School of Computation, Information and Technology

Technische Universität München

Master's Thesis in Computational Science and Engineering

Efficient Uncertainty Quantification for Power Networks

Thiago Lima Carneiro

School of Computation, Information and Technology

Technische Universität München

Master's Thesis in Computational Science and Engineering

Efficient Uncertainty Quantification for Power Networks

Author:	Thiago Lima Carneiro
1st examiner:	Prof. Hans-Joachim Bungatz
Assistant advisor:	M.Sc. Ravi Kislaya
1st advisors:	M.Sc. Christoph Ludwig; Dr. Dimitrios Loukrezis
Submission Date:	July 15th, 2024

I hereby declare that this thesis is entirely the result of my own work except where otherwise indicated. I have only used the resources given in the list of references.

July 15th, 2024 Thiago Lima Carneiro

Acknowledgments

I am grateful to Prof. Hans-Joachim Bungatz for his examination and the opportunity to work on this thesis at the Chair of Scientific Computing.
Moreover, this thesis would not have been possible without the invaluable support, patience, and guidance of my academic advisor – Ravi Kyslaya. He introduced me to a novel approach to *Sensitivity Analysis* and supported me throughout this work.
My profound thanks go to Christoph Ludwig and Dimitrios Loukrezis from Siemens Technology for providing the topic of this thesis and allowing me to work with their team. Their decisive leadership, assistance, and belief in my work were key to my success in this endeavor.
I thank my family for their support throughout my life. Their unwavering assistance has been a constant source of strength and motivation, without which this thesis would not have happened.

“Making the right use of a book doesn’t result in finding evil, but rather, as an appealing saying puts it, evidence of our hearts.”

-Jacob Grimm

Abstract

The emergence of renewable energy sources (RES) introduces new gateways of uncertainty within power grid networks, necessitating robust and efficient probabilistic models to account for the inherent variability. This variability stems from renewable energy generation and various other factors, including load fluctuations and technical failures. Developing and applying such probabilistic methods is crucial for ensuring that policymakers can make informed decisions based on reliable information, thereby enhancing the stability and resilience of the energy sector.

In the first part of this work, we present the basic theory of Uncertainty Propagation and Monte Carlo sampling, followed by the theory of Polynomial Chaos Expansion (PCE) and Sensitivity Analysis (ANOVA, Saltelli, and Rank-based Estimation). Examples and allusions of how we can apply the theories to probabilistic power flow analysis (PPF) are given throughout the sessions using representations of the PF system. In the second part of this thesis, we present a case study using the European high-voltage transmission network *1354pegase*, in which we analyze scenarios with different numbers of input random variables (RVs), ranging from 100 to 621 RVs. We then propose an efficient pipeline, in which PCE is combined with clustering techniques, and a new sensitivity analysis approach, with rank-based estimation, to reduce the size of the stochastic input space. Finally, we present the results and comparisons of the case analyses in the third and last part.

Index Terms: **Probabilistic Power Flow, Uncertainty Quantification, Surrogate Modeling, Polynomial Chaos Expansion, Sensitivity Analysis, Clustering.**

Contents

Part I.

Introduction and Background Theory

1. Introduction

In modern power systems, the ever-increasing integration of renewable energy sources [5] introduces new gateways of power fluctuations and poses significant challenges to traditional deterministic power flow analysis methods [5], [44]. Addressing these challenges demands a paradigm shift towards more advanced techniques to capture these systems' inherent uncertainties and complexities [1].

This thesis explores the intersection of two promising fields: Probabilistic Power Flow (PPF) Analysis and Surrogate Modeling. PPF amplifies traditional power flow analysis by considering probabilistic models and accounts for uncertainties arising from renewable energy generation, load fluctuations, and network incidents [69]. We perform traditional PPF analysis by applying Monte Carlo (MC) methods, and although accurate, they are computationally expensive [18], [27]. In contrast, surrogate models, also known as meta-models, offer computationally efficient alternatives to intensive MC simulations by approximating the behavior of intricate stochastic systems using simpler models.

1.1. Motivation

Power grids are the primary means of distributing electrical energy in modern society and countless equipments and activities require electricity to operate: factories, household appliances, computers, etc. This fundamental role of electricity requires a robust and reliable power grid infrastructure so that the demand for electrical energy is met in a constant and efficient manner. However, the advent of alternative sources for generating electrical energy has gradually increased the fluctuation levels in power generation and, consequently, the instability in the electrical distribution systems. The need to monitor and optimize electrical networks becomes critical to ensure acceptable security of the distribution systems. One of the crucial monitoring steps is the power flow analysis of a given network, in which we calculate the active and reactive powers, and the voltage magnitudes of each bus in a power grid.

A PPF analysis is necessary in order to measure how fluctuations in the power grid affect the distribution system as a whole. MC simulations are one of the most widely used means for this purpose, however, these simulations are computationally intensive. One of the ways to calculate the PPF analysis more efficiently is through probabilistic models such as Polynomial Chaos Expansion (PCE) [43], [32] and Gaussian Process. To create more efficient models for the PPF analysis, we explore in this work surrogate modeling with PCE, combined with *clustering techniques* [31], and a novel *rank-based sensitivity analysis* method [21], to reduce the stochastic space. With this combination of techniques, we can bring promising results in terms of efficiency in the PPF analysis.

Ultimately, the findings of this thesis aim to contribute to the broader academic community, industry practitioners, and policymakers, facilitating the adoption of advanced analysis techniques for enhancing the resilience, reliability, and sustainability of modern power systems.

1.2. Related Work

Several studies that present probabilistic models to solve the PPF analysis range from sample generation using Latin Hypercubes and Cholesky Decomposition [70] to the use of probabilistic models with Gaussian Process [47] and Information Gap Decision Theory (IGDT)[50], which addresses the problem of optimal power flow (OPF) applied to offshore wind farms and validates the application of the IGDT-based OPF model for the optimal operation of AC/DC power systems with high penetration of wind farms.

In addition, studies that investigate and propose applying PCE exploring adaptive sparsity schemes have been presented in [46]. This paper introduces the Basis-Adaptive Sparse Polynomial Chaos (BASPC) methodology for calculating the PPF analysis. BASPC relies on three state-of-the-art uncertainty quantification methodologies: the hyperbolic scheme to truncate the infinite polynomial chaos (PC) series, the Least Angle Regression (LARS) method to select the optimal degree of each univariate PC series, and the Copula method to address nonlinear correlations among input random variables.

Moreover, PCE has been extensively used as an efficient means of performing PPF analysis. In [41], PCE is combined with clustering techniques that reduce the stochastic space, increasing computation efficiency with PCE.

Other works explore the computation of PPF analysis through Hammersley-importance sampling and eigen-decomposition [36] and also through Copula and Latin hypercube sampling [13].

2. Power Flow Analysis

In the context of graph theory, an AC power grid can be represented as a graph G where N denotes the buses as vertices, and L denotes the transmission lines as edges, thus $G = (N, L)$, with the bus index set $\mathcal{N} = \{1, 2, ..., N\}$. The power grid network has a complex bus impedance matrix $Y \in \mathbb{C}^{N \times N}$ with entries represented by $Y_{ij} = G_{ij} + \jmath B_{ij} \in \mathbb{C}$ for all $i, j \in \mathcal{N}$ and $\jmath$ the imaginary unit.

2.1. Deterministic Power Flow

The deterministic power flow equations can be mathematically formulated, given the power phasor $\vec{s} \in \mathbb{C}^N$ and the voltage phasor $\vec{v} \in \mathbb{C}^N$ using two significant coordinate systems based on the voltage phasor representation: polar coordinates and rectangular coordinates [43]:

$$\text{polar: } \vec{v}_i = v_i \mathrm{e}^{\jmath \delta_i}, \vec{s}_i = p_i + \jmath q_i, \tag{2.1}$$

$$\text{rectangular: } \vec{v}_i = v_i^{\text{re}} + \jmath v_i^{\text{im}}, \vec{s}_i = p_i + \jmath q_i, \tag{2.2}$$

where p_i is the active power, q_i is the reactive power, and δ_i is the phase angle for all $i \in \mathcal{N}$. The deterministic *polar* power flow equations are defined $\forall i \in \mathcal{N}$ as:

$$x_i^p = \begin{bmatrix} v_i, & \delta_i, & p_i, & q_i \end{bmatrix}^\top \in \mathbb{R}^4, \tag{2.3}$$

$$p_i = \sum_{j \in \mathcal{N}} v_i v_j \left(G_{ij} \cos\left(\delta_i - \delta_j\right) + B_{ij} \sin\left(\delta_i - \delta_j\right) \right), \tag{2.4}$$

$$q_i = \sum_{j \in \mathcal{N}} v_i v_j \left(G_{ij} \sin\left(\delta_i - \delta_j\right) - B_{ij} \cos\left(\delta_i - \delta_j\right) \right), \tag{2.5}$$

with variables $\mathbf{x}^p = \left[x_1^{p\top}, \ldots, x_N^{p\top} \right]^\top \in \mathbb{R}^{4N}$ [43].

In a similar way, we define the deterministic *rectangular* power flow problem as:

$$x_i = \begin{bmatrix} v_i^{\text{re}}, & v_i^{\text{im}}, & p_i, & q_i, \end{bmatrix}^\top \in \mathbb{R}^4, \tag{2.6}$$

$$p_i = \sum_{j \in \mathcal{N}} G_{ij} \left(v_i^{\text{re}} v_j^{\text{re}} + v_i^{\text{im}} v_j^{\text{im}} \right) + B_{ij} \left(v_i^{\text{im}} v_j^{\text{im}} - v_i^{\text{re}} v_j^{\text{im}} \right), \tag{2.7}$$

$$q_i = \sum_{j \in \mathcal{N}} G_{ij} \left(v_i^{\mathrm{im}} v_j^{\mathrm{im}} - v_i^{\mathrm{re}} v_j^{\mathrm{im}} \right) - B_{ij} \left(v_i^{\mathrm{re}} v_j^{\mathrm{re}} + v_i^{\mathrm{im}} v_j^{\mathrm{im}} \right), \tag{2.8}$$

with variables $\mathbf{x} = \left[x_1^\top, \ldots, x_N^\top \right]^\top \in \mathbb{R}^{4N}$. The *polar* and *rectangular* power flow problems need specifications that define variables at each bus $i \in \mathcal{N}$, such that the problems involve $2N$ unknowns and $2N$ equations, becoming a well-defined system of equations. The most common bus specifications are *Slack buses* (a slack bus is used to balance the active power and reactive power), *Active / Reactive Power (PQ)*, *Active Power / Voltage (PV)*:

- *Polar* power flow specifications:

$$\text{Slack buses: } \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} x_i^p \overset{!}{=} \begin{bmatrix} v^{\mathrm{ref}} \\ \delta^{\mathrm{ref}} \end{bmatrix}, \tag{2.9}$$

$$\text{Actice / Reactive Power (PQ): } \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} x_i^p \overset{!}{=} \begin{bmatrix} p_i^{\mathrm{ref}}, \\ q_i^{\mathrm{ref}} \end{bmatrix}, \tag{2.10}$$

$$\text{Active Power / Voltage (PV): } \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} x_i^p \overset{!}{=} \begin{bmatrix} p_i^{\mathrm{ref}} \\ v_i^{\mathrm{ref}} \end{bmatrix}. \tag{2.11}$$

- *Rectangular* power flow specifications:

$$\text{Slack buses: } \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} x_i \overset{!}{=} \begin{bmatrix} v^{\mathrm{ref}} \\ 0 \end{bmatrix}, \tag{2.12}$$

$$\text{Actice / Reactive Power (PQ): } \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} x_i \overset{!}{=} \begin{bmatrix} p_i^{\mathrm{ref}} \\ q_i^{\mathrm{ref}} \end{bmatrix}, \tag{2.13}$$

$$\text{Active Power / Voltage (PV): } \begin{bmatrix} [0 \quad 0 \quad 1 \quad 0] x_i \\ (v_i^{\mathrm{re}})^2 + (v_i^{\mathrm{im}})^2 \end{bmatrix} \overset{!}{=} \begin{bmatrix} p_i^{\mathrm{ref}} \\ \left(v_i^{\mathrm{ref}} \right)^2 \end{bmatrix}. \tag{2.14}$$

The rectangular power flow formulation exhibits quadratic (polynomial) non-linearities in its variables, contrasting the trigonometric non-linearities inherent in the polar power flow formulation. This fundamental distinction renders the rectangular formulation advantageous for specific advanced analytical techniques. Specifically, the polynomial nature of the non-linearities in the rectangular power flow allows for more effective exploitation in PPF analysis and the application of Polynomial Chaos Expansion (PCE). The quadratic non-linearities facilitate the mathematical treatment of uncertainties within the power system, enhancing the ability to model and analyze the impacts of variability in system parameters, such as load fluctuations and renewable generation outputs. Consequently, the rectangular power flow formulation is a powerful tool for probabilistic studies and reliability assessments in power systems engineering.

2.2. Probabilistic Power Flow (PPF)

The PPF occurs when the power flow specification values in Equations [2.9 - 2.14] are defined using random variables (RVs) with distributions characterized with a probability density function (PDF). This approach allows for incorporating uncertainties in the input parameters, such as load demands and generation outputs, by modeling them probabilistically rather than deterministically. PPF analysis can more accurately reflect modern power systems' inherent variability and unpredictability by representing these input variables as RVs.

Due to their robustness and accuracy, Monte Carlo simulations are often used to solve the probabilistic power flow (PPF) problem [18], [27]. However, these simulations are computationally expensive and time-consuming, particularly for large-scale systems. To achieve reasonable accuracy of results, the number of samples to use Monte Carlo can become cumbersome or even impractical.

Additionally, when using surrogate modeling, the dimension of the RVs can be excessively high, making the computational load impractical. To address this, techniques such as clustering with *K-Means* [31], [10], [12], [37] can be employed to reduce the dimensionality of the RV space, especially in cases where the RVs are known to be correlated or influenced by factors such as geographical area [41].

In this context, Polynomial Chaos Expansion (PCE) becomes valuable when combined with clustering techniques and sensitivity analysis to reduce the RV space and perform PPF analysis. With this approach, we can obtain computationally efficient alternatives to Monte Carlo simulations. This procedure enables faster yet reliable solutions by approximating the stochastic behavior of power systems. By leveraging clustering techniques [41] and sensitivity analysis, we can reduce the problem's stochastic dimensionality, and by using PCE, we lead to significant improvements in computational efficiency without compromising the accuracy of the results, as we are going to show later in this work.

3. Uncertainty Propagation

Numerous phenomena and mechanisms in nature, such as the structure of solid materials and molecular dynamics, are described and modeled by physical laws developed through rigorous scientific work and observations aimed at elucidating the behavior of physical systems and, ultimately, the dynamics of nature. In this environment, mathematical modeling plays a crucial role in analyzing and studying these real-world models, enabling predicting and comprehending various events recurrent in science and engineering. In this context, numerous events are characterized by randomness and underlying uncertainties, which influence the performance and outcomes of engineering and natural systems. In the specific case examined in this work, we will explore how to apply forward uncertainty quantification in power flow analysis.

3.1. Random Variables and Statistics of Interest (SoI)

Our goal is to quantify uncertainties in complex, real-world systems that are governed by a model $\mathcal{F}$ characterized by a (usually non-linear) system of equations that lack known analytical solution and only have numerical solutions, e.g., the power flow non-linear system of equations in chapter 2. The model $\mathcal{F}$ depends on deterministic inputs $\boldsymbol{x}$, and stochastic inputs $\boldsymbol{\theta}$, i.e. $\mathcal{F} \equiv \mathcal{F}(\boldsymbol{x}, \boldsymbol{\theta})$. Let $n_{\text{det}} \in \mathbb{N}$ be the dimension of the deterministic space and $n_{\boldsymbol{\theta}} \in \mathbb{N}$ represent the dimension of the stochastic input space. Our numerical model $\mathcal{F}$ specifying the real-world complex phenomena (PPF in our case) is characterized as $\mathcal{F} : \mathbb{R}^{n_{\text{det}}} \times \mathbb{R}^{n_{\boldsymbol{\theta}}} \to \mathbb{R}^{n_{\boldsymbol{Y}}}$, i.e., a mapping from a n_{det}-dimensional deterministic input $\boldsymbol{x} \in \mathbb{R}^{n_{\text{det}}}$, and $n_{\boldsymbol{\theta}}$-dimensional stochastic input $\boldsymbol{\theta} \in \mathbb{R}^{n_{\boldsymbol{\theta}}}$, to our *quantity of interest* (QoI) $\boldsymbol{Y} \equiv \mathcal{F}(\boldsymbol{x}, \boldsymbol{\theta}) \in \mathbb{R}^{n_{\boldsymbol{Y}}}$, with dimension of the QoI stochastic space $n_{\boldsymbol{Y}} \in \mathbb{N}$.

We can now introduce the *multivariate real-valued random variable* $\boldsymbol{\theta}$, which is defined in the complete probability space $(\Omega_{\boldsymbol{\theta}}, A_{\boldsymbol{\theta}}, P_{\boldsymbol{\theta}})$, where $\Omega_{\boldsymbol{\theta}}$ is the sample space, $A_{\boldsymbol{\theta}}$ is the event space with σ-field on $\Omega_{\boldsymbol{\theta}}$, and $\mathrm{P}_{\boldsymbol{\theta}} : A_{\boldsymbol{\theta}} \to [0, 1]$ is the probability measure [33]. Moreover, we consider $\mathcal{B}^{n_{\boldsymbol{\theta}}}$ the representation of the Borel σ-field on $\mathbb{R}^{n_{\boldsymbol{\theta}}}$, and $\boldsymbol{\theta} : (\Omega_{\boldsymbol{\theta}}, A_{\boldsymbol{\theta}}) \to (\mathbb{R}^{n_{\boldsymbol{\theta}}}, \mathcal{B}^{n_{\boldsymbol{\theta}}})$ is a *continuous random vector* that we further simplify the notation $\boldsymbol{\theta} = \boldsymbol{\theta}(\boldsymbol{\omega})$ or $\theta_i = \theta(\omega_i)$ for each component $\{\omega_i\}_{i=1}^{n_{\boldsymbol{\theta}}}$. $\boldsymbol{\theta}$ is characterized by the probability density function (PDF) $\boldsymbol{\rho_{\theta}} : \mathbb{R}^{n_{\boldsymbol{\theta}}} \to \mathbb{R}_0^+$ (with $\rho_{\theta_i} : \mathbb{R} \to \mathbb{R}_0^+$ for each component $i = 1, \cdots, n_{\boldsymbol{\theta}}$). $\{\rho_{\theta_i}\}_{i=0}^{n_{\boldsymbol{\theta}}}$ are in the finite-dimensional second-order random space L^2, i.e., they have finite second statistical moment value as follows:

$$\mu_2[\theta_i] := \int_{\Omega_{\theta_i}} (\theta_i - \mathbb{E}[\theta_i])^2 \rho_{\theta_i}(\theta_i) d\theta_i \text{ for } i = 1, \cdots, n_{\boldsymbol{\theta}}. \tag{3.1}$$

The expectation value and variance are the first and second statistical moments, respectively:

$$\begin{aligned}\mu_1[\theta_i] &:= \mathbb{E}\left[\theta_i\right] = \int_{\Omega_{\theta_i}} \theta_i \rho_{\theta_i}\left(\theta_i\right) d\theta_i,\\ \mu_2[\theta_i] &:= \mathbb{V}\left[\theta_i\right] = \mathbb{E}\left[\theta_i^2\right] - \mathbb{E}\left[\theta_i\right]^2 .\end{aligned} \tag{3.2}$$

$\boldsymbol{Y}$ is a multivariate real-valued random variable and is defined in the complete probability space $(\Omega_{\boldsymbol{Y}}, A_{\boldsymbol{Y}}, P_{\boldsymbol{Y}})$, where $\Omega_{\boldsymbol{Y}}$ is the sample space, $A_{\boldsymbol{Y}}$ is the event space with σ-algebra, and $\mathrm{P}_{\boldsymbol{Y}} : A_{\boldsymbol{Y}} \to [0,1]$ is the probability measure. The multivariate random variable $\boldsymbol{Y} : (\Omega_{\boldsymbol{Y}}, A_{\boldsymbol{Y}}) \to (\mathbb{R}^{n_{\boldsymbol{Y}}}, \mathcal{B}^{n_{\boldsymbol{Y}}})$, with $\mathcal{B}^{n_{\boldsymbol{Y}}}$ the representation of the Borel σ-field on $\mathbb{R}^{n_{\boldsymbol{Y}}}$, is characterized by the PDF $\boldsymbol{\rho_Y} : \mathbb{R}^{n_{\boldsymbol{Y}}} \to \mathbb{R}_0^+$ (with $\rho_{Y_i} : \mathbb{R} \to \mathbb{R}_0^+$ for each component $i = 1, \cdots, n_{\boldsymbol{Y}}$). $\{\rho_{Y_i}\}_{i=0}^{n_{\boldsymbol{Y}}}$ are in the finite-dimensional second-order random space L^2. We are interested in evaluating the *statistics of interest* (SoI) of our QoI, and we look at two specific statistics given by

