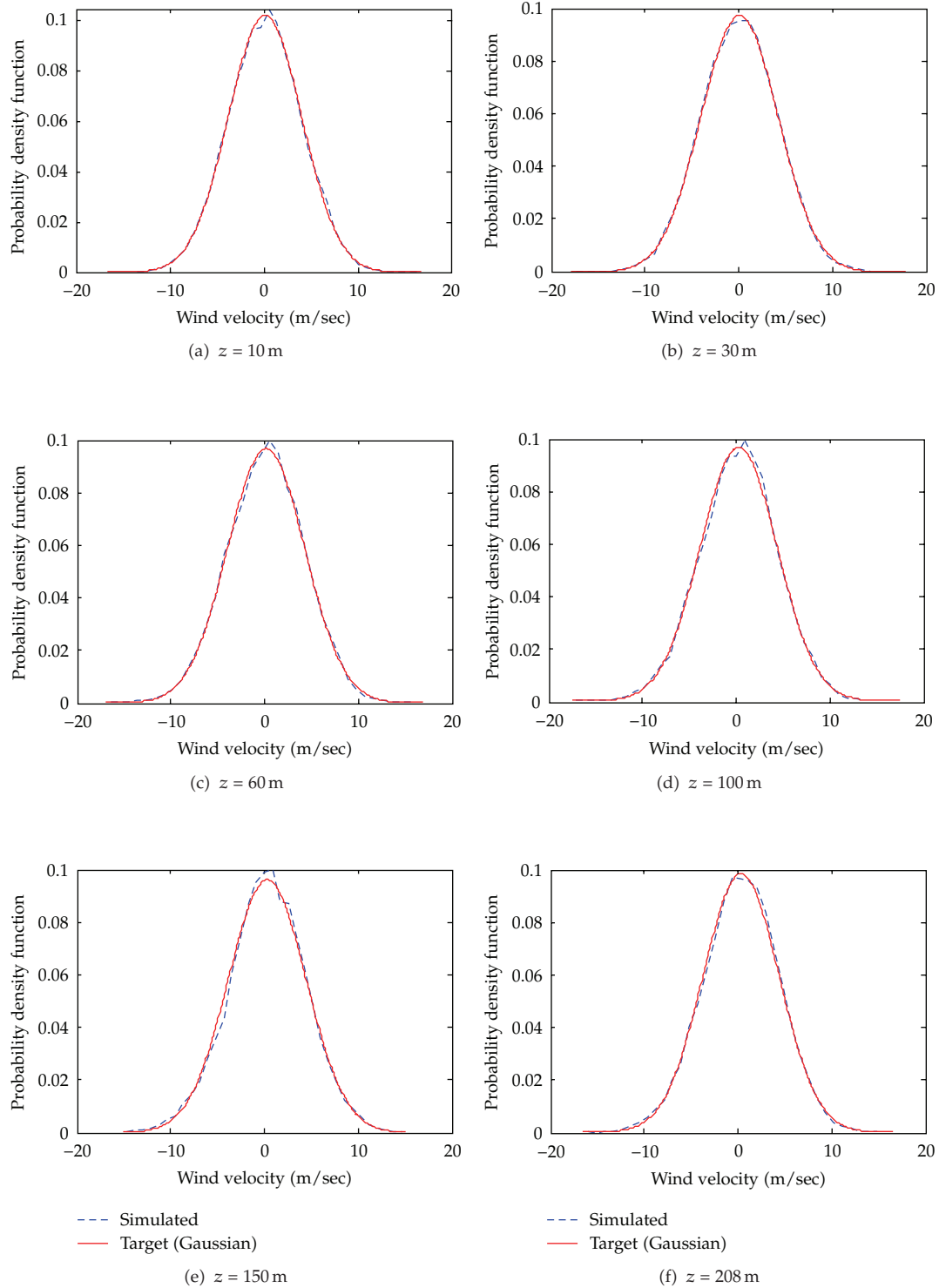
**Figure 9:** Mean values of sample generated with resorting to the SFFT or the FFT with reference to corresponding targets in Case 1.

is the highest due to the minimal time expense. It is from Table 3 observed that the NN-FFT presents the highest accuracy. Worth noting, the accuracy by the SFFT is a little lower than that by the NN-FFT. Forasmuch, the SFFT, generally, is the best.

