A Coupled Karhunen–Loève and Anisotropic Sparse Grid Interpolation Method for the Probabilistic Load Flow Problem

Brandon Johnson, Nathan L. Gibson, and Eduardo Cotilla-Sanchez

Oregon State University, Corvallis, OR, 97330, USA

Abstract

In the traditional load flow analysis, a key assumption is that the input variables, i.e., generator output and customer demand, are fixed in time and the associated response has no variability. This assumption, however, is no longer valid as the adoption of renewable energy resources add more variability and uncertainty to the modern electrical system. Addressing these concerns is the definition of the Probabilistic Load Flow (PLF) problem. The challenge of the PLF problem lies in handling high-dimensional input uncertainties and the non-linearity of the load flow equations. The most straightforward way to address these problems, but at the cost of computational time, is to perform a Monte Carlo method. This work, however, solves these problems—accuracy, high-dimensionality, and computational time—with a coupled Karhunen-Loève (KL) expansion and Anisotropic Sparse Grid algorithm. The proposed method is implemented and tested on the IEEE 118-bus test system and a modernized version of the IEEE Reliability Test System–1996, the Reliability Test System–Grid Modernization Lab Consortium (RTS-GMLC). Results for the 194-dimensional case show a decrease in computational time when compared to the 10,000 sample Monte Carlo method given a bound on mean and standard deviation error.

Keywords: Probabilistic power flow, sparse grid interpolation, stochastic collocation, Karhunen–Loève expansion, uncertainty quantification.

1. Introduction

Load flow studies are one of the most essential tools for grid operators because they provide a clear picture of the operating conditions of the power

system. Traditionally, the solution to a load flow analysis is assumed to be deterministic; that is, the system variables contain no randomness. In recent years, distributed energy resources, such as wind and solar, are being adopted at an increasing rate causing the power system to be subjected to increasing levels of variability. The intrinsic variability of renewable resources stems from the inability to accurately predict local weather patterns at relatively small and discrete time-steps. As a result of these changes in the power system, it is no longer sufficient to assume that both generation and demand are constants in the load flow problem. The reality is that it is much more accurate to assess the power systems response to a range of possible inputs.

The Probabilistic Load Flow (PLF) problem was first proposed by Borkowska [1] with the goal of handling uncertainties, e.g., variability in the energy supply, within a power flow analysis. Since then, there are two major ways of solving the PLF problem: numerically, i.e., using a Monte Carlo method [2, 3], or analytically, i.e., using a convolutional method [4]. As in the study of other complex systems, in power systems Monte Carlo methods allow for a simplistic approach for solving the PLF, but come with the downside of high computational burden. As a result of this characteristic, Monte Carlo methods are more suited as a baseline for checking the accuracy of other methods. Analytical methods, on the other hand, are mathematically complex and are subject to inaccuracies that vary depending on assumptions and approximations, but have the capability to solve the PLF problem in real time–as it is needed in operations and control.

The objective of analytical methods is to obtain the probability distribution function (PDF) and cumulative distribution function (CDF) of the state vector, i.e., active and reactive power injections, voltage, and angle. The difficulty of this problem lies within the non-linear nature of the load-flow equations and that the uncertain parameters that affect the power system are unlikely to be independent. To remedy these two problems researchers have employed these two relatively simple steps: 1) linearize the load flow equations and 2) assume that the uncertain parameters are independent; these two steps alone enable the use of convolutional techniques. Unfortunately, convolutional methods have three major downsides: 1) linearizing the power-flow equations around the operating point is sufficient only when deviations are close to the operating point 2) in most cases, the assumption of statistical independence of input variables is often false [5] and 3) convolutional techniques require a large amount of memory. For example, in the case where two random variables are represented by discrete functions with k impulses,

the resulting convolution will have k times k impulses [6].

Recent work in this area has attempted to forego the convolutional method and move in favor of methods that seek to approximate the target output variables using cumulant and polynomial expansion methods. References [6, 7] use the properties of cumulants–an alternative to moments–and the Gram-Charlier expansion to reduce the computation time while still maintaining a comparable accuracy to that of Monte Carlo methods. References [8, 9] have used a polynomial chaos expansion to reformulate the stochastic problem into a purely deterministic one without the use of random sampling. Similarly, Sun *et. al* [10] applied a sparse polynomial chaos expansion to a small test system with four wind farms and 21 random load sources. Reference [11] uses perturbation methods, derived from Galerkin methods, to resolve issues using polynomial approximations of the load flow equations. While the main benefit of these methods are the significant reduction in computation time, the assumptions made to calculate the output PDF lead to inaccuracies.

A third group of methods, called approximation methods, seek to leverage the efficiency of deterministic solvers to build the output PDFs, much like Monte Carlo methods, but with far fewer simulations. The first method of this kind–introduced in 1975 by Rosenblueth–is called the Point Estimate Method (PEM) [12]. The PEM bundles statistical information using the first few central moments of the input variable into K points for each variable, called *concentrations*. Rosenblueth's original work showed that the probabilistic problem can be reduced to 2^m deterministic simulations, where m is the number of random input variables. Since then, adaptations have been made to the original problem that can reduce the number of simulations to as little as $2m + 1$ while still retaining a high-level of accuracy [13, 14, 15]. While the PEMs possess the speed of the analytical solutions and the accuracy of Monte Carlo methods, they can only produce statistical moment information of the output variables. One solution to overcome this problem is to employ Stochastic Collocation (SC) methods.

While technically an approximate method, SC methods are able to produce the necessary PDFs and CDFs to assess the risk of a system with uncertain input variables. Much like PEMs, one creates a set of nodes, called *collocation* points, and then the model is solved using a deterministic solver at each node. One approach to the SC problem is to use the interpolation approach with Tensor Grids [16, 17]. Given the high-dimensionality of the PLF problem, it is more practical to use a subset of full Tensor Grids called Smolyak sparse grids [18]. Sparse grids are capable of reducing the impact of

high-dimensionality that afflicts other approximation methods. The authors in [19, 20] use a dimension-adaptive sparse grid interpolation, in conjunction with Copula Theory principles, to explore the impact of dependent random variables on the PLF problem. While being fairly accurate, this method is computationally burdensome due to having to determine the sparse grid as the problem evolves. This rules out the option of parallelizing the deterministic load flow simulations due to the interdependence of each simulation.

We should note that sparse grids alone do not reduce the dimension of the problem, therefore, we propose to apply the Karhunen–Loève (KL) expansion [21, 22, 23] to the random fields. The KL-expansion is a dimension reduction technique that, much like Principal Component Analysis (PCA), seeks to represent a random process as a series combination of a complete set of deterministic functions with corresponding random coefficients. Our work also proposes coupling the KL-expansion with an *anisotropic* sparse grid [24], a more generalized Smolyak sparse grid, further improving the performance of approximate methods. By reducing the number of dimensions and assigning importance to each dimension, our method provides a significant computational savings while maintaining a comparable accuracy to existing methods. Furthermore, the PDF and CDF of the output can be computed directly by using the sparse grid interpolant.