• expected value:

$$\mathbb{E}[\mathcal{F}(\boldsymbol{x},\boldsymbol{\theta})] := \int_{\Omega_{\boldsymbol{Y}}} \mathcal{F}(\boldsymbol{x},\boldsymbol{\theta}) \boldsymbol{\rho_Y} d\boldsymbol{Y}, \tag{3.3}$$

• and variance:

$$\mathbb{V}[\mathcal{F}(\boldsymbol{x},\boldsymbol{\theta})] := \int_{\Omega_{\boldsymbol{Y}}} (\mathcal{F}(\boldsymbol{x},\boldsymbol{\theta}) - \mathbb{E}[\mathcal{F}(\boldsymbol{x},\boldsymbol{\theta})])^2 \boldsymbol{\rho_Y} d\boldsymbol{Y} = \mathbb{E}\left[\mathcal{F}(\boldsymbol{x},\boldsymbol{\theta})^2\right] - \mathbb{E}[\mathcal{F}(\boldsymbol{x},\boldsymbol{\theta})]^2. \tag{3.4}$$

A third SoI derives from the variance, namely the standard deviation given by:

$$\sigma[\mathcal{F}(\boldsymbol{x},\boldsymbol{\theta})] := \sqrt{\mathbb{V}[\mathcal{F}(\boldsymbol{x},\boldsymbol{\theta})]}, \tag{3.5}$$

and the covariance is defined as:

$$\mathrm{Cov}(Y_i, Y_j) = \mathbb{E}[(Y_i - \mathbb{E}[Y_i])(Y_j - \mathbb{E}[Y_j])] \; \forall \, i,j = 1, \ldots, n_{\boldsymbol{Y}}. \tag{3.6}$$

In this work, we assume that the random variables are independent and identically distributed (i.i.d.); thus, the joint distribution yields:

$$\boldsymbol{\rho_\theta} = \prod_{i=1}^{n_{\boldsymbol{\theta}}} \rho_{\theta_i}, \tag{3.7}$$

and

$$\boldsymbol{\rho_Y} = \prod_{i=1}^{n_{\boldsymbol{Y}}} \rho_{Y_i}, \tag{3.8}$$

where ρ_{θ_i} and ρ_{Y_i} are *marginal PDFs*.

Computing the statistics of interest (SoI), such as expected values and variances, necessitates the calculation of multivariate integrals, which, in our case, are of dimension $n_{\mathbf{Y}}$. Typically, these high-dimensional integrals do not have a closed form for the integrand, which originates from a numerical model and must be evaluated numerically. This process requires discrete realizations of $\boldsymbol{\theta}$, denoted as *samples* $\boldsymbol{\theta}^{(i)} = \boldsymbol{\theta}(\boldsymbol{\omega})$, where $\boldsymbol{\omega}$ represents a specific event. These samples, in turn, result in samples of our QoI $\mathbf{Y}^{(i)} = \mathcal{F}^{(i)} = \mathcal{F}(\boldsymbol{x}, \boldsymbol{\theta}^{(i)})$ through forward propagation (see Section 3.2).

3.2. Forward Uncertainty Quantification Pipeline

We incorporate uncertainties into our model $\mathcal{F}$ to more accurately approximate the behaviors of the real-world system, which is inherently stochastic. In this context, Uncertainty Quantification (UQ) problems can be integrated into our numerical model through two categories: *Forward Uncertainty Propagation* (or forward UQ) and *Inverse Uncertainty Quantification* (or inverse UQ).

Forward UQ involves considering the model inputs as deterministic and stochastic, enabling the modeling of stochastic inputs using random variables with PDFs derived from experimental data, theoretical knowledge, or professional opinion. Figure 3.1 illustrates the functioning of the forward UQ pipeline.

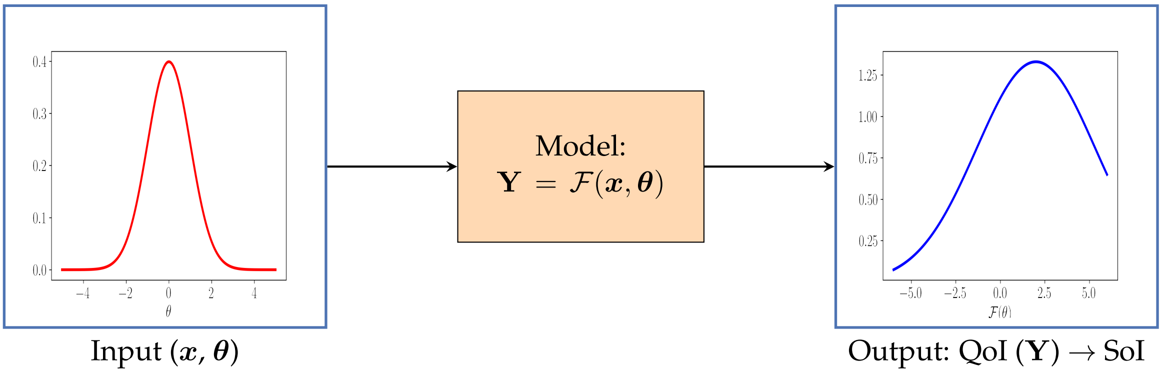

0.4
0.3
0.2
0.1
0.0
−4
−2
0
2
4
θ
Model:
$\mathbf{Y} = \mathcal{F}(\boldsymbol{x}, \boldsymbol{\theta})$
1.25
1.00
0.75
0.50
0.25
−5.0
−2.5
0.0
2.5
5.0
$\mathcal{F}(\theta)$
Input $(\boldsymbol{x}, \boldsymbol{\theta})$
Output: QoI $(\mathbf{Y}) \rightarrow$ SoI

Figure 3.1.: Forward UQ Pipeline.

Inverse UQ is the process of quantifying input uncertainties by working backward from experimental data. This approach aims to refine the estimation of input uncertainties, thereby enhancing the precision of these initial, often ad-hoc, specifications [66].

This study will focus exclusively on forward UQ. Through the forward UQ pipeline, we obtain a set of Quantities of Interest (QoI), from which we can calculate the SoI, such as the expected value and variance.

3.3. Monte Carlo (MC) Sampling

One of the most widely used sampling algorithms in forward uncertainty quantification is Monte Carlo sampling [42], [28], [33]. The primary objective of this method is to generate samples of the QoI using (pseudo-)random number generators, such that the expected value of the QoI, given the number of samples $N_{\text{samples}} \in \mathbb{N}$, is:

$$\widehat{\mu}_1[\mathcal{F}(\boldsymbol{x}, \boldsymbol{\theta})] := \frac{1}{N_{\text{samples}}} \sum_{i=1}^{N_{\text{samples}}} \mathcal{F}\left(\boldsymbol{x}, \boldsymbol{\theta}^{(i)}\right) = \frac{1}{N_{\text{samples}}} \sum_{i=1}^{N_{\text{samples}}} \mathbf{Y}^{(i)}, \tag{3.9}$$

and the sampling variance is given by:

$$\mathbb{V}[\mathcal{F}(\boldsymbol{x}, \boldsymbol{\theta})] = \frac{1}{N_{\text{samples}} - 1} \sum_{i=1}^{N_{\text{samples}}} \left(\mathbf{Y}^{(i)} - \widehat{\mu}_1[\mathcal{F}(\boldsymbol{x}, \boldsymbol{\theta})]\right)^2, \tag{3.10}$$

assuming that the samples $\left\{\boldsymbol{\theta}^{(i)}\right\}_{i=1}^{N_{\text{samples}}}$ are *i.i.d.*.
To express two essential properties of the estimator given in Eq. 3.9, we must first define two crucial quantities of estimators: the *bias* and the *root mean squared error* (RMSE).
The bias of an estimator $\hat{H}$, given the true value (H) of the estimated quantity, is:

$$\text{BIAS}(\hat{H}) = \mathbb{E}[\hat{H}] - H, \tag{3.11}$$

and an estimator is said to be *unbiased* if $\text{BIAS}(\hat{H}) = 0$, i.e. $\mathbb{E}[\hat{H}] = H$.

The RMSE of an estimator is defined by:

$$\text{RMSE}(\hat{H}) = \sqrt{\mathbb{E}\left[(\hat{H} - H)^2\right]} = \sqrt{\text{BIAS}(\hat{H})^2 + \mathbb{V}[\hat{H}]}. \tag{3.12}$$

Now, we can state the two major properties of the estimator from Eq. 3.9:

1. The estimator $\widehat{\mu}_1[\mathcal{F}(\boldsymbol{x}, \boldsymbol{\theta})]$ from MC sampling is an *unbiased* one for $\mathbb{E}[\mathcal{F}(\boldsymbol{x}, \boldsymbol{\theta})]$:

$$\mathbb{E}\left[\widehat{\mu}_1[\mathcal{F}(\boldsymbol{x}, \boldsymbol{\theta})]\right] = \mathbb{E}[\mathcal{F}(\boldsymbol{x}, \boldsymbol{\theta})]. \tag{3.13}$$

2. The RMSE for the estimator $\widehat{\mu}_1[\mathcal{F}(\boldsymbol{x}, \boldsymbol{\theta})]$ is given by:

$$\text{RMSE}\left[\widehat{\mu}_1[\mathcal{F}(\boldsymbol{x}, \boldsymbol{\theta})]\right] = \sqrt{\frac{\mathbb{V}[\mathcal{F}(\boldsymbol{x}, \boldsymbol{\theta})]}{N_{\text{samples}}}}. \tag{3.14}$$

From Eq. 3.14, the rate of convergence of the $\text{RMSE}\,[\widehat{\mu}_1[\mathcal{F}(\boldsymbol{x},\boldsymbol{\theta})]]$ is $\mathcal{O}(1/\sqrt{N_{\text{samples}}})$ [35], indicating that the convergence rate is solely dependent on the number of generated samples. Notably, a lower RMSE corresponds to a higher accuracy of the results. The RMSE can be reduced by decreasing the sampling variance or increasing the number of samples. In this study, we will utilize a minimum sample size of 10^4 for MC simulations of our *Case Study*. This minimum sample size is based on the Gelman-Rubin statistics (see Sub-section 7.2.2) to ensure reliable accuracy of results.

3.4. PPF with MC

In this thesis, we apply Monte Carlo methods to the power flow analysis by adding random fluctuations to the specifications given by equations 2.13. We generate the fluctuation's samples $\boldsymbol{\theta}^{(i)} = \boldsymbol{\theta}(\boldsymbol{\omega}_i)$ for $i = 1, \dots, N_{\text{samples}}$ from the joint probability distribution $\boldsymbol{\rho_\theta}$ of the RV $\boldsymbol{\theta}$ (see Section 3.1), and assign them to the active and reactive loads in the following manner:

$$p_{\text{load}} = p_{\text{load}}^{\text{ref}} \left(1 + \boldsymbol{\theta}^{(i)}\epsilon\right), \tag{3.15}$$

$$q_{\text{load}} = q_{\text{load}}^{\text{ref}} \left(1 + \boldsymbol{\theta}^{(i)}\epsilon\right), \tag{3.16}$$

where $\epsilon \in \mathbb{R}^+$ is a control parameter of fluctuation intensity, that is, we can scale the fluctuations of a given distribution sample to higher or lower values by changing the values of ϵ. In other words, Monte Carlo simulations allow for assessing the power system's performance under various operating conditions by incorporating random fluctuations into the power flow equations [2.6 - 2.8] and 2.13 through Algorithm 1.