## 5. Conclusions

This study presented the SFFT algorithm to further enhance the computational efficiency of simulating the multivariate stochastic processes. More specifically, the main purpose of the proposed SFFT algorithm, simultaneously with resorting to the spline interpolation and FFT algorithms, is to reduce the number of the Cholesky decomposition. The main conclusions of the present research effort on the SFFT algorithm can be drawn as follows.

- (1) The proposed SFFT algorithm is able to efficiently generate the longitudinal wind velocity fluctuations that are spatially correlated according to a prescribed coherence function and that possess the phase angles following a prescribed phase angle function.



**Figure 10:** Probability density functions of generated sample functions displayed in Figure 3 versus corresponding target (Gaussian).

**Table 2:** Comparison of time expense among different simulation schemes (noted that the data come from [46]).

Different schemes	Time expense (sec)	
	10 location points	20 location points
SFFT	8.35	29.84
CL-FFT	12.76	47.80
SL-FFT	10.80	40.36
NN-FFT	20.24	74.98
FFT	43.03	301.55

**Table 3:** Comparisons of MRSE and EF among different simulation schemes (noted that the data come from [46]).

Locatio point	MRSE				EF			
	SFFT	CL-FFT	SL-FFT	NN-FFT	SFFT	CL-FFT	SL-FFT	NN-FFT
1	0.1693	0.1896	0.2342	0.1655	0.9812	0.9804	0.9747	0.9818
2	0.1688	0.1791	0.2330	0.1648	0.9810	0.9802	0.9745	0.9817
3	0.1681	0.1782	0.2313	0.1643	0.9814	0.9806	0.9751	0.9821
4	0.1675	0.1774	0.2295	0.1636	0.9813	0.9805	0.9751	0.9819
5	0.1667	0.1764	0.2278	0.1633	0.9808	0.9801	0.9746	0.9815
6	0.1662	0.1758	0.2263	0.1625	0.9809	0.9802	0.9747	0.9816
7	0.1654	0.1748	0.2243	0.1621	0.9811	0.9804	0.9751	0.9817
8	0.1645	0.1737	0.2222	0.1617	0.9810	0.9803	0.9752	0.9816
9	0.1638	0.1727	0.2204	0.1607	0.9813	0.9806	0.9756	0.9819
10	0.1627	0.1713	0.2177	0.1597	0.9820	0.9813	0.9765	0.9825

- (2) The proposed SFFT algorithm has considerable advantage over the FFT algorithm. Likewise, this superiority will become more and more remarkable with the increasing of the parameters  $N$ ,  $n$ , and  $T$ .
- (3) Although introducing the spline interpolation approximation in decomposing the cross-power spectral density matrix, the SFFT and FFT algorithms practically achieve the same computational accuracy.
- (4) Employing the SFFT algorithm, the temporal autocorrelation and cross-correlation functions of any sample function are identical to the target autocorrelation and cross-correlation functions, respectively. Likewise, the estimated spectra derived from these sample time histories and the target spectra are in remarkably good agreement with each other.
- (5) The SFFT, generally, is the best in many interpolation methods.

Eventually, it is worth emphasizing that since the spline interpolation is able to significantly reduce the number of the Cholesky decomposition of time-varying power spectral matrix, the SFFT algorithm can improve the performance of the SR method for the simulation of the multivariate nonstationary stochastic processes. It is known that the power spectrum of nonstationary stochastic process is time-varying; thus, meaning that the FFT algorithm is very hard to be used in this case. Only when the time-varying power spectrum satisfies some special conditions, the FFT can be utilized. However, it is anticipated that

the SFFT algorithm can be used to enhance the computational efficiency of simulating the multivariate nonstationary stochastic processes, since the spline interpolation technique is not affected by the time-varying power spectrum.