The remainder of the paper is organized as follows. Section 2 includes a mathematical formulation for the KL-expansion. Section 3 provides a formal background for SC and the mathematical framework for anisotropic sparse grids. Then, in Section 4 we discuss how both the KL-expansion and ansiotropic sparse grids are coupled to solve the PLF problem. Thereafter, in Section 5 we outline the implementation and metrics used for accuracy analysis. Lastly, Section 6 shows the results of implementing the proposed method and the concluding remarks follow in Section 7.

2. Dimension Reduction and the Karhunen–Loève (KL) Expansion

2.1. Mathematical Formulation

We begin by assuming there is a (spatial) correlation among the random input variables, and thus treat each as a sample of an underlying spatial stochastic process, which we represent efficiently using a KL-expansion, i.e., a series expansion in orthogonal eigenfunctions of the covariance structure. For example, multiple independent solar farms (in a given region) can be treated as a random process. One can treat them as either temporally (start

4

producing near the same time) or spatially (similar solar irradiance, cloud coverage, etc.) correlated where the actual value of solar production is a random variable based on several other random variables. Consider the random process with two variables $Y(x, \omega)$ where x is an index parameter (representing either time or space) and the outcome $\omega \in \Omega$, where Ω is the space of the underlying random variables. The random process $Y(x, \omega)$ has mean $\bar{Y}(x)$ and covariance function $C(x, s) = \text{Cov}(Y(x), Y(s))$. Then, $Y(x, \omega)$ has the orthogonal decomposition

$$Y(x, \omega) = \bar{Y}(x) + \sum_{n=1}^{\infty} \sqrt{\lambda_n} \phi_n(x) \xi_n(\omega), \tag{1}$$

where $\{(\lambda_n, \phi_n)\}_{n=1}^{\infty}$ are the eigenvalues and orthogonal eigenfunctions, which are the solutions to a Frendholm integral equation of the second kind, defined as

$$\lambda \phi(s) = \int C(x, s) \phi(x) dx. \tag{2}$$

The random variables $\{\xi_n\}_{n=1}^{\infty}$ in (1) are uncorrelated and satisfy

$$E[\xi_n] = 0, \quad \text{Cov}[\xi_n, \xi_k] = \delta_{nk}, \quad n, k \geq 1, \tag{3}$$

where δ_{nk} is the Kronecker-delta function. It is often practical to truncate the KL-expansion in (1) to a finite number of terms d. The truncated KL-expansion then becomes

$$Y(x, \omega) \approx \bar{Y}(x) + \sum_{n=1}^{d} \sqrt{\lambda_n} \phi_n(x) \xi_n(\omega), \quad d \geq 1. \tag{4}$$

The number of d terms to keep in the truncated KL-expansion is often determined by examining the decay rate of the eigenvalues λ_n as the index n increases. The decay rate of the eigenvalues depends inversely on the correlation length of the random process [17]. This implies that having a long correlation length (i.e., strongly correlated variables) results in a fast decay of the eigenvalues. Conversely, an uncorrelated process with zero correlation length proves to be the limit of this method in which there is no eigenvalue decay. It is common to select d such that the remaining eigenvalue contributions are negligible. In practice, it is often possible to capture 90% of the input variability with only 10 variables.

2.2. The Need for Stochastic Collocation

While we will cover stochastic collocation in more detail in Section 3, it is important to discuss why the KL-expansion is necessary in the first place. Stochastic collocation methods, like PEMs, are very desirable because they help solve the PLF problem non-intrusively. That is, one can leverage fast deterministic solvers on a given set of samples or collocation points. In this way, PEMs may also be considered a collocation method. Unfortunately, in large problem spaces collocation methods suffer from something called the "curse of dimensionality", i.e., the problem scales poorly in high dimensions. One way to overcome this limitation is to use a sparse set of collocation points. Although this lessens the impact caused by high-dimensionality, it does not effectively reduce the number of dimensions in the original problem space. It is in this way how–by using the KL-expansion to reduce the dimensionality–collocation methods become computationally tractable. We choose each ξ_n in (4) to be an independent random variable uniformly distributed over the interval $[-1, 1]$. This then allows us to use quadrature rules (please see Section 4) to deterministically sample each ξ_n; thus, mapping the stochastic problem into a deterministic one. Note that by choosing ξ_n to be uniformly distributed over $[-1, 1]$ means one must scale the KL-expansion by $\sqrt{3}$ to ensure that each variable has a variance of 1.

3. Stochastic Collocation and Tensor Grid Interpolants

The goal of SC methods is to solve a set of governing equations (e.g., a set of partial differential equations) at discrete nodes called collocation points. In this context, Monte Carlo methods can also be considered a collocation method where the collocation points are determined randomly. In cases where the input distributions are compatible with Askey scheme orthogonal polynomials, it is more efficient to use cubature rules to determine the nodal set [17]. With this in mind, the following section serves as the mathematical foundation for determining the nodal set used for tensor grid and sparse grid collocation methods. For more details on the methods presented here please see references [18, 25, 26, 24, 17].

3.1. Tensor Product Collocation

For multivariate cases with dimensionality $d > 1$, for any $1 \leq i \leq d$, let Q_{m_i} be an interpolating operator such that

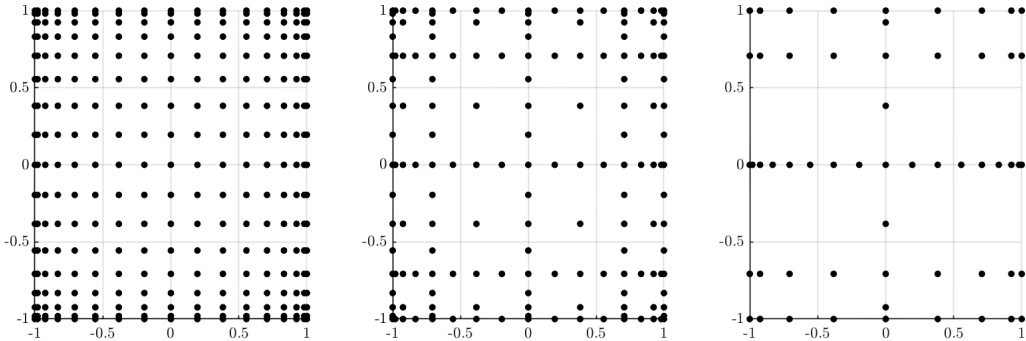$$Q_{m_i}[f] = \Pi_{m_i} f(Z_i) \in \mathbb{P}_{m_i}(Z_i), \tag{5}$$

Figure 1: Two-dimensional nodes for one-dimensional extrema of the Chebyshev polynomials at level $k = 5$. Left: Full Tensor Grid. The total number of nodes is 289. Middle: Smolyak Sparse Grid. The total number of nodes is 161. Right: Anisotropic Sparse Grid with $\gamma_2/\gamma_1 = 2$. The total number of nodes is 57.

