Electrical Power and Energy Systems 104 (2019) 472–480

Contents lists available at ScienceDirect

Electrical Power and Energy Systems

journal homepage: www.elsevier.com/locate/ijepes

Probabilistic power flow computation using quadrature rules based on discrete Fourier transformation matrix

Qing Xiao, Shaowu Zhou*

College of Mechanical and Electrical Engineering, Hunan University of Science and Technology, Xiangtan, Hunan Province, China

A R T I C L E I N F O

Keywords:
Probabilistic power flow
Nataf transformation
Discrete Fourier transformation matrix
Multivariate quadrature rule

A B S T R A C T

This paper sets out to develop an efficient algorithm for probabilistic power flow (PPF) computation. Nataf transformation is introduced to transform PPF problem to the independent standard normal space, an algorithm based on Hermite polynomials is employed to determine the equivalent correlation coefficient in normal space. Using the real part and imaginary part of discrete Fourier transformation matrix (DFTM), two quadrature rules are developed for PPF computation. Testing on a modified IEEE 118-bus system including wind farms, the proposed methods are compared with the point estimate method (PEM) for calculating the mean and standard deviation of PPF outputs, a detailed discussion is also given for the accuracy of these two algorithms.

1. Introduction

As renewable generators are integrated into modern power systems, the probabilistic power flow (PPF) technique is often introduced to handle the uncertainties in power flow equations. In the context of PPF, the random variables in power flow equations can be referred to as inputs; the solutions of power flow equations are called outputs, and PPF aims to obtain statistical information of outputs. Hitherto, various algorithms have been proposed for PPF computation, and they can be classified as probabilistic methods and constructive methods.

Monte Carlo simulation (MCS) epitomizes probabilistic methods. According to the probability distributions of PPF inputs, MCS generates samples to represent various scenarios in power systems, and performs deterministic power flow equations to yield samples of bus voltages, phase angles, active power flow and reactive power flow, whereby cumulative distribution functions (CDF) of PPF outputs can be established [1,2]. If correlated non-normal PPF inputs are involved, Nataf transformation can be used to generate samples with prescribed marginal distributions and correlation matrix [3]. However, when MCS is used for PPF computation, a large sample size is required to yield convergent and accurate results, leading to a heavy computational burden. In order to accelerate the convergence rate, more effective sampling techniques are developed, such as Quasi-Monte Carlo simulation [4], Latin hypercube sampling [5–8], Latin supercube sampling [9] and blind number theory method [10].

The constructive methods employ a multivariate polynomial model to represent the function relationship between PPF inputs and outputs: $y = H(\boldsymbol{X})$ (see Eq. (17) of the paper):

$$y = H(\boldsymbol{X}) \simeq \sum_{j} a_j x_1^{l_1} \cdots x_i^{l_i} \cdots x_m^{l_m}. \tag{1}$$

The first kind of constructive methods aim to determine the coefficients a_j, then $y = H(\boldsymbol{X})$ can be represented by an explicit polynomial model, and the statistical moments of y can be easily obtained. Because an rth-order orthogonal polynomial $P_r(x)$ is a linear combination of $1, x, x^2, \ldots, x^r$, the surrogate model in Eq. (1) can also be expressed as a sum of orthogonal polynomials [11,12]:

$$y \simeq \sum_{j} c_j P_{l_1}(x_1) \cdots P_{l_i}(x_i) \cdots P_{l_m}(x_m), \tag{2}$$

then the coefficients c_j can be determined using the orthogonality property of orthogonal polynomials. When the number of inputs m is large, the model in Eq. (2) based on tensor product structure would suffer the curse of dimensionality. Besides, the cumulant method [13–16] may also fall in this category, which employs cumulants to characterize the statistical feature of PPF inputs, and determines the coefficients by Jacobian matrix of power flow equations.