## Appendix

### Cholesky Decomposition

For a given CPSD matrix with the element  $|S_{jk}^0(\omega)|e^{i\theta_{jk}}$  ( $k = 1, 2, \dots, n; j = 1, 2, \dots, n$ ), the element of the lower triangular matrix  $H(\omega)$  can be computed as follows:

$$\begin{aligned} H_{11}(\omega) &= |H_{11}(\omega)|e^{i\vartheta_{11}} \\ &= \left[ |S_{11}^0(\omega)|e^{i\theta_{11}} \right]^{1/2} \\ &= \left[ |S_{11}^0(\omega)| \right]^{1/2} e^{i(1/2)\theta_{11}} \\ &\Rightarrow \begin{cases} |H_{11}(\omega)| = \left[ |S_{11}^0(\omega)| \right]^{1/2}, \\ \vartheta_{11} = \theta_{11} = 0, \end{cases} \end{aligned}$$

$$\begin{aligned} H_{j1}(\omega) &= |H_{j1}(\omega)|e^{i\vartheta_{j1}} \\ &= \frac{|S_{j1}^0(\omega)|e^{i\theta_{j1}}}{|H_{11}(\omega)|e^{i\vartheta_{11}}} \\ &= \frac{|S_{j1}^0(\omega)|}{|H_{11}(\omega)|} e^{i\theta_{j1}} \quad (j = 2, 3, \dots, n) \\ &\Rightarrow \begin{cases} |H_{j1}(\omega)| = \frac{|S_{jk}^0(\omega)|}{|H_{11}(\omega)|} & (j = 2, 3, \dots, n), \\ \vartheta_{j1} = \theta_{j1} & (j = 2, 3, \dots, n), \end{cases} \end{aligned}$$

$$\begin{aligned} H_{kk}(\omega) &= |H_{kk}(\omega)|e^{i\vartheta_{kk}} \\ &= \left[ |S_{kk}^0(\omega)|e^{i\theta_{kk}} - \sum_{r=1}^{k-1} |H_{kr}(\omega)|e^{i\vartheta_{kr}} |H_{rk}(\omega)|e^{-i\vartheta_{kr}} \right]^{1/2} \\ &= \left[ |S_{kk}^0(\omega)| - \sum_{r=1}^{k-1} |H_{kr}(\omega)| \cdot |H_{rk}(\omega)| \right]^{1/2} e^{i(1/2)\theta_{kk}} \quad (k = 2, 3, \dots, n) \\ &\Rightarrow \begin{cases} |H_{kk}(\omega)| = \left[ |S_{kk}^0(\omega)| - \sum_{r=1}^{k-1} |H_{kr}(\omega)| \cdot |H_{rk}(\omega)| \right]^{1/2} & (k = 2, 3, \dots, n), \\ \vartheta_{kk} = \theta_{kk} = 0 & (k = 2, 3, \dots, n), \end{cases} \end{aligned}$$