Algorithm 1 PPF with MC.

```
Input: i.i.d. samples from the joint distribution $\boldsymbol{\rho_\theta}$; Deterministic inputs $\boldsymbol{x}$.
Output: QoI $\mathbf{Y}$; SoI
for $1 \leq i \leq$ N$_{\text{samples}}$ do
    $p_{\text{load}} \leftarrow p_{\text{load}}^{\text{ref}} \left(1 + \boldsymbol{\theta}^{(i)}\epsilon\right)$        ▷ Set RV realization of active powers
    $q_{\text{load}} \leftarrow q_{\text{load}}^{\text{ref}} \left(1 + \boldsymbol{\theta}^{(i)}\epsilon\right)$        ▷ Set RV realization of reac. powers
    Run $\mathbf{Y}^i = \mathcal{F}(\boldsymbol{x}, \boldsymbol{\theta}^{(i)})$        ▷ Evaluate model for iteration step $i$
    $\mathbf{Y}[i] \leftarrow \mathbf{Y}^{(i)}$
end for
$\hat{\mu} = mean(\mathbf{Y})$
$\mathbb{V}[\mathbf{Y}] = variance(\mathbf{Y})$
Return $\hat{\mu}$, $\mathbb{V}[\mathbf{Y}]$, $\mathbf{Y}$
```

Thus, we lead to obtaining the stochastic output results:

$$\mathbf{Y} = \mathcal{F}(\boldsymbol{x}, \boldsymbol{\theta}) = \left[\mathbf{x}_1^\top, \ldots, \mathbf{x}_N^\top\right]^\top \in \mathbb{R}^{4N \times N_{\text{samples}}}, \tag{3.17}$$

$$\hat{\mu} := \frac{1}{N_{\text{samples}}} \sum_{i=1}^{N_{\text{samples}}} \mathbf{Y}^{(i)}, \tag{3.18}$$

$$\mathbb{V}[\mathbf{Y}] = \frac{1}{N_{\text{samples}} - 1} \sum_{i=1}^{N_{\text{samples}}} (\mathbf{Y}^{(i)} - \hat{\mu})^2. \tag{3.19}$$

To derive the Quantity of Interest from Equations 3.17 to 3.19, it is necessary to perform multiple evaluations of the model $\mathcal{F}(\boldsymbol{x}, \boldsymbol{\theta})$, corresponding to the number of samples of $\boldsymbol{\theta}$. Executing these simulations can be time-intensive, and despite the accuracy of our QoI, Monte Carlo simulations may prove to be highly inefficient or even impractical, depending on the size of the sample set.

4. Polynomial Chaos Expansion (PCE)

The development of the so-called Polynomial Chaos (PC) decomposition dates back to the 1930s, pioneered by Norbert Wiener [65]. This mathematical framework gained renewed attention in engineering through the work of Ghanem et al. [23]. PCE is an infinite series expansion of an output random variable, expressed in orthogonal polynomials of the input random variables. Initially introduced by Wiener for Gaussian input random variables, a proof of convergence accompanied the methodology [14]. This proof cemented the original PCE, referred to as the classical PCE.

The classical PCE was subsequently extended to a generalized PCE [68] to accommodate non-Gaussian input variables, broadening its applicability. Approximations of PCE, achieved by truncating its infinite series, allow for the creation of surrogate models to solve UQ problems. These surrogate models enhance the efficiency of UQ computations while maintaining good accuracy of results, making PCE a valuable tool in uncertainty quantification.

4.1. General PCE

Let a random vector $\boldsymbol{\theta} = (\theta_1, \cdots, \theta_{n_{\boldsymbol{\theta}}})$ be in a finite-dimensional second-order random space $L^2(\Omega_{\boldsymbol{\theta}}, A_{\boldsymbol{\theta}}, P_{\boldsymbol{\theta}})$ [9], as defined in Section 3.1. We can expand any stochastic response $\mathbf{Y}$ with a finite second moment in a convergent series of orthogonal polynomials of the random inputs according to the Cameron-Martin theorem [6], [43]. Thus, a model $\mathcal{P} : \boldsymbol{\theta} \rightarrow \mathbf{Y}$ can be defined such that

$$\mathbf{Y} = \mathcal{P}(\boldsymbol{\theta}) = \sum_{i=0}^{\infty} \boldsymbol{b}_i \boldsymbol{\Psi}_i(\boldsymbol{\theta}), \tag{4.1}$$

where $\{\boldsymbol{\Psi}_i\}$ refers to the multivariate orthogonal polynomial basis constructed as a tensor product of univariate orthogonal polynomials Φ. A single multivariate polynomial is defined as follows:

$$\boldsymbol{\Psi}_i(\boldsymbol{\theta}) = \prod_{k=1}^{n_{\boldsymbol{\theta}}} \Phi_{i_k}\left(\theta_k\right). \tag{4.2}$$

The index i_j, for $j = 0, \ldots, n_{\boldsymbol{\theta}}$, refers to the j-th degree of the i-th univariate polynomial basis.

The space of second-order random variables L^2 is a Lebesgue space (Hilbert space) and is equipped with the inner product and norm, given $f, g \in L^2$, defined by:

$$\langle f, g\rangle = \mathbb{E}[f(\boldsymbol{\theta})g(\boldsymbol{\theta})] = \int_{\Omega_{\boldsymbol{\theta}}} f(\boldsymbol{\theta})g(\boldsymbol{\theta})\boldsymbol{\rho}(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta},\ \|f\| = \sqrt{\langle f, f\rangle}. \tag{4.3}$$

The orthogonality of the polynomials $\boldsymbol{\Psi}_i$ holds with repect to L^2 when satisfying

$$\mathbb{E}\left[\boldsymbol{\Psi}_l(\boldsymbol{\theta})\boldsymbol{\Psi}_k(\boldsymbol{\theta})\right] = \int_{\Omega_{\boldsymbol{\theta}}} \boldsymbol{\Psi}_l(\boldsymbol{\theta})\boldsymbol{\Psi}_k(\boldsymbol{\theta})\boldsymbol{\rho}(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta} = \|\boldsymbol{\Psi}_l\|_2^2\,\delta_{lk}, \tag{4.4}$$

and the same holds for the univariate polynomials:

$$\mathbb{E}\left[\Phi_l\left(\theta_i\right)\Phi_k\left(\theta_i\right)\right] = \int_{\Omega_{\theta_i}} \Phi_l(\theta_i)\Phi_k(\theta_i)\rho_i(\theta_i)\mathrm{d}\theta_i = \gamma_l\delta_{lk}, \tag{4.5}$$

where $\mathbb{E}[\cdot]$ is the expected-value operator, $\boldsymbol{\rho}$ and ρ_i are the probability density functions (PDFs) $\boldsymbol{\rho_\theta} : \mathbb{R}^{n_{\boldsymbol{\theta}}} \to \mathbb{R}_0^+$, $\gamma \in \mathbb{R}^+$ is a positive scalar, and δ is the Kronecker-delta.

If we consider the PDFs $\boldsymbol{\rho_\theta}$ that fit within the Askey-Wilson scheme (see Table 4.1) [3], we can determine the corresponding orthogonal polynomials almost instantaneously. However, for other types of PDFs, the expansion must be constructed to ensure the orthogonality property is maintained [61].

It is crucial to highlight that PCE is a good approximation model because the polynomial basis is orthogonal concerning the PDFs of the input random variables, which leads to the best approximation we can get when truncating the polynomial series. For this reason, the class of the polynomial basis depends on the input's stochastic distribution.

Table 4.1.: Askey-Wilson Scheme Table.

PDF Type	Support	Polynomial Basis
Beta	$(-1, 1)$	Jacobi
Gamma	$(0, \infty)$	Laguerre
Gaussian	$(-\infty, \infty)$	Hermite
Uniform	$[-1, 1]$	Legendre

4.1.1. Truncation of Polynomial Series

The infinite series expansion in Eq. 4.1 is impractical for real-world applications. Therefore, to apply PCE in practical scenarios, the series in Eq. 4.1 is usually truncated to include only the first $L+1$ polynomials with a degree of at most p, yielding:

$$L + 1 = \binom{n_{\boldsymbol{\theta}} + p}{p} = \frac{(n_{\boldsymbol{\theta}} + p)!}{n_{\boldsymbol{\theta}}!p!}. \tag{4.6}$$

4.1.2. Statistical Moments with PCE

The computation of statistical moments using PCE is not only straightforward but also highly efficient. With PCE, one can quickly and accurately determine the expected value and the variance, as follows:

$$\mu(\mathbf{Y}) = \boldsymbol{b}_0 \tag{4.7}$$

$$\sigma^2(\mathbf{Y}) = \sum_{i=1}^{L} \boldsymbol{b}_i^2 \langle \Phi_i, \Phi_i \rangle \tag{4.8}$$

4.2. Evaluation of PCE Coefficients

The PCE coefficients can be evaluated in several ways and methods [67] with intrusive and non-intrusive approaches. In this section, we discuss three of them: *The Pseudo-spectral Approach*, *The Regression Approach*, and *The Stochastic Galerkin Method*.

4.2.1. The Pseudo-spectral Approach

The pseudo-spectral approach, classified as a *non-intrusive projection method*, leverages the orthogonality of the polynomial basis [61]. This method involves multiplying Equation 4.1 by $\boldsymbol{\Psi}_j$ and integrating with respect to the joint PDF $\rho_{\boldsymbol{\theta}}$, yielding:

$$\boldsymbol{b}_j = \mathbb{E}\left[\mathcal{P}(\boldsymbol{\theta})\boldsymbol{\Psi}_j(\boldsymbol{\theta})\right] \equiv \int_{\Omega_{\boldsymbol{\theta}}} \mathcal{P}(\boldsymbol{\theta})\boldsymbol{\Psi}_j(\boldsymbol{\theta})\rho_{\boldsymbol{\theta}}(\boldsymbol{\theta})d\boldsymbol{\theta} = \langle \boldsymbol{Y}, \boldsymbol{\Psi}_j \rangle. \tag{4.9}$$

With Eq. 4.9, one can obtain the polynomial coefficients $\boldsymbol{b}_j$. In practice, Eq. 4.9 is estimated using classical methods for numerical integration. These methods involve approximating the multidimensional integral through a weighted sum. This approach enables the transformation of a complex integral into a more manageable form. It facilitates its evaluation by summing the integrand values at specific points, each multiplied by a corresponding weight as follows:

$$\hat{\boldsymbol{b}}_j \approx \sum_{i=1}^{K} w_i \mathcal{P}\left(\boldsymbol{\theta}^{(i)}\right) \boldsymbol{\Psi}_j\left(\boldsymbol{\theta}^{(i)}\right), \tag{4.10}$$

where K is the number of integration points of a chosen quadrature rule. Several numerical quadrature techniques can be employed to select the integration points $\boldsymbol{\theta}^{(i)}$ and their corresponding weights w_i. Notable examples include the Gauss quadrature and the trapezoidal rule, among others. These methods offer various approaches to accurately approximate the value of the integral by strategically choosing the points and weights to optimize the numerical estimation.