is an interpolating polynomial of degree m_i, for a given function f in the Z_i, variable by using $m_i + 1$ distinct nodes in the set $\Theta^{m_i}{}_1 = \{Z^{(1)}{}_1, \ldots, Z^{m_i}{}_i\}$. Then the most straightforward approach to interpolating f in the entire space $I_z \subset \mathbb{R}^d$ is to use the tensor product approach. That is,

$$Q_M = Q_{m_1} \otimes \cdots \otimes Q_{m_d}, \tag{6}$$

and the nodal set is

$$\Theta_M = \Theta_1^{m_1} \times \cdots \times \Theta_1^{m_d}, \tag{7}$$

where the total number of nodes is $M = m_1 \times \cdots \times m_d$. In the case where the total number of points $M = m^d$, the interpolation error follows

$$(I - Q_M)f[m] \propto M^{-\alpha/d}, \tag{8}$$

where the constant $\alpha > 0$ depends on the smoothness of the function f [17]. For large dimension $d \gg 1$, the total number of points $M = m^d$ grows very fast for large d. Again, this is commonly referred to as the "curse of dimensionality."

3.2. Sparse Grid Collocation

A much more practical approach rather than the full Tensor Product Collocation are Smolyak sparse grids [18]. The Smolyak sparse grid construction is based on the tensor product construction but they only represent a subset of the full tensor grid. The construction of the grid takes the form

$$Q_N = \sum_{N-d+1 \leq |\boldsymbol{i}| \leq N} (-1)^{N-|\boldsymbol{i}|} \binom{d-1}{N-|\boldsymbol{i}|} \cdot (Q_{i_1} \otimes \cdots \otimes Q_{i_d}), \qquad (9)$$

where $N \geq d$ is an integer denoting the *level* of the construction. The nodal set, i.e., the sparse grid, is

$$\Theta_M = \bigcup_{N-d+1 \leq |\boldsymbol{i}| \leq N} (\Theta_1^{i_1} \times \cdots \times \Theta_1^{i_d}). \qquad (10)$$

It is typically desired for the grid to take a nested form. One popular choice for a nested grid are the Clenshaw-Curtis nodes, which are the extrema of the Chebyshev polynomials, and are defined as

$$Z_j^{(i)} = -\cos \frac{\pi(j-1)}{m_i^k - 1}, \qquad j = 1, ..., m_i^k, \qquad (11)$$

where an additional index is introduced via the subscript k, often described as the level of the Clenshaw-Curtis grid (a higher level creates a finer grid). While there is not a closed form expression for the total collocation points M in terms of k and d, in high-dimensional problems the total number of points is estimated by

$$M \sim 2^k d^k / k! \qquad d \gg 1. \qquad (12)$$

Although the curse of dimensionality has been lessened through the use of sparse grids, it still exists.

3.3. Anisotropic Sparse Grid Collocation

Until now the focus of sparse grids has been on methods that treat each dimension *isotropically*; that is, each dimension is treated equally. This assumption, however, is only valid for problems where there is either a weak dimensional dependence or that dependence is unknown. If, on the other hand, the problem exhibits a strong dimension-dependent variation then it is desirable to use an anisotropic formulation of the sparse grid problem [24, 27].

Let $\boldsymbol{i}$ be a d-dimensional level vector where each dimension n is defined by $i_n \in \mathbb{N}_+$ and let $\gamma = \{\gamma_1, \gamma_2, \ldots, \gamma_d\} \in \mathbb{R}_+^d$ be a d-dimensional weight vector for each stochastic dimension ω. We then denote the anisotropic sparse

grid of 0-based index w, spatial dimension d, and anisotropy vector γ, by $\mathbb{A}(w, d, \gamma)$, which has the form

$$\mathbb{A}(w, d, \gamma) = \sum_{\mathbf{i} \in Y_\gamma(w,d)} c_\gamma(\mathbf{i})(Q_{i_1} \otimes \cdots \otimes Q_{i_d}), \tag{13}$$

with

$$c_\gamma(\mathbf{i}) = \sum_{\substack{\mathbf{j} \in \{0,1\}^d \\ \mathbf{i}+\mathbf{j} \in X_\gamma(w,d)}} (-1)^{|\mathbf{j}|}, \tag{14}$$

and the selection region of acceptable product rules defined by

$$Y_\gamma(w, d) = X_\gamma(w, d) \backslash X_\gamma\left(w - \frac{|\gamma|}{\underline{\gamma}}, d\right), \tag{15}$$

where

$$X_\gamma(w, d) = \left\{ \mathbf{i} \in \mathbb{N}_+^d, \mathbf{i} \geq \mathbf{1} : \sum_{n=1}^d (i_n - 1)\gamma_n \leq w\underline{\gamma} \right\}, \tag{16}$$

with $\underline{\gamma} = \min_{1 \leq n \leq d} \gamma_n$ and $|\gamma| = \sum_{n=1}^d \gamma_n$. The *selection criteria* written in (15) and (16) can be simplified to the following expression:

$$w\underline{\gamma} - |\gamma| < \sum_{n=1}^d (i_n - 1)\gamma_n \leq w\underline{\gamma}. \tag{17}$$

For a product rule to be included in the sparse grid of sparse grid level w, the product rule's level vector $\mathbf{i}$ must satisfy the above criteria.

As stated before, if each dimension is not equally important, then the anisotropic method can account for that by appropriately setting the weight vector. For instance, if the first dimension contributes 25% more to the variance of the problem than the second, then the ratio of those two components would be $\gamma_2/\gamma_1 = 4/3$. This may seem counterintuitive but in the anisotropic formulation a lower weight value means a higher importance. Additionally, generating the isotropic Smolyak sparse grid is a special case of the anisotropic method. This is done by setting the components of the weight vector to be equal, i.e., $\gamma_1 = \gamma_2 = \cdots = \gamma_d$. In this sense, the anisotropic method is the more generalized version of the sparse grid method.

One more important thing to note about anisotropic sparse grid is that, while the weight vector may be chosen adaptively, as shown in [26], the

9

relative weight vector can also be chosen using *a priori* or *a posteriori* information instead. In the former, the weight vector can be found by using information about the eigenpairs of the covariance matrix (λ_n, ϕ_n). In the latter, the dependence of the nth input variable can be found by simply "freezing" the other input variables to their mean and observe how the output varies [24].