The unscented transformation (UT) method [17,18], point estimate method (PEM) [19–21] and Taguchi method [22,23] can be classified as the second kind of constructive methods. The emphasis of these algorithms is not on coefficients, but to develop a multivariate quadrature rule to calculate the moment of each monomial in Eq. (1). More specifically, they select a set of points $(t_{1,s}, \ldots, t_{i,s}, \ldots, t_{m,s})$ $(s = 1, \ldots, n)$, and assign each point with a weight p_s, such that the following equations

* Corresponding author.
E-mail address: swzhou@hnust.edu.cn (S. Zhou).

https://doi.org/10.1016/j.ijepes.2018.07.021
Received 2 March 2018; Received in revised form 19 May 2018; Accepted 11 July 2018
0142-0615/

can be satisfied for all monomials in Eq. (1):

$$\sum_{s=1}^{n} p_s t_{1,s}^{l_1}\cdots,t_{i,s}^{l_i}\cdots,t_{m,s}^{l_m} = E[x_1^{l_1}\cdots x_i^{l_i}\cdots x_m^{l_m}], \tag{3}$$

With different polynomial models in Eq. (1), different algorithms can be developed for PPF computation. The computational burden of UT and PEM increases linearly with respect to the number of PPF inputs, while the computational burden of Taguchi method depends on the number of PPF inputs and the level of the orthogonal array, a higher level helps improve the accuracy, but it would also dramatically increase the computational burden [23].

This paper aims to develop an efficient multivariate quadrature rule for PPF computation. With Nataf transformation, PPF problem is mapped to independent standard normal space, and each PPF outputs is expressed as a multiple integral with an implicit integrand. Using the real part and imaginary part of discrete Fourier transformation matrix (DFTM), two quadrature rules are developed, and an analysis of these two algorithms is also given in the context of the moment matching equations.

The rest of the paper is outlined as follows: Section 2 introduces Nataf transformation and presents a new methodology to calculate the equivalent correlation coefficient in normal space. Section 3 formulates the PPF problem, and presents the procedures of multivariate quadrature rules for PPF computation. In Section 4, two quadrature rules are developed. In Section 5, a case study is performed, and a discussion is given on the performance of the proposed methods for PPF computation. Finally, Section 6 gives some relevant conclusions.

2. Nataf transformation

Most algorithms developed for PPF computation require that random variables in power flow equations should be independent of each other. For PPF problem involving correlated inputs, Nataf transformation can be employed to transform correlated random variables to independent standard normal variables.

2.1. Framework of Nataf transformation

Let $\boldsymbol{X} = (x_1, ..., x_i, ..., x_m)$ be a random vector, let $F_i(x_i)$ be the cumulative distribution function (CDF) of x_i $(i = 1, ..., m)$. Using the marginal transformation, x_i can be obtained by [24]:

$$x_i = F_i^{-1}[\Phi(z_i)] \tag{4}$$

where $F_i^{-1}(\cdot)$ is the inverse CDF of x_i, z_i is a standard normal variable, $\Phi(\cdot)$ is the CDF of z_i.

With Eq. (4), $\boldsymbol{X}$ can be generated from a standard normal vector $\boldsymbol{Z} = (z_1, ..., z_i, ..., z_m)$:

$$\begin{pmatrix} x_1 \\ \vdots \\ x_i \\ \vdots \\ x_m \end{pmatrix} = \begin{pmatrix} F_1^{-1}[\Phi(z_1)] \\ \vdots \\ F_i^{-1}[\Phi(z_i)] \\ \vdots \\ F_m^{-1}[\Phi(z_m)] \end{pmatrix}. \tag{5}$$

In order to generate a correlated random vector $\boldsymbol{X}$, a correlated standard normal vector $\boldsymbol{Z}$ should be employed. Let $\boldsymbol{R_X} = \{\rho_x(i, j)\}$ $(i, j = 1, ..., m)$ be the correlation matrix of $\boldsymbol{X}$;$\rho_x(i, j)$ denotes the correlation coefficient between x_i and x_j. Let $\boldsymbol{R_Z} = \{\rho_z(i, j)\}$ be the correlation matrix of $\boldsymbol{Z}$;$\rho_z(i, j)$ denotes the correlation coefficient between z_i and z_j. In general, $\rho_x(i, j) \neq \rho_z(i, j)$, and an appropriate value of $\rho_z(i, j)$ should be determined. This issue can be handled by the algorithm in Section 2.2.