The so-called *simulation method* relies upon the choice of M random integration points, with integration weights being $1/M$, yielding:

$$\hat{\boldsymbol{b}}_j \approx \frac{1}{M}\sum_{i=1}^{M}\mathcal{P}\left(\boldsymbol{\theta}^{(i)}\right)\boldsymbol{\Psi}_j\left(\boldsymbol{\theta}^{(i)}\right). \tag{4.11}$$

This corresponds to applying Monte Carlo simulation to estimate the expectation value in Eq. 4.7. In this case, the accuracy of the coefficients depends on the chosen sampling approach.

4.2.2. Regression Approach

The regression approach is an alternative to the pseudo-spectral approach as a non-intrusive method. It consists of computing the PCE coefficients which provide the best approximation of $\boldsymbol{Y} = \mathcal{F}(\boldsymbol{x}, \boldsymbol{\theta})$ by using regression methods such as Least Squares [61]. We can elaborate better on this method considering a truncated PC expansion with $L+1$ terms, with:

$$\mathbf{b} = [\boldsymbol{b}_0, \dots, \boldsymbol{b}_L]^\top, \tag{4.12}$$

$$\boldsymbol{\Psi}(\boldsymbol{\theta}) = [\Psi_0(\boldsymbol{\theta}), \dots, \Psi_L(\boldsymbol{\theta})]^\top. \tag{4.13}$$

Eq. 4.1 can be then rewritten in a truncated form as:

$$\boldsymbol{Y} = \mathbf{b}^\top\boldsymbol{\Psi}(\boldsymbol{\theta}) \tag{4.14}$$

Consider the set of input realizations $\mathcal{X} \equiv [\boldsymbol{\theta}^{(1)} \dots \boldsymbol{\theta}^{(n)}]^\top$, namely an *experimental design*, where $n = N_{\text{samples}}$ and $\mathcal{Y}$ is the set of corresponding model evaluations. The problem consists in finding the vector of coefficients $\hat{\mathbf{b}}$ that minimize the sum of squared errors such that:

$$\hat{\mathbf{b}} = \arg\min_{\mathbf{b}} \sum_{i=1}^{n}\left(\mathbf{b}^\top\boldsymbol{\Psi}\left(\boldsymbol{\theta}^{(i)}\right) - \mathcal{F}\left(\boldsymbol{\theta}^{(i)}\right)\right)^2. \tag{4.15}$$

The solution for the problem from Eq. 4.15 can be obtained in closed form as follows [61]:

$$\hat{\mathbf{b}} = \left(\boldsymbol{\Psi}^\top\boldsymbol{\Psi}\right)^{-1}\boldsymbol{\Psi}^\top\mathcal{Y}, \tag{4.16}$$

where the entries for $\boldsymbol{\Psi}$ are given by:

$$\boldsymbol{\Psi}_{ij} = \boldsymbol{\Psi}_j \left(\boldsymbol{\theta}^{(i)} \right) \quad i = 1, \dots, n \quad j = 0, \dots, L. \tag{4.17}$$

To get a well-posed problem, the number of model evaluations n must be greater than the number $L + 1$ of unknown coefficients. The rule of thumb $n = 2L$ generally leads to good results [61].

4.2.3. Stochastic Galerkin Method

The stochastic Galerkin method relies on the PC expansion to find a solution for the coefficients. The method is similar to the Finite Elements (FE) method in finding the best approximation for the PCE coefficients when embedding the PC expansion in the model's equations. The orthogonal polynomials work as the shape functions in the FE analogy, and the coefficients can be found by solving the resulting system of equations from the PC expansion embedded in the system's equations. The stochastic Galerkin method unfolds in a series of steps, each building upon the previous one. This iterative process is as follows:

1. Expand the stochastic inputs as a series of PCE.
2. Write the model's solution as a p-th order PCE.
3. Insert both expansions above into the system's equations $\mathcal{F}$.
4. Take advantage of the polynomials orthogonality to get a system of equations with $L + 1$ unknowns.

When following the steps described above in the Power Flow equation, we obtain the equations as described in table 4.2. The stochastic Galerkin method is an intrusive one and we need to modify the system's equations. It is necessary access to the model, however there is no quadrature error such as in the pseudo-spectral approach. This method is more accurate, however more complex to be applied.

The idea of intrusive PCE is to plug the gPCE into the system's equations $\mathcal{F}(\boldsymbol{x}, \boldsymbol{\theta})$, truncating the series to $L + 1$ functions and performing a Galerkin projection [4], while the main idea of non-intrusive PCE is to compute the coefficient functions as numerical integrals, where the integral in Eq. 4.9 is a Banach space-valued integral [17].

Table 4.2.: Power Flow Equations and Statistical Moments in Terms of PCE coefficients [41].

Rectangular power flow in terms of PCE coefficients with $i \in \mathcal{N}, l \in \mathcal{L}$, where $\mathcal{L} = \{0, \dots, L\}$
$\langle \mathbf{\Psi}_l, \mathbf{\Psi}_l \rangle (p_{i,l}) = \sum_{j \in \mathcal{N}} \sum_{l_0, l_1 \in \mathcal{L}} \left\langle \mathbf{\Psi}_{l_0} \mathbf{\Psi}_{l_1}, \mathbf{\Psi}_l \right\rangle \left(G_{ij} \left(v^{\mathrm{re}}_{i,l_0} v^{\mathrm{re}}_{j,l_1} + v^{\mathrm{im}}_{i,l_0} v^{\mathrm{im}}_{j,l_1} \right) + B_{ij} \left(v^{\mathrm{im}}_{i,l_0} v^{\mathrm{re}}_{j,l_1} - v^{\mathrm{re}}_{i,l_0} v^{\mathrm{im}}_{j,l_1} \right) \right)$
$\langle \mathbf{\Psi}_l, \mathbf{\Psi}_l \rangle (q_{i,l}) = \sum_{j \in \mathcal{N}} \sum_{l_0, l_1 \in \mathcal{L}} \left\langle \mathbf{\Psi}_{l_0} \mathbf{\Psi}_{l_1}, \mathbf{\Psi}_l \right\rangle \left(G_{ij} \left(v^{\mathrm{im}}_{i,l_0} v^{\mathrm{re}}_{j,l_1} - v^{\mathrm{re}}_{i,l_0} v^{\mathrm{im}}_{j,l_1} \right) - B_{ij} \left(v^{\mathrm{re}}_{i,l_0} v^{\mathrm{re}}_{j,l_1} + v^{\mathrm{im}}_{i,l_0} v^{\mathrm{im}}_{j,l_1} \right) \right)$
Moments of squared line current magnitudes with $ij \in L, v^{\mathrm{re}}_{ij,l} = v^{\mathrm{re}}_{i,l} - v^{\mathrm{re}}_{j,l}, v^{\mathrm{im}}_{ij,l} = v^{\mathrm{im}}_{i,l} - v^{\mathrm{im}}_{j,l}$
$\mathbb{E}\left[i^2_{i \to j} \right] = \left\| y^{\mathrm{br}}_{ij} \right\|^2 \sum_{l \in \mathcal{L}} \langle \mathbf{\Psi}_l, \mathbf{\Psi}_l \rangle \left(\left(v^{\mathrm{re}}_{ij,l} \right)^2 + \left(v^{\mathrm{im}}_{ij,l} \right)^2 \right)$
$\sigma \left[i^2_{i \to j} \right]^2 = \left\| y^{\mathrm{br}}_{ij} \right\|^4 \sum_{l_0, l_1, l_2, l_3 \in \mathcal{L}} \left\langle \mathbf{\Psi}_{l_0} \mathbf{\Psi}_{l_1} \mathbf{\Psi}_{l_2}, \mathbf{\Psi}_{l_3} \right\rangle \left(v^{\mathrm{re}}_{i,l_0} v^{\mathrm{re}}_{ij,l_1} v^{\mathrm{re}}_{i,l_2} v^{\mathrm{re}}_{ij,l_3} + 2 v^{\mathrm{re}}_{ij,l_0} v^{\mathrm{re}}_{ij,l_1} v^{\mathrm{im}}_{ij,l_2} v^{\mathrm{im}}_{ij,l_3} + v^{\mathrm{im}}_{ij,l_0} v^{\mathrm{im}}_{ij,l_1} v^{\mathrm{im}}_{ij,l_2} v^{\mathrm{im}}_{ij,l_3} \right) - \mathbb{E}\left[i^2_{i \to j} \right]^2$
Moments of squared voltage magnitudes with $i \in \mathcal{N}$
$\mathbb{E}\left[V^2_i \right] = \sum_{l \in \mathcal{L}} \langle \mathbf{\Psi}_l, \mathbf{\Psi}_l \rangle \left(\left(v^{\mathrm{re}}_{i,l} \right)^2 + \left(v^{\mathrm{im}}_{i,l} \right)^2 \right)$
$\sigma \left[V^2_i \right]^2 = \sum_{l_0, l_1, l_2, l_3 \in \mathcal{L}} \left\langle \mathbf{\Psi}_{l_0} \mathbf{\Psi}_{l_1} \mathbf{\Psi}_{l_2}, \mathbf{\Psi}_{l_3} \right\rangle \left(v^{\mathrm{re}}_{i,l_0} v^{\mathrm{re}}_{i,l_1} v^{\mathrm{re}}_{i,l_2} v^{\mathrm{re}}_{i,l_3} + 2 v^{\mathrm{re}}_{i,l_0} v^{\mathrm{re}}_{i,l_1} v^{\mathrm{im}}_{i,l_2} v^{\mathrm{im}}_{i,l_3} + v^{\mathrm{im}}_{i,l_0} v^{\mathrm{im}}_{i,l_1} v^{\mathrm{im}}_{i,l_2} v^{\mathrm{im}}_{i,l_3} \right) - \mathbb{E}\left[V^2_i \right]^2$

5. Sensitivity Analysis

When performing forward UQ over a numerical model $\mathcal{F}$, it is known that uncertainty parameters $\boldsymbol{\theta}$ influence the model's outcomes. However, the extent and manner of this influence remain unknown, as some input uncertainties may be more influential than others on a specific system output. Sensitivity Analysis is crucial in this context, enabling the analysis of how uncertainties in the input $\boldsymbol{\theta}$ affect the QoI $\boldsymbol{Y}$ [49]. This chapter elaborates on three essential methods to perform Sensitivity Analysis: ANOVA, Saltelli, and Rank-based Estimation. The latter, the most recent one, was used to perform a Sensitivity Analysis of the problem analyzed in this work.