4. The Coupled KL-Expansion and Anisotropic Sparse Grid applied to the PLF problem

With the previous sections in mind, Fig. 2 depicts the full proposed solution to the PLF problem through the use of a coupled KL-expansion and anisotropic sparse grid method. First, in Step 1 we compute the means of each input variable, the eigenpairs (λ, ϕ) using (2)-(3), and determine the d number of eigenpairs to keep. Next, in Step 2 we generate the sparse grid for the stochastic collocation method. As discussed at the beginning of Section 3, it is typical to use cubature rules for the generation of the collocation nodes; the most common of which is the Clenshaw-Curtis quadrature rule [28]. This is because the Clenshaw-Curtis quadrature rule theoretically produces the fewest number of nodes and will, thereby, be the fastest. Typically, one needs to be cautious in using quadratures rules in high order (dimension) systems. In certain cases, and with some quadrature rules, there can be large negative weights (not to be confused with the anisotropic weight vector) associated with the quadrature nodes. The presence of large negative weights can introduce significant sources of error in the uncertainty approximation. However, it has been shown in [29] and [30] that, for Smolyak grids, this is not an issue as convergence of the approximation is still guaranteed.

Fejér's "second" quadrature [31] is almost identical to the Clenshaw-Curtis quadrature but with one important difference; the endpoint quadrature nodes are weighted to zero, i.e., $f(-1) = 0$ and $f(+1) = 0$. The Fejér second quadrature rule is advantageous because it overcomes a feature of the KL-expansion in which we call "covariance extrapolation". Instead of strictly interpolating from the random input variables, the KL-expansion treats the data as samples–in space and random space–of a hypothetical true covariance function, then constructs a low-dimensional representation of this function, and re-samples from it. In this way the KL-expansion produces many more realizations than we originally had. Furthermore, these realizations can potentially have more extreme values than represented in the original data thus
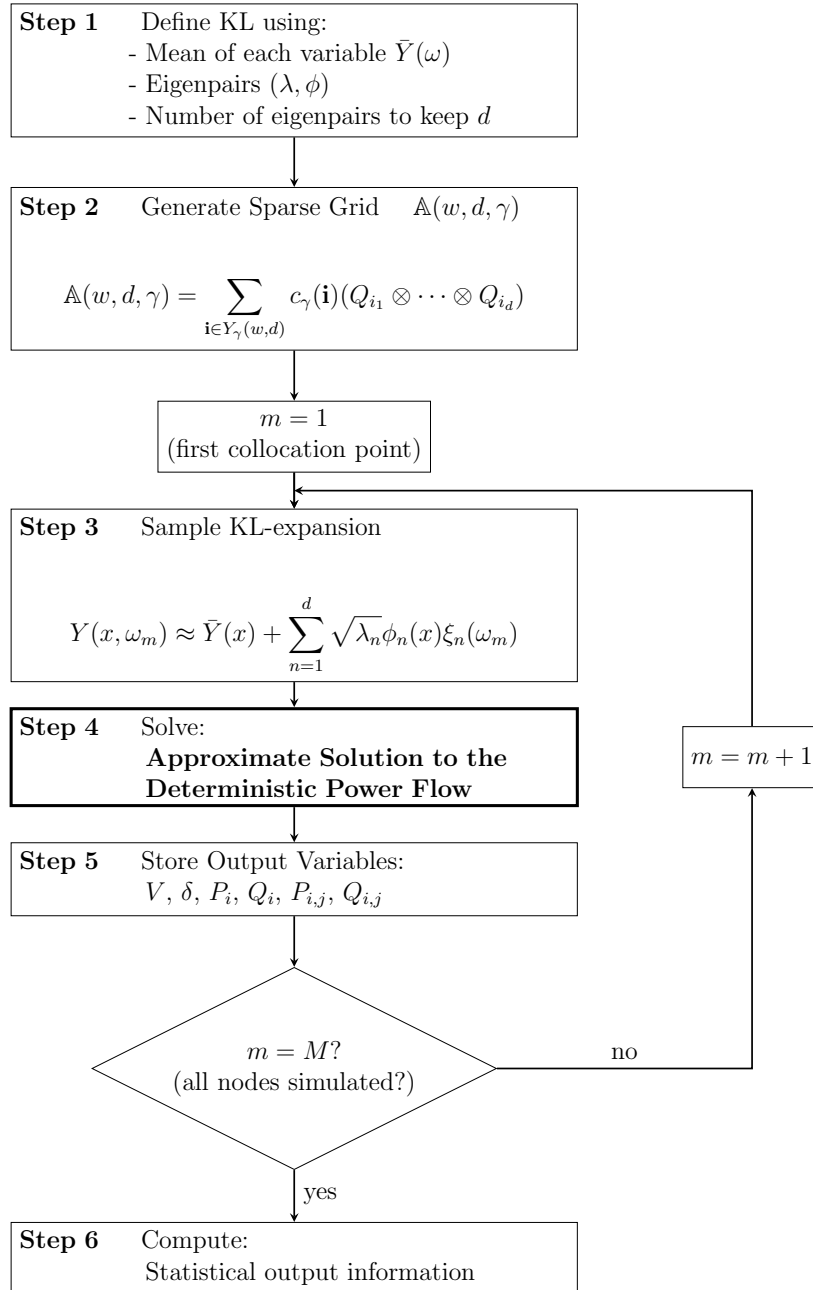
Figure 2: Flowchart for the proposed algorithm. See Section 4 for a joint description of each step of the methodology.

giving the appearance of extrapolating the values of the data. By not considering the most extreme samples, we lessen the chance of infeasible input realizations.

After generating the sparse grid in Fig. 2-Step 2 we then assemble the KL-expansion in (1) and sample the independent random variables ξ_n using the m-th collocation point. Subsequently, in Step 4 we solve the deterministic power flow. In Step 5, after each power flow simulation, we store bus voltages (V) and angles (δ), bus active (P_i) and reactive (Q_i) power injections, and branch active ($P_{i,j}$) and reactive ($Q_{i,j}$) flows. After saving all variables the algorithm checks for more collocation points to simulate, if there are, m is incremented and the process is repeated. Otherwise, we proceed to compute relevant statistical information on the output variables to complete the algorithm.

5. Experimental Setup

5.1. Software and Toolboxes for PLF and Sparse Grids

A MATLAB toolbox set for each of the sparse grids discussed in this paper were originally developed by J. Burkhardt [32], and we adapted them for use with power systems. In addition to the sparse grid toolboxes, we make use of the open-source power system toolbox MATPOWER [33] to calculate all deterministic load flow solutions. While there exist other commercially available power system analysis software the sharing of data between these toolboxes reduces computational overhead. All simulations were performed on a machine with Intel Core i7-7700HQ CPU 2.8 GHz PC and 16GB of RAM.