By matching $\rho_z(i, j)$ to $\rho_x(i, j)$, a correlation matrix $\boldsymbol{R_Z}$ of $\boldsymbol{Z}$ can be obtained, and $\boldsymbol{X}$ can be generated from an independent standard normal vector $\boldsymbol{U} = (u_1, ..., u_i, ..., u_m)$. Below are the detailed procedures:

1. Perform Cholesky decomposition on $\boldsymbol{R_Z}$ to obtain the lower triangular matrix $\boldsymbol{L}$:

$$\boldsymbol{R_Z} = \boldsymbol{L}\boldsymbol{L}^T. \tag{6}$$

2. Transform $\boldsymbol{U}$ to the correlated standard normal vector $\boldsymbol{Z}$:

$$\boldsymbol{Z} = \boldsymbol{L}\boldsymbol{U}, \tag{7}$$

and the correlation matrix of $\boldsymbol{Z}$ would be $\boldsymbol{R_Z}$.

3. Based on inverse CDFs of x_i $(i = 1, ..., m)$, transform $\boldsymbol{Z}$ to $\boldsymbol{X}$ by Eq. (5).

The procedures can also be expressed as:

$$\begin{pmatrix} u_1 \\ \vdots \\ u_i \\ \vdots \\ u_m \end{pmatrix} \xrightarrow[\boldsymbol{Z}=\ \boldsymbol{LU}]{\boldsymbol{R_Z}=\ \boldsymbol{LL}^T\ \downarrow} \begin{pmatrix} z_1 \\ \vdots \\ z_i \\ \vdots \\ z_m \end{pmatrix} \xrightarrow{x_i=F_i^{-1}[\Phi(z_i)]} \begin{pmatrix} x_1 \\ \vdots \\ x_i \\ \vdots \\ x_m \end{pmatrix}. \tag{8}$$

Note that the above transformation defines an implicit function relationship between a correlated random vector $\boldsymbol{X}$ and an independent standard normal vector $\boldsymbol{U}$, which can be denoted as:

$$\boldsymbol{X} = \psi(\boldsymbol{U}). \tag{9}$$

2.2. Determining equivalent correlation coefficient in standard normal space

Let x_i and x_j be two arbitrary random variables in $\boldsymbol{X}$. Here, a weighted sum of Hermite polynomials are employed to approximate the marginal transformation in Eq. (4):

$$\begin{aligned} x_i &= F_i^{-1}[\Phi(z_i)] \simeq \sum_{k=0}^{n} a_k H_k(z_i), \\ x_j &= F_j^{-1}[\Phi(z_j)] \simeq \sum_{l=0}^{n} b_l H_l(z_j). \end{aligned} \tag{10}$$

where $H_k(z)$ is the kth-order Hermite polynomial. The coefficients a_k and b_l in Eq. (10) can be calculated as [25]:

$$\begin{aligned} a_k &= \tfrac{1}{k!}\int_{-\infty}^{\infty} H_k(z_i)F_i^{-1}[\Phi(z_i)]\phi(z_i)dz_i, \\ b_l &= \tfrac{1}{l!}\int_{-\infty}^{\infty} H_l(z_j)F_j^{-1}[\Phi(z_j)]\phi(z_j)dz_j, \end{aligned} \tag{11}$$

where $\phi(\cdot)$ is the probability distribution function (PDF) of a standard normal variable.

For a concise expression, denote $\rho_x(i, j)$ by ρ_x, denote $\rho_z(i, j)$ by ρ_z. According to Eq. (10), it has:

$$\rho_x\sigma_i\sigma_j + \mu_i\mu_j = E[x_i x_j] \simeq \sum_{k=0}^{n}\sum_{l=0}^{n} a_k b_l E[H_k(z_i)H_l(z_j)], \tag{12}$$

where μ_i, μ_j denote the means of x_i, x_j respectively; σ_i, σ_j denote the standard deviations respectively.

Because [26]:

$$E[H_k(z_i)H_l(z_j)] = \begin{cases} k!\rho_z^k & k = l \\ 0 & k \neq l, \end{cases} \tag{13}$$

it has:

$$\rho_x\sigma_i\sigma_j + \mu_i\mu_j \simeq \sum_{k=0}^{n} k! a_k b_k \rho_z^k. \tag{14}$$

As shown in Eq. (14), ρ_x is expressed as a polynomial of ρ_z. For a given ρ_x between x_i and x_j, ρ_z can be determined by solving the polynomial equation in Eq. (14), a valid solution is restricted by:

$$-1 \leqslant \rho_z \leq 1 \quad \text{and } \rho_z\rho_x \geqslant 0. \tag{15}$$