ANOVA, or Analysis of Variance, was first introduced by Hoeffding in the 1940s concerning his work on U-statistics [29]. Since its inception, the method has evolved and been studied across several fields, including mathematics [59], statistics [19], finance [25], and various engineering disciplines [51]. ANOVA decomposes the variance of a dataset to attribute portions of the variance to different sources of variation, thereby identifying the impact of each input variable on the output variability.

Sobol's method, presented by I.M. Sobol in 1993, is a foundational technique for calculating global sensitivity indices of a function of independent variables [60]. Building on Sobol's work, Andrea Saltelli developed new methods to efficiently calculate Sobol's indices [54], [55], [56], [57]. Saltelli's method has become a standard for global sensitivity analysis and is known for its robustness in handling complex models. It decomposes the output variance into fractions attributed to inputs and their interactions, providing a comprehensive view of the input-output relationship.

Until recently, rank-estimation methods had not been fully developed to calculate Sobol's indices. However, a recent study by Gamboa et al. [21] introduced a new class of rank-based estimators to calculate Sobol's indices. This new method shows substantial efficiency improvements, making it a valuable tool for sensitivity analysis. Rank-based estimation provides a non-parametric approach, offering robustness against outliers and model assumptions.

In summary, while ANOVA and Saltelli's methods have long been established for sensitivity analysis, the recent advancements in rank-based estimation methods represent a significant leap forward in the efficiency and robustness of calculating sensitivity indices.

5.1. Analysis of Variance (ANOVA)

Considering the multivariate random variable $\boldsymbol{\theta}$ with a probability density measure defined in Eq. 3.7, and our QoI $\boldsymbol{Y} = \mathcal{F}(\boldsymbol{x}, \boldsymbol{\theta})$, the classical ANOVA is structured as a hierarchical expansion of $\boldsymbol{Y}$ in terms of its stochastic input variables θ_i with increasing dimensions [60]. To elaborate on this method, let $D_\theta = \{1, \cdots, n_{\boldsymbol{\theta}}\}$ be the index set of the marginals θ_i of $\boldsymbol{\theta}$, and let $\boldsymbol{v} \subset D_\theta$ be a non-empty subset of indices from D_θ. Additionally, we define $\boldsymbol{\rho_v} = \prod_{\boldsymbol{v}} \rho_{\theta_i}$, with $i \in \boldsymbol{v}$.

We can express each marginal distribution $\{\mathbf{Y}_r = \mathcal{F}_r(\boldsymbol{x}, \boldsymbol{\theta})\}_{r=1}^{n_{\boldsymbol{Y}}}$ of the output random vector as a finite sum of functions of subsets of its inputs $\{\theta_i\}_{i=1}^{n_{\boldsymbol{\theta}}}$:

$$\mathcal{F}_r(\boldsymbol{x}, \boldsymbol{\theta}) = \mathcal{F}_r^0 + \sum_{i=1}^{n_{\boldsymbol{\theta}}} \mathcal{F}_r^i(\boldsymbol{x}, \theta_i) + \sum_{1 \le i < j \le n_{\boldsymbol{\theta}}} \mathcal{F}_r^{i,j}(\boldsymbol{x}, \theta_i, \theta_j) + \ldots + \mathcal{F}_r^{1,2,\ldots,n_{\boldsymbol{\theta}}}(\boldsymbol{x}, \boldsymbol{\theta}), \tag{5.1}$$

which can be rewritten in a compact form:

$$\mathcal{F}_r(\boldsymbol{x}, \boldsymbol{\theta}) = \mathcal{F}_r^0 + \sum_{\boldsymbol{v} \subset D_\theta} \mathcal{F}_r^{\boldsymbol{v}}(\boldsymbol{x}, \boldsymbol{\theta_v}), \tag{5.2}$$

where $\mathcal{F}_r^0$ is a scalar function, and $\mathcal{F}_r^{\boldsymbol{v}}(\boldsymbol{x}, \boldsymbol{\theta_v})$ is a $|\boldsymbol{v}|$-variate component function describing the joint effect of $\boldsymbol{\theta_v}$ on $\boldsymbol{Y}$ for $|\boldsymbol{v}| > 0$ [52]. The total number of component functions in Eq. 5.2 is $2^{n_{\boldsymbol{\theta}}}$. To ensure desirable orthogonal properties of the component functions, *strong annihilating conditions* are applied, yielding:

$$\int_{\Theta_i} \mathcal{F}_r^{\boldsymbol{v}}(\boldsymbol{x}, \boldsymbol{\theta_v}) \boldsymbol{\rho}_i(\boldsymbol{\theta}_i) \mathrm{d}\boldsymbol{\theta}_i = 0, \quad \forall i \in \boldsymbol{v}, \forall \boldsymbol{v} \subset D_\theta, \tag{5.3}$$

i.e., the strong annihilating conditions relevant to the ANOVA require all nonconstant component functions $\mathcal{F}_r^{\boldsymbol{v}}(\boldsymbol{x}, \boldsymbol{\theta_v})$ to integrate to zero concerning the marginal density of each random variable with index in $\boldsymbol{v}$, $\forall \boldsymbol{v} \subset D_\theta$.

The decomposition from Eq. 5.2 with the annihilating conditions from Eq. 5.3 is named *Sobol-Hoeffding* or Analysis of Variance (ANOVA) decomposition and allows the following decomposition of the variance:

$$\mathbb{V}[\mathcal{F}_r(\boldsymbol{x}, \boldsymbol{\theta})] = \sum_{i=1}^{n_{\boldsymbol{\theta}}} \mathbb{V}[\mathcal{F}_r^i(\boldsymbol{x}, \theta_i)] + \sum_{1 \le i < j \le n_{\boldsymbol{\theta}}} \mathbb{V}[\mathcal{F}_r^{i,j}(\boldsymbol{x}, \theta_i, \theta_j)] + \ldots + \mathbb{V}[\mathcal{F}_r^{1,2,\ldots,n_{\boldsymbol{\theta}}}(\boldsymbol{x}, \boldsymbol{\theta})], \tag{5.4}$$

which can be writen in a compact form as below:

$$\mathbb{V}[\mathcal{F}_r(\boldsymbol{x}, \boldsymbol{\theta})] = \sum_{\boldsymbol{v} \subset D_{\boldsymbol{\theta}}} \mathbb{V}[\mathcal{F}_r^{\boldsymbol{v}}(\boldsymbol{x}, \boldsymbol{\theta_v})]. \tag{5.5}$$

From Eq. 5.5, one can define the total Sobol's indices for the global sensitivity analysis [58], [60], which can be extended for the cases where our output $\mathbf{Y}$ is a realization of a random vector with stochastic dimention $n_{\boldsymbol{Y}}$. In this latter case, we consider the matrix of total Sobol's indices $\boldsymbol{S_v} \in \mathbb{R}^{n_{\boldsymbol{Y}} \times (2^{n_{\boldsymbol{\theta}}}-1)}$, with $n_{\boldsymbol{Y}}$ rows and $2^{n_{\boldsymbol{\theta}}} - 1$ columns, with elements given by:

$$S_{r\boldsymbol{v}} = \frac{\mathbb{V}[\mathcal{F}_r^{\boldsymbol{v}}]}{\mathbb{V}[\mathcal{F}_r]}, \tag{5.6}$$

where $\mathcal{F}_r$ stands for a scalar value from our QoI $\mathbf{Y}$, for all $r = 1, \ldots, n_{\boldsymbol{Y}}$ and $\boldsymbol{v} \subset D_\theta$.

Similarly, the matrix of first-order Sobol's indices $\boldsymbol{S} \in \mathbb{R}^{n_{\boldsymbol{Y}} \times n_{\boldsymbol{\theta}}}$, with $n_{\boldsymbol{Y}}$ rows and $n_{\boldsymbol{\theta}}$ columns, has its elements given by:

$$S_{rj} = \frac{\mathbb{V}[\mathcal{F}_r^j]}{\mathbb{V}[\mathcal{F}_r]}, \tag{5.7}$$

for all $r = 1, \ldots, n_{\boldsymbol{Y}}$ and $j = 1, \ldots, n_{\boldsymbol{\theta}}$.

These indices quantify the relative contribution of the input subset $\boldsymbol{v}$ to the overall output variance $\mathbb{V}[\mathcal{F}]$. In essence, Sobol's indices assess the significance of the stochastic inputs in relation to the uncertainty of the output. In particular, the first-order Sobol's indices individually measure each stochastic input's importance.

Applying ANOVA and computing Sobol's indices is highly computationally expensive, especially when the stochastic space dimension is high (*curse of dimensionality*). More efficient methods for the analysis of variance are necessary, and we elaborate on two of them in the following sections: *Saltelli* and *Rank-Estimation*.