5.2. Metrics for Performance Comparisons

In addition to comparing computational burden for our proposed method we use the following equation for statistical accuracy comparisons:

$$\varepsilon_\mu = \left| \frac{\mu_{\text{MCS}} - \mu_{\text{grid}}}{\mu_{\text{MCS}}} \right| \qquad \varepsilon_\sigma = \left| \frac{\sigma_{\text{MCS}} - \sigma_{\text{grid}}}{\sigma_{\text{MCS}}} \right|, \qquad (18)$$

where μ_{MCS} and σ_{MCS} represent the mean and standard deviation of the Monte Carlo simulation method and will be henceforth used as reference values. Similarly, μ_{grid} and σ_{grid} represent the mean and standard deviation of various sparse grid methods. In addition to error analysis we make use of

12

the Kullback-Leibler divergence (KLD) measure [34] to ascertain the likeness of the output distributions of the sparse grid methods to the Monte Carlo method. The KLD measure is defined as follow

$$\mathrm{KLD}_f = \sum_x f_{\mathrm{MCS}}(x) \log \left(\frac{f_{\mathrm{MCS}}(x)}{f_{\mathrm{grid}}(x)} \right), \tag{19}$$

where $f_{\mathrm{MCS}}(x)$ and $f_{\mathrm{grid}}(x)$ are the output PDFs of the Monte Carlo method and the various grid methods, respectively. Lastly, the CDFs used for visualization are calculated by creating and sampling the sparse grid interpolant, which approximates (5), using the method in [35].

6. Experimental Results and Discussion

6.1. Test Cases for Proposed Methodology

In this paper, we test the proposed methodology on the IEEE 118-bus system [36] and on a modified version of the IEEE Reliability Test System–1996, the Reliability Test System–Grid Modernization Lab Consortium (RTS--GMLC) [37]. For the IEEE 118-bus system we will adopt the assumptions first listed in [2] whereby:

- **Generation units**. Each generation plant is divided into four units with the same power production and a forced outage rate of 0.09. A binomial distribution is used to model each generation plant, and the mean of this input random variable is set to the base case power production of the corresponding generation plant.

- **Load demand**. The active and reactive power of the load buses are modeled as normal distributions, whose means equal the base case data, and whose standard deviations are set arbitrarily as follows: 7% from bus #1 to bus #33, 4% from bus #34 to bus #59, 9% from bus #60 to bus #79, and 5% from bus #80 to bus #118.

In total, there are 194 random variables for this test case. Although the random variables considered in the test case are rudimentary, the intent of choosing them is to make comparisons between the classical method presented in [2] and our proposed method.

The second test case, the RTS-GMLC, is a product of the U.S. Department of Energy's initiative to modernize the grid. While much of the test

case remains the same as the original RTS, e.g., transmission line parameters, generator ratings, etc., researchers have mapped the synthetic case over the Southern California, Nevada, and Arizona region. This mapping enables the use of solar and wind resource maps to model the effect of photovoltaic (PV) solar, rooftop photovoltaic (RTPV) solar, concentrated solar power (CSP), wind, and hydro energy sources. A year worth of data for each type of renewable energy is quantized into discrete data points at 1-hour and 5-min intervals. In addition to these generation profiles, the RTS-GMLC dataset includes customer demand which is determined from energy usage data from the various utilities in the region. For this paper we will constrain the problem to using the PV, RTPV, and customer demand information; totaling 59 random variables.

6.2. Additional Assumptions and Implementation Specifics

For each test case we choose the number of d dimensions to keep for the truncated KL-expansion such that the percent total contribution of each eigenpair is greater than 90%. It should be noted that eigenpairs for each random input source are computed separately; that is, the covariance matrix is built using spatial information for each generator type (PV, RTPV, etc.) and load. For example, in the RTS-GMLC test case the covariance matrix for the PV data is a 25×25 matrix that explains the joint variability of each PV plant. In the case where the random input source is defined by a PDF and not data, the covariance matrix is built using 10,000 uniformly generated samples of the input source. This assumption leads us to keep 12 (out of 194) and 6 (out of 59) total dimensions for the IEEE 118-bus system and the RTS-GMLC test case, respectively.

Next, we choose the weight vector γ *a priori* using the recursive relationship $\gamma_1 = 1$; $\gamma_n = 2\gamma_{n-1}$ for $n \leq d$. Similar to the eigenpair calculation, this is done for each type of random input source. This choice means that the largest eigenpairs for each random input source will receive the maximum allowed sampling, denoted as l_{max}, and the proceeding eigenpairs will be less than or equal to l_{max}. For these experiments $l_{max} = 5$. The Monte Carlo samples used here for performance comparison are taken on the random variables ξ_n in the KL-expansion as compared to randomly sampling the random input variables. We consider representing the input uncertainty as a separate problem and are interested in how sampling the KL-expansion using quadrature nodes compares to random sampling.
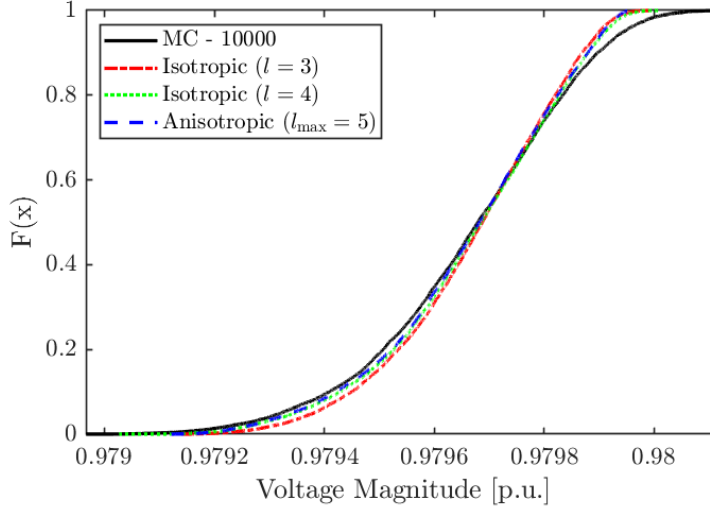
Figure 3: Voltage magnitude at Bus 93 in the IEEE 118-Bus system.

Lastly, since we are dealing with numerical methods it is important to mention the convergence and what would cause a power flow to not converge. In this paper we use a simplistic power flow method, which does not guarantee a global, or even local, optimum. Simply put, a simple power flow method tries to balance the generation with demand, how ever it can. Reactive power generator limits are sometimes optionally enforced in order to obtain a first approximation and then refine after reactive power compensation. Convergence (and the rate of convergence) of such methods highly depends on the system's initial conditions. In cases where there are large mismatches between generation and demand the solver may take several hundred iterations to converge, if at all. There does exist globally convergent methods, such as [38], which can overcome this problem all together.