$$\begin{aligned}
H_{jk}(\omega) &= |H_{jk}(\omega)| e^{i\vartheta_{jk}} \\
&= \frac{|S_{jk}^0(\omega)| e^{i\vartheta_{jk}} - \sum_{r=1}^{k-1} |H_{jr}(\omega)| e^{i\vartheta_{jr}} |H_{rk}(\omega)| e^{-i\vartheta_{kr}}}{|H_{kk}(\omega)| e^{i\vartheta_{kk}}} \\
&= \frac{|S_{jk}^0(\omega)| e^{i\vartheta_{jk}} - \sum_{r=1}^{k-1} |H_{jr}(\omega)| \cdot |H_{rk}(\omega)| e^{i(\vartheta_{jr} - \vartheta_{kr})}}{|H_{kk}(\omega)| e^{i\vartheta_{kk}}} \\
&= \frac{|S_{jk}^0(\omega)| e^{i\vartheta_{jk}} - \sum_{r=1}^{k-1} |H_{jr}(\omega)| \cdot |H_{rk}(\omega)| e^{i\vartheta_{jk}}}{|H_{kk}(\omega)| e^{i\vartheta_{kk}}} \\
&= \frac{|S_{jk}^0(\omega)| - \sum_{r=1}^{k-1} |H_{jr}(\omega)| \cdot |H_{rk}(\omega)|}{|H_{kk}(\omega)|} e^{i\vartheta_{jk}} \quad (k = 2, 3, \dots, n; j = k + 1, k + 2, \dots, n) \\
\Rightarrow \begin{cases} |H_{jk}(\omega)| = \frac{|S_{jk}^0(\omega)| - \sum_{r=1}^{k-1} |H_{jr}(\omega)| \cdot |H_{rk}(\omega)|}{|H_{kk}(\omega)|} \\ \vartheta_{jk} = \theta_{jk} \end{cases} & \begin{cases} (k = 1, 2, \dots, n; j = k + 1, k + 2, \dots, n), \\ (k = 1, 2, \dots, n; j = k + 1, k + 2, \dots, n). \end{cases}
\end{aligned} \tag{A.1}$$

Subsequently,  $|H_{jk}(\omega)|$  ( $k = 1, 2, \dots, n; j = k, k + 1, k + 2, \dots, n$ )s can be obtained by resorting to the Cholesky decomposition of a real matrix which consists of the element  $|S_{jk}^0(\omega)|$  ( $k = 1, 2, \dots, n; j = 1, 2, \dots, n$ ). Likewise, the phase angle contents with the relationship  $\vartheta_{jk} = \theta_{jk}$  ( $k = 1, 2, \dots, n; j = 1, 2, \dots, n$ ).

## Acknowledgments

The work reported in this paper has been supported by the National Science Foundation of China with Grant no. 11162005. This support is gratefully acknowledged. The authors wish to thank the reviewers for their careful, unbiased, and constructive suggestions that significantly improved the quality of this paper.

## References

- [1] M. P. Mignolet and P. T. D. Spanos, "Recursive simulation of stationary multivariate random processes. Part I," *American Society of Mechanical Engineers*, vol. 54, no. 3, pp. 674–680, 1987.
- [2] P.-T. D. Spanos and M. P. Mignolet, "Recursive simulation of stationary multivariate random processes. Part II," *American Society of Mechanical Engineers*, vol. 54, no. 3, pp. 681–687, 1987.
- [3] A. Iannuzzi and P. Spinelli, "Artificial wind generation and structural response," *Journal of Structural Engineering*, vol. 113, no. 12, pp. 2382–2398, 1987.
- [4] G. Deodatis and M. Shinozuka, "Autoregressive model for non-stationary stochastic processes," *Journal of Engineering Mechanics*, vol. 114, no. 11, pp. 1995–2012, 1988.
- [5] D. Novak, S. Stoyanoff, and H. Herda, "Error assessment for wind histories generated by autoregressive method," *Structural Safety*, vol. 17, no. 2, pp. 79–90, 1995.
- [6] W. Gersch and J. Yonemoto, "Synthesis of multivariate random vibration systems: a two-stage least squares AR-MA model approach," *Journal of Sound and Vibration*, vol. 52, no. 4, pp. 553–565, 1977.