5.2. Saltelli

Considering $n = N_{\text{samples}}$ and the existence of two independent input sample matrices $\mathbf{X}$, $\mathbf{W} \in \mathbb{R}^{n \times k}$, with generic elements x_{ij}, $w_{ij} \in \mathbb{R}$, and $k = n_{\boldsymbol{\theta}}$, we define the following: Let $\mathbf{X}_{\mathbf{W}}^{(j)}$ be the matrix where all columns are from $\mathbf{X}$, except the j-th column, which is from $\mathbf{W}$. Similarly, we define $\mathbf{W}_{\mathbf{X}}^{(j)}$. Using these matrices, we can calculate the numerator of the first-order Sobol's indices from Equation 5.7 and the local Sobol's indices from Equation 5.6 as follows [53]:

$$\mathbb{V}[\mathcal{F}_r^j] = \mathbb{V}_{X_j}(\mathbb{E}_{\mathbf{X}_{\sim j}}(\mathbf{Y}_r \mid X_j)) = \tfrac{1}{n}\textstyle\sum_{i=1}^n \mathcal{F}_r(\mathbf{X})_i(\mathcal{F}_r(\mathbf{W}_{\mathbf{X}}^{(j)})_i - \mathcal{F}_r(\mathbf{W})_i), \tag{5.8}$$

$$\mathbb{V}[\mathcal{F}_r^{\boldsymbol{v}}] = \mathbb{V}_{\mathbf{X}_{\sim j}}(\mathbb{E}_{X_j}(\mathbf{Y}_r \mid \mathbf{X}_{\sim j})) = \mathbb{V}\left[\mathbf{Y}_r\right] - \tfrac{1}{n}\textstyle\sum_{i=1}^n \mathcal{F}_r(\mathbf{X})_i(\mathcal{F}_r(\mathbf{X})_i - \mathcal{F}_r(\mathbf{X}_{\mathbf{W}}^{(j)})_i), \tag{5.9}$$

$\forall j = 1, \cdots, k$ and $\boldsymbol{v} \subset D_\theta$. $\{\mathcal{F}_r(\mathbf{X})_i\}_{r=1}^{n_{\boldsymbol{Y}}}$ represents the model outputs evaluated at the sample points $i = 1, \cdots, n$ from matrix $\mathbf{X}$. In addition, $\mathcal{F}_r(\mathbf{W}_{\mathbf{X}}^{(j)})$ and $\mathcal{F}_r(\mathbf{X}_{\mathbf{W}}^{(j)})$ represent the model output at the modified sample points, where the j-th column has been swapped between $\mathbf{X}$ and $\mathbf{W}$. These calculations allow for the quantification of the contribution of each input variable to the output variance, facilitating a comprehensive sensitivity analysis. However, generating additional input samples and new MC simulations is necessary to calculate the outputs considering the swapped matrices and the additional sample. In the next section, we will see a new way of calculating Sobol's indices without generating new samples as necessary to compute Eq. 5.8.

5.3. Rank-Estimation

Rank estimation is a new class of estimators to calculate Sobol's indices [21]. This new class allows the computation of Sobol's indices in a way that does not require to create any specific sampling type, such as in the case of Saltelli sensitivity analysis, making computation way more feasible. To start presenting this class, we shall discuss Chatterjee's correlation coefficient, which relies on Cramér-von-Mises indices [22].

5.3.1. Chatterjee's correlation coefficient

Let $(\boldsymbol{\theta}, \boldsymbol{Y})$ be a pair of real-valued multivariate random variables with an i.i.d. sample $\{\boldsymbol{\theta}^j, \mathbf{Y}^j\}_{j=1}^n$, also we assume that $n = N_{\text{samples}}$ and the stochastic dimensions $n_{\boldsymbol{\theta}}$, $n_{\boldsymbol{Y}} \in \mathbb{N}$ are not necessarily equal. The pairs of marginal distributions $\{\theta_l^j, Y_r^j\}_{j=1}^n$ for all $l = 1, \cdots, n_{\boldsymbol{\theta}}$ and $r = 1, \cdots, n_{\boldsymbol{Y}}$ are rearranged in such a way that $\theta_l^1 < \ldots < \theta_l^n$.

We can introduce the correlation coefficient matrix $\xi_n(\boldsymbol{\theta}, \boldsymbol{Y}) \in \mathbb{R}^{n_{\boldsymbol{Y}} \times n_{\boldsymbol{\theta}}}$, which is an extension of the real-valued correlation coefficient between two general uni-variate (X, Z) random variables defined by Chatterjee in [15].

Given the rank $\pi_l(j)$ of $\{\theta_l^j\}_{j=1}^n$, that is, the number of i such that $\theta_l^i \leq \theta_l^j$, we define $N_l(j)$ as:

$$N_l(j) = \begin{cases} \pi_l^{-1}(\pi_l(j) + 1) & \text{if } \pi_l(j) + 1 \leq n \\ \pi_l^{-1}(1) & \text{if } \pi_l(j) = n \end{cases}. \tag{5.10}$$

The entries $\xi_n^{rl} = \xi_n(\theta_l, Y_r)$ of the correlation coefficient matrix, for all $l = 1, \cdots, n_{\boldsymbol{\theta}}$ and $r = 1, \cdots, n_{\boldsymbol{Y}}$, are given by:

$$\xi_n^{rl} = \frac{\sum_{j=1}^n \frac{1}{n} \sum_{k=1}^n \mathbf{1}_{\{Y_r^k \leq Y_r^j\}} \mathbf{1}_{\{Y_r^k \leq Y_r^{N_l(j)}\}} - (\frac{1}{n} \sum_{k=1}^n \mathbf{1}_{\{Y_r^j \leq Y_r^k\}})^2}{\sum_{j=1}^n F_n(Y_r^j)(1 - F_n(Y_r^j))}, \tag{5.11}$$

with the indicator function $\mathbf{1}_{\{X_i \leq x\}}$:

$$\mathbf{1}_{\{X_i \leq x\}} = \begin{cases} 1 & \text{if } X_i \leq x \\ 0 & \text{otherwise} \end{cases}, \tag{5.12}$$

and the empirical distribution function of the marginal distribution Y_r given by:

$$F_n(\tau) = \sum_{k=1}^{n} \mathbf{1}_{\{Y_r^k \leq \tau\}}. \tag{5.13}$$

We then obtain the correlation coefficient matrix:

$$\xi_n(\boldsymbol{\theta}, \boldsymbol{Y}) = \begin{bmatrix} \xi_n^{11} & \xi_n^{12} & \cdots & \xi_n^{1n_{\boldsymbol{\theta}}} \\ \xi_n^{21} & \xi_n^{22} & \ddots & \xi_n^{2n_{\boldsymbol{\theta}}} \\ \vdots & \ddots & \ddots & \vdots \\ \xi_n^{n_{\boldsymbol{Y}}1} & \xi_n^{n_{\boldsymbol{Y}}2} & \cdots & \xi_n^{n_{\boldsymbol{Y}}n_{\boldsymbol{\theta}}} \end{bmatrix} \in \mathbb{R}^{n_{\boldsymbol{Y}} \times n_{\boldsymbol{\theta}}}, \tag{5.14}$$

that consistently estimates how each marginal distribution $\{Y_r\}_{r=1}^{n_{\boldsymbol{Y}}}$ depends on the marginal distributions $\{\theta_l\}_{l=1}^{n_{\boldsymbol{\theta}}}$. A matrix entry $\xi_n^{rl} \in [0,1]$ is 0 if and only if the variables θ_l and Y_r are independent and 1 if and only if one variable is a measurable function of the other, i.e. $Y_r = f(\theta_l)$.

5.3.2. Rank-estimation definition

With the results from above, we can levarage the construction of a new family of estimators for Sobol's indices [21], [60]. Considering our model $\boldsymbol{Y} \equiv \mathcal{F}(\boldsymbol{x}, \boldsymbol{\theta})$, we want to estimate the first-order Sobol's indices $S_n^{rl} = S_n^{rl}(\theta_l, Y_r)$ for all sample pairs $\{\theta_l^j, Y_r^j\}_{j=1}^n$ of marginal distributions (θ_l, Y_r) given $l = 1, \cdots, n_{\boldsymbol{\theta}}$ and $r = 1, \cdots, n_{\boldsymbol{Y}}$ and $N_l(j)$ defined in Eq. 5.10. The matrix $S_n(\boldsymbol{\theta}, \boldsymbol{Y})$ with first-order Sobol's indices of the marginal distribution pairs (θ_l, Y_r) has its elements given by:

$$S_n^{rl} := \frac{\frac{1}{n}\sum_{j=1}^n Y_r^j Y_r^{N_l(j)} - \left(\frac{1}{n}\sum_{j=1}^n Y_r^j\right)^2}{\frac{1}{n}\sum_{j=1}^n \left(Y_r^j\right)^2 - \left(\frac{1}{n}\sum_{j=1}^n Y_r^j\right)^2}. \tag{5.15}$$

We can then represent $S_n(\boldsymbol{\theta}, \boldsymbol{Y})$ as follows:

$$S_n(\boldsymbol{\theta}, \boldsymbol{Y}) = \begin{bmatrix} S_n^{11} & S_n^{12} & \cdots & S_n^{1n_{\boldsymbol{\theta}}} \\ S_n^{21} & S_n^{22} & \ddots & S_n^{2n_{\boldsymbol{\theta}}} \\ \vdots & \ddots & \ddots & \vdots \\ S_n^{n_{\boldsymbol{Y}}1} & S_n^{n_{\boldsymbol{Y}}2} & \cdots & S_n^{n_{\boldsymbol{Y}}n_{\boldsymbol{\theta}}} \end{bmatrix} \in \mathbb{R}^{n_{\boldsymbol{Y}} \times n_{\boldsymbol{\theta}}}. \tag{5.16}$$

To illustrate how this method is straightforward, an example on how to implement it in Python for an arbitrary Y_r is given in B.1.

6. Clustering Algorithms

Clustering, or cluster analysis, is a fundamental technique in data analysis that is valuable for identifying and categorizing distinct groups within datasets. As an unsupervised learning problem, clustering does not rely on predefined labels, allowing the data to reveal underlying structures. Numerous algorithms are available to implement clustering, each offering unique approaches and benefits. We want specifically to cluster stochastic input data based on geographical locations.

This chapter provides a concise overview of three notable clustering algorithms: Affinity Propagation, Clauset-Newman-Moore greedy modularity, and K-Means. We will focus on applying the K-Means algorithm in our case study, demonstrating its practical utility and effectiveness in data categorization.

6.1. Affinity Propagation

Affinity propagation uses as input similarity measures amid pairs of data points. This method does not require the number of clusters to be specified beforehand [20]. Instead, it simultaneously considers all data points as potential exemplars or cluster centers. The algorithm iteratively refines the selection of exemplars by exchanging real-valued messages between data points. These messages convey two types of information: responsibility, which reflects how well-suited a point is to serve as an exemplar for another point, and availability, which indicates the appropriateness of a point being chosen as an exemplar by another point [39].

The message-passing process continues until convergence, resulting in a high-quality set of exemplars and their corresponding clusters. This iterative refinement ensures that the final set of exemplars and clusters is based on the overall data structure rather than an arbitrary initial configuration. The ability of affinity propagation to dynamically determine the number of clusters and to iteratively improve the clustering quality makes it a powerful alternative to traditional methods like K-Means, especially in complex and diverse data environments.

We will not use this method in this thesis as we want to specify beforehand the number of clusters, and this can be done more efficiently using K-Means.