6.3. Accuracy Comparisons

Figure 3 shows the output voltage at Bus 83 of the IEEE 118-bus system CDF for various levels of isotropic sparse grids, an anisotropic sparse grid with $l_{max} = 5$, and the reference Monte Carlo simulations which are performed using 10,000 random samples. We observe that the each sparse grid method accurately predicts the response variable with the $l = 3$ isotropic sparse grid performing slightly worse than both the $l = 4$ and the anisotropic sparse grid. This result is expected because as one increases the level of the

15

Table 1: Statistical accuracy metrics for the IEEE 118-Bus system

		V	δ	P_i	Q_i	$P_{i,j}$	$Q_{i,j}$
Isotropic ($l = 3$)	ε_μ [%]	0.01	0.05	0.03	0.09	0.11	0.04
	ε_σ [%]	4.16	0.71	12.0	5.22	6.78	5.28
	KLD_f	0.84	0.05	0.06	0.12	0.18	0.13
Isotropic ($l = 4$)	ε_μ [%]	0.01	0.05	0.03	0.09	0.11	0.04
	ε_σ [%]	4.16	0.71	12.0	5.22	6.78	5.28
	KLD_f	0.56	0.01	0.04	0.08	0.13	0.09
Anisotropic ($l_{\max} = 5$)	ε_μ [%]	0.01	0.05	0.03	0.09	0.11	0.04
	ε_σ [%]	4.70	0.68	13.0	6.00	7.71	5.90
	KLD_f	0.63	0.02	0.04	0.08	0.14	0.09

sparse grid the accuracy of the interpolant also increases.

Looking at the accuracy metrics listed in Table 1 we see that there is no difference between the two isotropic sparse grids. We suspect this is the result of the relatively low variance of the input sources. With respect to the KLD measure we should keep in mind that lower numbers indicate a better approximation of the true solution. The ideal approximation is one where the KLD measure is, in fact, zero. Here, by comparing the KLD measure between the two isotropic sparse grids we notice that we do indeed produce a more accurate statistical approximation with a higher level sparse grid. However, the big takeaway from this experiment, which will be further supported by the next section, is that the anisotropic sparse grid is able to maintain a comparable accuracy with far fewer collocation points. In this experiment, there is only 0.5-1% difference in accuracy.

Figure 4 shows the CDF for active and reactive power line flow from Bus 203 to Bus 209 of the RTS-GMLC test case. As we can see, each of the three sparse grid methods closely match the Monte Carlo simulations used as a reference. The differences between the two methods are even harder to see in Table 2. Theory tells us that as we increase the level of sparse grid, i.e., the number of collocation points, we should expect to see a higher degree of accuracy. This is not evident by looking the ε_μ and ε_σ measures alone. Upon inspecting the KLD measure does one see that the $l = 5$ is better than the $l = 4$ sparse grid. More importantly, the KLD measure shows us that the proposed anisotropic sparse grid method competes with the $l = 5$ isotropic sparse method while having roughly 2,000 fewer collocation points.
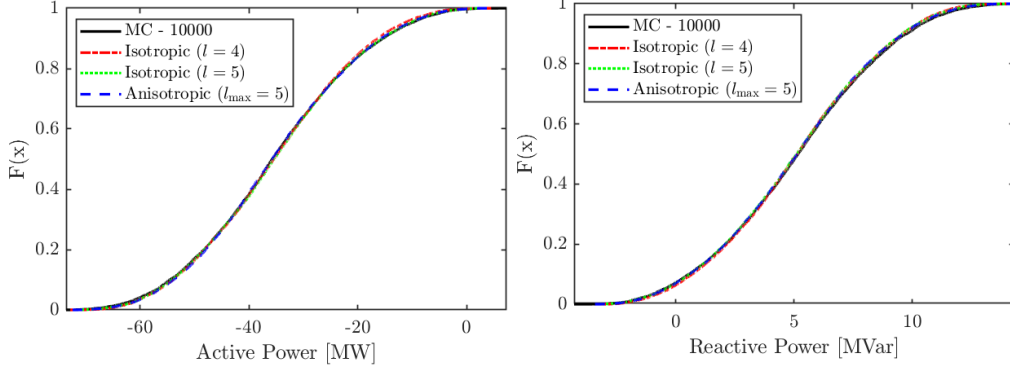
Figure 4: Reliability Test System–Grid Modernization Lab Consortium (RTS-GMLC) active and reactive power line flows for the branch between Bus 203 and Bus 209. Negative values indicate the direction of the flow, therefore, in this case active power is flowing predominantly from Bus 209 to Bus 203.

Table 2: Statistical accuracy metrics for RTS-GMLC

		V	δ	P_i	Q_i	$P_{i,j}$	$Q_{i,j}$
Isotropic $(l = 4)$	ε_μ [%]	1e-3	0.40	0.21	0.69	0.86	0.45
	ε_σ [%]	1.20	0.27	3.21	1.03	1.48	1.84
	KLD_f	0.38	5e-3	0.03	3e-3	8e-3	0.01
Isotropic $(l = 5)$	ε_μ [%]	1e-3	0.40	0.21	0.69	0.86	0.45
	ε_σ [%]	1.20	0.27	3.21	1.03	1.48	1.84
	KLD_f	0.19	3e-3	0.02	3e-3	6e-3	7e-3
Anisotropic $(l_{\max} = 5)$	ε_μ [%]	1e-3	0.40	0.21	0.69	0.86	0.45
	ε_σ [%]	1.20	0.27	3.21	1.06	1.48	1.91
	KLD_f	0.44	3e-3	0.02	3e-3	6e-3	7e-3

6.4. Computational Burden Comparisons

Computational burden comparisons for the IEEE 118-bus system are found in Table 3. To see which portion of the method takes the longest to perform we have broken up the total time spent into four segments: eigen-pair calculation, sparse grid construction, KL-expansion, and deterministic load flow solutions. Again, for comparison we will use 10,000 Monte Carlo simulations which, in this case, take 34.3 s to complete. The first thing to notice is the total number of samples–or collocation points for the sparse grids–required per method. We observe that as the number of points, i.e.,

Table 3: Computational comparison for the anisotropic sparse grid method for the PLF problem on the IEEE 118-bus system

	MC-10000	Isotropic Sparse Grid ($l=3$)	Isotropic Sparse Grid ($l=4$)	Anisotropic Sparse Grid ($l_{\max}=5$)
N_{samples}[†]	10,000	337	3,249	213
$t_{\text{eigenpairs}}$	0.01 s	0.01 s	0.01 s	0.01 s
t_{grid}	-	0.08 s	0.75 s	0.19 s
t_{KL}	1.10 s	0.01 s	0.04 s	0.01 s
$t_{\text{P.F.}}$	34.3 s	1.50 s	11.8 s	0.88 s
Total time	**35.4 s**	**1.60 s**	**12.6 s**	**1.09 s**

[†] Out of the original 194 random variables 12 are kept after dimension reduction

Table 4: Computational comparison for the anisotropic sparse grid method for the PLF problem on the RTS-GMLC system

	MC-10000	Isotropic Sparse Grid ($l=4$)	Isotropic Sparse Grid ($l=5$)	Anisotropic Sparse Grid ($l_{\max}=5$)
N_{samples}[†]	10,000	545	2,561	489
$t_{\text{eigenpairs}}$	0.03 s	0.03 s	0.03 s	0.03 s
t_{grid}	-	0.10 s	0.57 s	0.15 s
t_{KL}	0.56 s	0.01 s	0.03 s	0.02 s
$t_{\text{P.F.}}$	30.3 s	1.92 s	8.15 s	1.81 s
Total time	**30.9 s**	**2.06 s**	**8.78 s**	**2.01 s**

[†] Out of the original 59 random variables 6 are kept after dimension reduction

level, increases, the total computational burden increases as well. This is a factor of it both taking longer to generate the sparse grid points (as shown in t_{grid}) as well as the time it takes to perform each deterministic load flow. By comparing each method we see that the anisotropic sparse grid interpolant is roughly 1.7x faster than the next fastest sparse grid method. As we have mentioned earlier, this significant decrease in computational time only costs 0.5 - 1% decrease in accuracy. We see a similar result for the RTS-GMLC test case in Table 4. In this case, the anistropic sparse grid is only 0.05 seconds faster than the $l = 4$ isotropic sparse grid but, again, provides the same level of accuracy as the $l = 5$ isotropic sparse grid.