- [7] F. Kozin, "Autoregressive moving average models of earthquake records," *Probabilistic Engineering Mechanics*, vol. 3, no. 2, pp. 58–63, 1988.
- [8] A. Kareem and Y. Li, "Digital simulation of wind load effects," in *Proceedings of the 6th ASCE Specialty Conference on Probabilistic Mechanics, and Structural and Geotechnical Reliability*, pp. 284–287, July 1992.
- [9] R. Rossi, M. Lazzari, and R. Vitaliani, "Wind field simulation for structural engineering purposes," *International Journal for Numerical Methods in Engineering*, vol. 61, no. 5, pp. 738–763, 2004.
- [10] M. Shinozuka and C. M. Jan, "Digital simulation of random processes and its applications," *Journal of Sound and Vibration*, vol. 25, no. 1, pp. 111–128, 1972.
- [11] M. Grigoriu, "On the spectral representation method in simulation," *Probabilistic Engineering Mechanics*, vol. 8, no. 2, pp. 75–90, 1993.
- [12] G. Deodatis, "Simulation of ergodic multivariate stochastic processes," *Journal of Engineering Mechanics*, vol. 122, no. 8, pp. 778–787, 1996.
- [13] M. Grigoriu, "Algorithm for generating samples of homogeneous Gaussian fields," *Journal of Engineering Mechanics*, vol. 129, no. 1, pp. 43–49, 2003.
- [14] L. Chen and C. W. Letchford, "Simulation of multivariate stationary Gaussian stochastic processes: hybrid spectral representation and proper orthogonal decomposition approach," *Journal of Engineering Mechanics*, vol. 131, no. 8, pp. 801–808, 2005.
- [15] S. O. Rice, "Mathematical analysis of random noise," in *Selected Papers on Noise and Stochastic Processes*, N. Wax, Ed., pp. 133–294, Dover, New York, NY, USA, 1954.
- [16] H. Goto and K. Toki, "Structural response to non-stationary random excitation," in *Proceedings of the 4th World Conference on Earthquake Engineering, (WCEE '69)*, Santiago, Chile, 1969.
- [17] L. E. Borgman, "Ocean wave simulation for engineering design," *Journal of the Waterways and Harbors Division*, vol. 95, no. 4, pp. 557–586, 1969.
- [18] M. Shinozuka, "Simulation of multivariate and multidimensional random processes," *Journal of the Acoustical Society of America*, vol. 49, no. 1, pp. 357–368, 1971.
- [19] M. Shinozuka, "Monte Carlo solution of structural dynamics," *Computers and Structures*, vol. 2, no. 5-6, pp. 855–874, 1972.
- [20] G. Deodatis and M. Shinozuka, "Simulation of seismic ground motion using stochastic waves," *Journal of Engineering Mechanics*, vol. 115, no. 12, pp. 2723–2737, 1989.
- [21] Y. Li and A. Kareem, "Simulation of multivariate random processes: hybrid DFT and digital filtering approach," *Journal of Engineering Mechanics*, vol. 119, no. 5, pp. 1078–1098, 1993.
- [22] M. Shinozuka, M. Kamata, and C. B. Yun, "Simulation of earthquake ground motion as multivariate stochastic process," Tech. Rep. number 1989.5, Princeton-Kajima Joint Research, Department of Civil Engineering and Operations Research, Princeton University, Princeton, NJ, USA, 1989.
- [23] J. N. Yang, "Simulation of random envelope processes," *Journal of Sound and Vibration*, vol. 21, no. 1, pp. 73–85, 1972.
- [24] J. N. Yang, "On the normality and accuracy of simulated random processes," *Journal of Sound and Vibration*, vol. 26, no. 3, pp. 417–428, 1973.
- [25] M. Shinozuka, "Digital simulation of random processes in engineering mechanics with the aid of FFT technique," in *Stochastic Problems in Mechanics*, S. T. Ariaratnam and H. H. E. Leipholz, Eds., pp. 277–286, University of Waterloo Press, Waterloo, Canada, 1974.
- [26] L. E. Wittig and A. K. Sinha, "Simulation of multi-correlated random processes using the FFT Algorithm," *Journal of the Acoustical Society of America*, vol. 58, no. 3, pp. 630–634, 1975.
- [27] Y. Li and A. Kareem, "Simulation of multivariate non-stationary random processes by FFT," *Journal of Engineering Mechanics*, vol. 117, no. 5, pp. 1037–1058, 1991.
- [28] M. Loeve, *Probability Theory*, Van Nostrand, Princeton, NJ, USA, 1963.
- [29] J. D. Holmes, R. Sankaran, K. C. S. Kwok, and M. J. Syme, "Eigenvector modes of fluctuating pressures on low-rise building models," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 69–71, pp. 697–707, 1997.
- [30] M. Di Paola and I. Gullo, "Digital generation of multivariate wind field processes," *Probabilistic Engineering Mechanics*, vol. 16, no. 1, pp. 1–10, 2001.
- [31] M. Rathinam and L. R. Petzold, "A new look at proper orthogonal decomposition," *SIAM Journal on Numerical Analysis*, vol. 41, no. 5, pp. 1893–1925, 2003.
- [32] X. Chen and A. Kareem, "Proper orthogonal decomposition-based modeling, analysis, and simulation of dynamic wind load effects on structures," *Journal of Engineering Mechanics*, vol. 131, no. 4, pp. 325–339, 2005.
- [33] G. Solari and L. Carassale, "Modal transformation tools in structural dynamics and wind engineering," *Wind and Structures*, vol. 3, no. 4, pp. 221–241, 2000.