In Figure 5, we show how the proposed method scales, both in accuracy and computational time, as a function of dimension importance. In Section 6.2 we establish that the weight vector used in these simulations is calculated
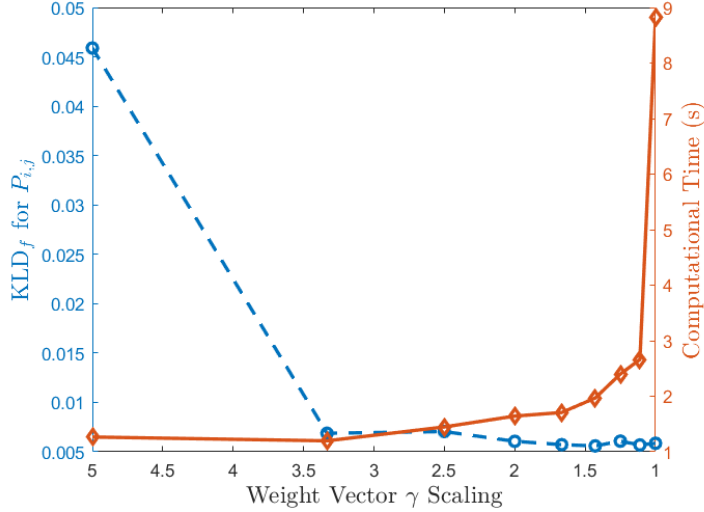
Figure 5: Comparison between computational time and accuracy, both as a function of the anisotropic weight vector γ. Graphs are generated using the RTS-GMLC test case and setup as described in Section 6.1.

using a recursive relationship. To generate the plots in Fig. 5 we modify the equation by adding a scalar ζ such that $\gamma_n = \zeta \gamma_{n-1}$ for $n \leq d$. In this equation, when ζ is large the dimensions after the first are treated with less importance. When $\zeta = 1$ each dimension is treated with the same importance and we create an fully isotropic sparse grid. This plot shows that as we move towards an isotropic sparse grid (increased collocation points) the estimator becomes more accurate. The more points simulated in each dimension, the better the uncertainty can be represented. This, however, comes at an exponential increase in the computation time. This plot goes to show that not only are the number of dimensions important to consider but also the number of collocation points as well.

6.5. Methods to Increase Performance

One very useful feature of the proposed method is that there are several ways to tune the method for either increases in accuracy and/or decreases in computational burden. The first way is to change the number of dimensions to keep in the KL-expansion. Reducing the number of eigenpairs to keep will significantly decrease the number of total collocation points need to be created, and thereby decreasing computational burden. In most cases, the first eigenpair $\{\lambda_1, \phi_1\}$ accounts for greater than 80% of the total variance in

19

the system which leads us to question how important the proceeding eigen-pairs are to the accuracy of the system. The second way is parallelizing the deterministic load flow solutions. Depending on the number of cores available and the access to, for example, the Parallel Computing Toolbox^{TM} in MATLAB, parallelizing this segment of the code requires changing one line of code. Third, increasing or decreasing l_{max} changes the maximum allowed level for the anisotropic sparse grid. Increasing l_{max} means having more collocation nodes for each dimension, i.e., a higher level of accuracy, but at the cost of increased computational burden. The opposite is true for a decrease in l_{max}. Lastly, changing the weight vector γ will have a similar effect as changing l_{max}. One way to increase the performance is to weigh higher dimension by their normalized eigenvalues. This will have the outcome to increase/decrease the number of KL-expansion terms to keep and will impact the accuracy and/or computational burden.

7. Conclusion

In this paper, we have established a formal methodology for implementing the coupled KL-expansion and anisotropic sparse grid to solve the PLF problem. Using two test cases, the IEEE 118-bus system and the more modernized grid RTS-GMLC, we have shown that our methodology is able to alleviate the "curse of dimensionality" issue present in the PLF problem. Furthermore, our method can solve the high-dimensional load flow problem 30 times faster while only having a 0.05% and 6% error in the mean and standard deviation when compared to 10,000 Monte Carlo simulations. In the modernized RTS-GMLC we show that, by taking advantage of the correlated input data, we are able to accurately and efficiently approximate their effect on the output variables. In addition to the observed high performance, we discuss how to tune the method to further increase accuracy and/or decrease computational burden, the easiest of which to implement is the parallelization of the deterministic load flow problem. Depending on the infrastructure available, this change could enable this method to be used as a real-time assessment of uncertainty given a large number of random variables.

References

[1] B. Borkowska, Probabilistic Load Flow, IEEE Trans. Power Appar. Syst. PAS-93 (1974) 1–6. doi:10.1109/TPAS.1974.293973.

[2] J. M. Morales, J. Pérez-Ruiz, Point estimate schemes to solve the probabilistic power flow, IEEE Trans. Power Syst. 22 (2007) 1594–1601. doi:10.1109/TPWRS.2007.907515.

[3] P. Chen, Z. Chen, B. Bak-Jensen, Probabilistic load flow: A review, 3rd Int. Conf. Deregul. Restruct. Power Technol. DRPT 2008 (2008) 1586–1591. doi:10.1109/DRPT.2008.4523658.

[4] R. N. Allan, A. M. L. da Silva, R. C. Burchett, Evaluation methods and accuracy in probabilistic load flow solutions, IEEE Trans. Power Appar. Syst. PAS-100 (1981) 2539–2546. doi:10.1109/TPAS.1981.316721.

[5] M. L. Kloubert, C. Rehtanz, Enhancement to the combination of point estimate method and Gram-Charlier Expansion method for probabilistic load flow computations, IEEE Manchester PowerTech (2017). doi:10.1109/PTC.2017.7980901.

[6] P. Zhang, S. T. Lee, Probabilistic Load Flow Computation Using the Method of Combined Cumulants and Gram-Charlier Expansion, IEEE Trans. Power Syst. 19 (2004) 676–682. doi:10.1109/TPWRS.2003.818743.

[7] M. Fan, S. Member, V. Vittal, G. T. Heydt, L. Fellow, Probabilistic Power Flow Studies for Transmission Systems With Photovoltaic Generation Using Cumulants, IEEE Trans. Power Syst. 27 (2012) 2251–2261. doi:10.1109/TPWRS.2012.2190533.

[8] R. R. Appino, T. Mühlpfordt, T. Faulwasser, V. Hagenmeyer, On Solving Probabilistic Load Flow for Radial Grids using Polynomial Chaos, in: 2017 IEEE Manchester PowerTech, 2017, pp. 1–6.

[9] H. Wu, Y. Zhou, S. Dong, Y. Song, Probabilistic Load Flow Based on Generalized Polynomial Chaos, IEEE Trans. Power Syst. 32 (2017) 820–821.

[10] X. Sun, Q. Tu, J. Chen, C. Zhang, X. Duan, Probabilistic load flow calculation based on sparse polynomial chaos expansion, IET Gener. Transm. Distrib. 12 (2018) 2735–2744. doi:10.1049/iet-gtd.2017.0859.

[11] Y. Zhou, H. Wu, C. Gu, Y. Song, A Novel Method of Polynomial Approximation for Parametric Problems in Power Systems, IEEE Trans. Power Syst. 32 (2017) 3298–3307. doi:`10.1109/TPWRS.2016.2623820`.

[12] E. Rosenblueth, Two-point estimates in probabilities, Appl. Math. Model. 5 (1981) 329–335. doi:`10.1016/S0307-904X(81)80054-6`.

[13] M. E. Harr, Probabilistic estimates for multivariate analyses, Appl. Math. Model. 13 (1989) 313–318. doi:`10.1016/0307-904X(89)90075-9`.

[14] K. S. Li, Point Estimate Method for Calculating Statistical Moments, J. Eng. Mech. 118 (1992) 1506–1511. doi:`10.1061/(ASCE)0733-9399(1992)118:7(1506)`.

[15] H. Hong, An efficient point estimate method for probabilistic analysis, Reliab. Eng. Syst. Saf. 59 (1998) 261–267. doi:`10.1016/S0951-8320(97)00071-9`.

[16] D. Xiu, Fast numerical methods for stochastic computations: A review, in: Commun. Comput. Phys., 2009, pp. 242–272. doi:`10.1.1.148.5499`. `arXiv:arXiv:1011.1669v3`.

[17] D. Xiu, Numerical Methods for Stochastic Computations : A Spectral Method Approach, Princeton University Press, 2010.

[18] S. A. Smolyak, Quadrature and interpolation formulas for tensor products of certain classes of functions, Doklady Akademii Nauk 148 (1963) 1042–1045.

[19] J. Tang, F. Ni, F. Ponci, A. Monti, Dimension-Adaptive Sparse Grid Interpolation for Uncertainty Quantification in Modern Power Systems: Probabilistic Power Flow, IEEE Trans. Power Syst. 31 (2016) 907–919. doi:`10.1109/TPWRS.2015.2404841`.

[20] F. Ni, P. H. Nguyen, J. F. G. Cobben, Basis-adaptive sparse polynomial chaos expansion for probabilistic power flow, IEEE Transactions on Power Systems 32 (2017) 694–704. doi:`10.1109/TPWRS.2016.2558622`.

[21] D. D. Kosambi, Statistics in function space, The Journal of the Indian Mathematical Society 7 (1943) 76–88.

[22] K. Karhunen, On linear methods in probability and statistics, Ann. Acad. Sci. Fenn. Ser. AI Math.-Phys (1947) 1—-79.

[23] M. Loève, Probability Theory II, volume 46 of *Graduate Texts in Mathematics*, Springer New York, New York, NY, 1978. doi:`10.1007/978-1-4684-9464-8`.

[24] F. Nobile, R. Tempone, C. G. Webster, An Anisotropic Sparse Grid Stochastic Collocation Method for Partial Differential Equations with Random Input Data, SIAM J. Numer. Anal. 46 (2008) 2411–2442. doi:`10.1137/070680540`. `arXiv:1404.2647`.

[25] M. Griebel, T. Grestner, Numerical Integration using Sparse-grids, Numer. Algorithms 18 (1998) 209–232.

[26] T. Gerstner, M. Griebel, Dimension – Adaptive Tensor – Product Quadrature, Computing 71 (2003) 65–87. doi:`10.1007/s00607-003-0015-5`.

[27] J. Burkardt, The "Combining Coefficient" for Anisotropic Sparse Grids, Interdiscip. Cent. Appl. Math. & Inf. Technol. Dep. Virginia Tech (2012) 1–14. URL: `https://people.sc.fsu.edu/~jburkardt/presentations/sgmga_coefficient.pdf`.

[28] C. W. Clenshaw, A. R. Curtis, A method for numerical integration on an automatic computer, Numerische Mathematik 2 (1960) 197–205.

[29] E. Novak, K. Ritter, The curse of dimension and a universal method for numerical integration, in: Multivariate approximation and splines, Springer, 1997, pp. 177–187.

[30] T. Gerstner, M. Griebel, Numerical integration using sparse grids, Numerical algorithms 18 (1998) 209.

[31] L. Fejér, Mechanische quadraturen mit positiven cotesschen zahlen, Mathematische Zeitschrift 37 (1933) 287–309.

[32] J. Burkardt, John Burkhardt's Home Page, 2019. URL: `https://people.sc.fsu.edu/~jburkardt/`.

[33] R. D. Zimmerman, C. E. Murilla-Sanchez, R. J. Thomas, MATPOWER: Steady-State Operations, Planning and Analysis Tools for Power Systems Research and Education, IEEE Trans. Power Syst. 26 (2011) 12–19.

[34] S. Kullback, R. A. Leibler, On information and sufficiency, The annals of mathematical statistics 22 (1951) 79–86.

[35] K. L. Judd, L. Maliar, S. Maliar, R. Valero, Smolyak method for solving dynamic economic models: Lagrange interpolation, anisotropic grid and adaptive domain, J. Econ. Dyn. Control 44 (2014) 92–123. doi:`10.1016/j.jedc.2014.03.003`.

[36] R. Christie, Power Systems Test Case Archive, 2000. URL: `http://labs.ece.uw.edu/pstca/`.

[37] C. Barrows, A. Bloom, A. Ehlen, J. Ikaheimo, J. Jorgenson, D. Krishnamurthy, J. Lau, B. McBennett, M. O'Connell, E. Preston, A. Staid, G. Stephen, J. Watson, The ieee reliability test system: A proposed 2019 update, IEEE Transactions on Power Systems (2019) 1–1. doi:`10.1109/TPWRS.2019.2925557`.

[38] B. Ghaddar, J. Marecek, M. Mevissen, Optimal power flow as a polynomial optimization problem, IEEE Transactions on Power Systems 31 (2016) 539–546.