- [34] L. Carassale and G. Solari, "Monte Carlo simulation of wind velocity fields on complex structures," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 94, no. 5, pp. 323–339, 2006.
- [35] Q. Ding, L. Zhu, and H. Xiang, "Simulation of stationary Gaussian stochastic wind velocity field," *Wind and Structures*, vol. 9, no. 3, pp. 231–243, 2006.
- [36] M. M. Rocha, S. V. S. Cabral, and J. D. Riera, "A comparison of proper orthogonal decomposition and Monte Carlo simulation of wind pressure data," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 84, no. 3, pp. 329–344, 2000.
- [37] X. Chen and A. Kareem, "On the application of stochastic decomposition in the analysis of wind effects," in *Proceedings of the International Conference on Advances in Structural Dynamics, (ASD '00)*, pp. 135–142, Hong Kong Polytechnic University, 2000.
- [38] M. Shinozuka and G. Deodatis, "Simulation of stochastic processes by spectral representation," *Applied Mechanics Reviews*, vol. 44, no. 4, pp. 191–204, 1991.
- [39] G. Deodatis and R. C. Micaletti, "Simulation of highly skewed non-Gaussian stochastic processes," *Journal of Engineering Mechanics*, vol. 127, no. 12, pp. 1284–1295, 2001.
- [40] J. C. Kaimal, J. C. Wyngaard, Y. Izumi, and O. R. Cote, "Spectral characteristics of surface-layer turbulence," *Journal Publications Royal Meteorological Society*, vol. 98, pp. 563–589, 1972.
- [41] A. G. Davenport, "The dependence of wind load upon meteorological parameters," in *Proceedings of the International Research Seminar on Wind Effects on Building and Structures*, pp. 19–82, University of Toronto Press, Toronto, Canada, 1968.
- [42] L. Kristensen and N. O. Jensen, "Lateral coherence in isotropic turbulence and in the natural wind," *Boundary Layer Meteorology*, vol. 17, no. 3, pp. 353–373, 1979.
- [43] E. Simiu and R. H. Scanlan, *Wind Effects on Structures*, John Wiley & Sons, New York, NY, USA, 1986.
- [44] M. Di Paola, "Digital simulation of wind field velocity," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 74–76, pp. 91–109, 1998.
- [45] U. Peil and G. Telljohann, "Lateral turbulence and dynamic response," *Eurodyn*, vol. 96, no. 1, pp. 207–211, 1996.
- [46] C. Z. Liu, *Numerical simulation and prediction of fluctuating wind velocity time series of civil engineering structures*, M.S. thesis, Department of Civil Engineering, Shanghai University, Shanghai, China, 2009.
- [47] J. Y. Fu, S. G. Liang, and Q. S. Li, "Prediction of wind-induced pressures on a large gymnasium roof using artificial neural networks," *Computers and Structures*, vol. 85, no. 3-4, pp. 179–192, 2007.





# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