6.2. Clauset-Newman-Moore greedy modularity

Greedy modularity works on detecting communities by *modularity maximization* [45], which involves defining a benefit function, known as modularity, that measures the quality of network divisions into communities [24]. The following equation gives the considered modularity function[1]:

$$Q(\gamma) = \frac{1}{2m} \sum_{ij} \left(A_{ij} - \gamma \frac{k_i k_j}{2m} \right) \delta_{g_i g_j}, \tag{6.1}$$

where m is the number of edges (or the sum of all edge weights as in [8]), A is the adjacency matrix of the graph G, k_i is the (weighted) degree of node i, γ is the resolution parameter, and $\delta_{g_i g_j}$ is 1 if nodes g_i and g_j are in the same community and 0 otherwise. Clauset-Newman-Moore greedy modularity maximization finds the community partition with the largest modularity [16].
As this method relies on the maximization of Equation 6.1, we can set an interval for the desired number of clusters. However, the exact final number of clusters, based on our desired interval, is determined by the maximization result of Equation 6.1, which means we cannot precisely control the final total number of clusters. Consequently, this method is not well-suited for clustering power grid buses based on their geographical location, and hence, we do not employ it in this work.

6.3. K-Means Clustering

The K-Means algorithm is widely used for clustering data partitionally into distinct groups based on feature similarity. This unsupervised learning technique aims to partition a given set of N data points into n clusters, where each data point belongs to the cluster with the nearest mean. The algorithm minimizes the within-cluster variance, known as the sum of squared errors (SSE) $J = \sum_{j=1}^{n} \sum_{z_i \in C_j} \|z_i - c_j\|^2$, where $\mathbf{c} = [c_1, \cdots, c_n]^T \in \mathbb{R}^{n \times d}$ is the operator with chosen centroids, $\mathbf{z} = [z_1, \cdots, z_N]^T \in \mathbb{R}^{N \times d}$ is the operator with data points, and d is the dimensionality of the data set. In our case, we use K-Means to cluster the stochastic input data to specific regions of the grid.

The steps involved in the K-Means algorithm are described below:

[1] https://networkx.org/documentation/stable/reference/algorithms/generated/networkx.algorithms.community.modularity_max.greedy_modularity_communities.html

Algorithm 2 K-Means.

```
Input: Number of clusters $n$; Initial centroids $\mathbf{c}_0 = \mathbf{z}[n\ \ random\ \ indices]$
Output: Clustered data sets $\mathbf{C}$
$iterations \leftarrow 0$
$initialize\ \mathbf{D} \in \mathbb{R}^{N \times n}$
$\mathbf{C} \leftarrow \emptyset$
$\mathcal{N} \leftarrow dataset\ \ indices$
while $iterations \leq max\ \ iteration\ \ number$ do
    for i in $\mathcal{N}$ do
        for $1 \leq j \leq n$ do
            $d\left(z_i, c_j\right) \leftarrow \left\|z_i - c_j\right\|^2$    ▷ Distance between each data point and each centroid
            $D_{ij} \leftarrow d\left(z_i, c_j\right)$
        end for
        $\mathbf{C}[j] \leftarrow \min_{\forall j=1,\ldots,n} D[i, j]$    ▷ Assign each data point to the closest centroid
    end for
    for j in $n$ do
        $c_j \leftarrow \frac{1}{\|\mathbf{C}[j]\|} \sum_{z_i \in \mathbf{C}[j]} z_i$    ▷ Update centroids using current cluster
    end for
    $iterations \leftarrow iterations + 1$
end while
```

In this work, K-Means clustering is used to group the buses of the power grid network based on their geographical coordinates. Specifically, we can represent the coordinates of the buses as $\mathbf{z} \in \mathbb{R}^{N \times 2}$. This approach allows us to cluster the buses effectively by their spatial locations, which can be particularly useful for simplifying the analysis and reduction of random variables. Moreover, spatial clustering is beneficial in the context of power grids due to the high correlation of fluctuations arising from spatial proximity. For example, photovoltaic (PV) power generation exhibits a significant correlation in output fluctuations due to the close physical proximity of PV installations [34], [62].

Part II.

Implementation of Study Case

7. Implementation of Case Study: *1354pegase*

The *Case 1354pegase*[1] is a Pandapower (see Section A.1) test case network representing a portion of the European high voltage transmission network. The grid network consists of 1354 buses, of which 621 are load buses (with active/reactive powers). In this case study, we evaluate the *1354pegase* grid from Fig. 7.1, using as benchmark the scenarios where Monte Carlo (MC) simulations are performed, and i.i.d. fluctuation samples are assigned to each bus load in the grid, according to the specifications of Eq. 2.13. This results in an input stochastic space with dimension $n_{\boldsymbol{\theta}}$, where each marginal distribution of the stochastic power fluctuations $\boldsymbol{\theta}$ is assigned to the corresponding load bus as described in Eqs. 3.15 and 3.16.

7.1. Case Study Definition

We assume that each bus in the *1354pegase* network should operate with voltage magnitudes $\mathbf{v}$ such that $0.95 \leq \mathbf{v} \leq 1.09$ ($\mathbf{v}$ is considered p.u.), respecting an European high transmission network operational constraints [40], [30]. If any bus operates outside this range of voltage magnitudes, the network is at risk of failure or collapse, and such buses are considered critical. The PPF analysis aims to identify potential critical buses within the grid, given the probability distributions of the fluctuations, and to determine the probability that these critical buses will operate outside the specified voltage limits. We can state our problem in the following mathematical formulation:

$$\mathbb{P}\left(\mathrm{v}_j \leq v_j^{\max}\right) \leq \epsilon_V, \forall_{j \in \mathcal{N}}, \tag{7.1}$$

$$\mathbb{P}\left(\mathrm{v}_j \geq v_j^{\min}\right) \leq \epsilon_V, \forall_{j \in \mathcal{N}}, \tag{7.2}$$

where for our case $v_j^{\min} = 0.95$, $v_j^{\max} = 1.09$, and ϵ_V stands for an acceptable violation probability of the voltage magnitudes.

This case study proposes a new efficient pipeline for performing the PPF analysis, allowing us to gain a time advantage over Monte Carlo simulations in identifying and calculating the likelihood of critical buses. This pipeline leverages grid partitions to reduce the stochastic input space $\boldsymbol{\theta}$, followed by a sensitivity analysis with rank-based estimation to identify the buses that most influence the critical ones. Finally, PCE is used to calculate

[1] https://arxiv.org/abs/1603.01533

the likelihood that the critical buses will operate outside the specified voltage limits, given the reduced stochastic space from the sensitivity analysis.

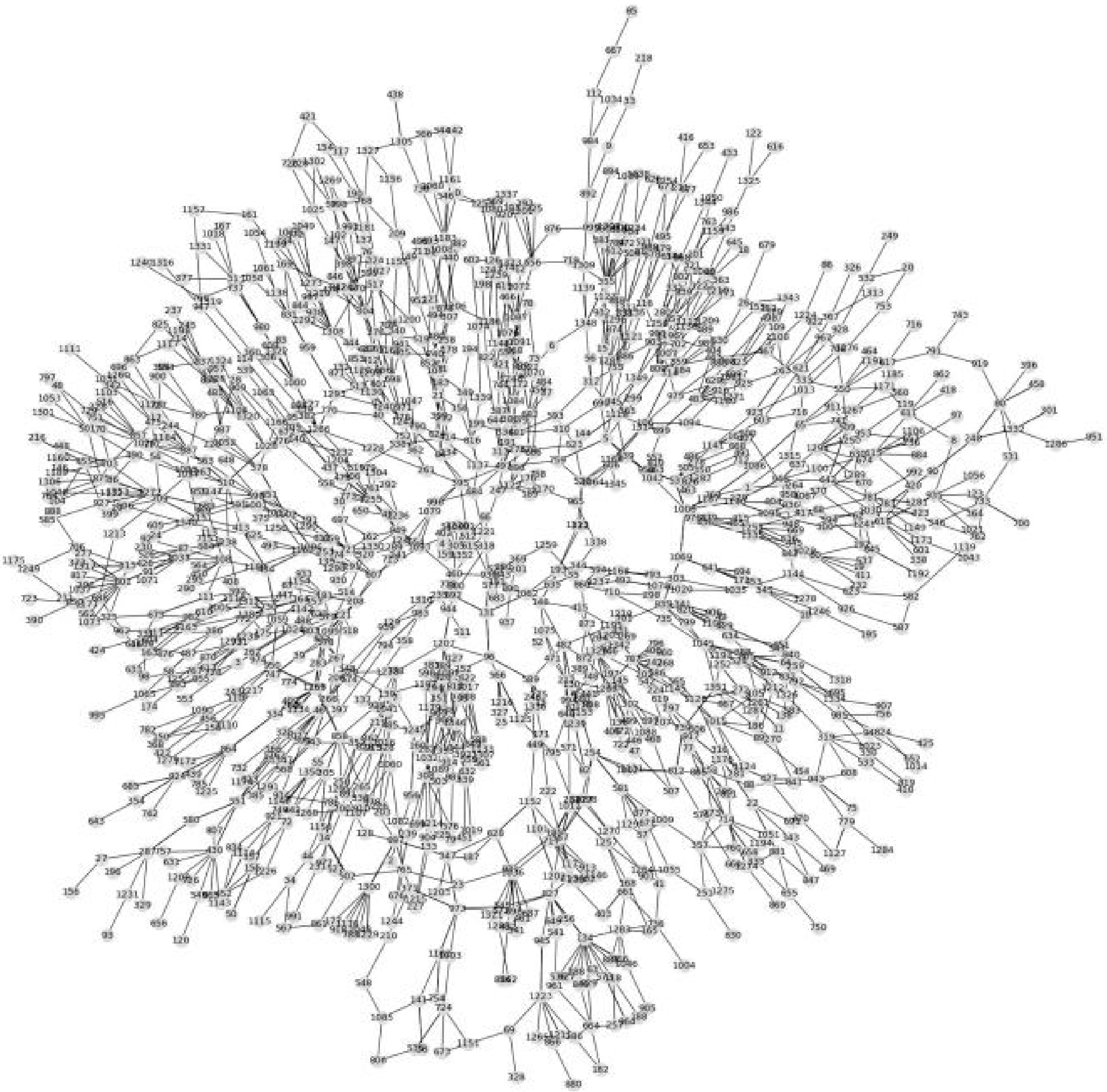


Figure 7.1.: Representation of total grid *1354pegase* with bus labels.

7.2. Standard cases

Five standard cases were considered to perform a UQ analysis across different scenarios for the grid *1354pegase*. These cases are structured to explore various grid partition configurations to understand how grid uncertainties behave under different partition schemes. In each standard case, the load buses within the same partition share an identical distribution, implying they are entirely correlated. However, the distributions of load buses across different partitions are considered i.i.d.. The partitions where obtained applying the K-Means algorithm as explained in section 6.3 and implemented with code B.3.

The standard configurations are as follows:

1. Grid with 100 partitions ($n_{\boldsymbol{\theta}} = 100$).

2. Grid with 200 partitions ($n_{\boldsymbol{\theta}} = 200$).

3. Grid with 300 partitions ($n_{\boldsymbol{\theta}} = 400$).

4. Grid with 500 partitions ($n_{\boldsymbol{\theta}} = 500$).

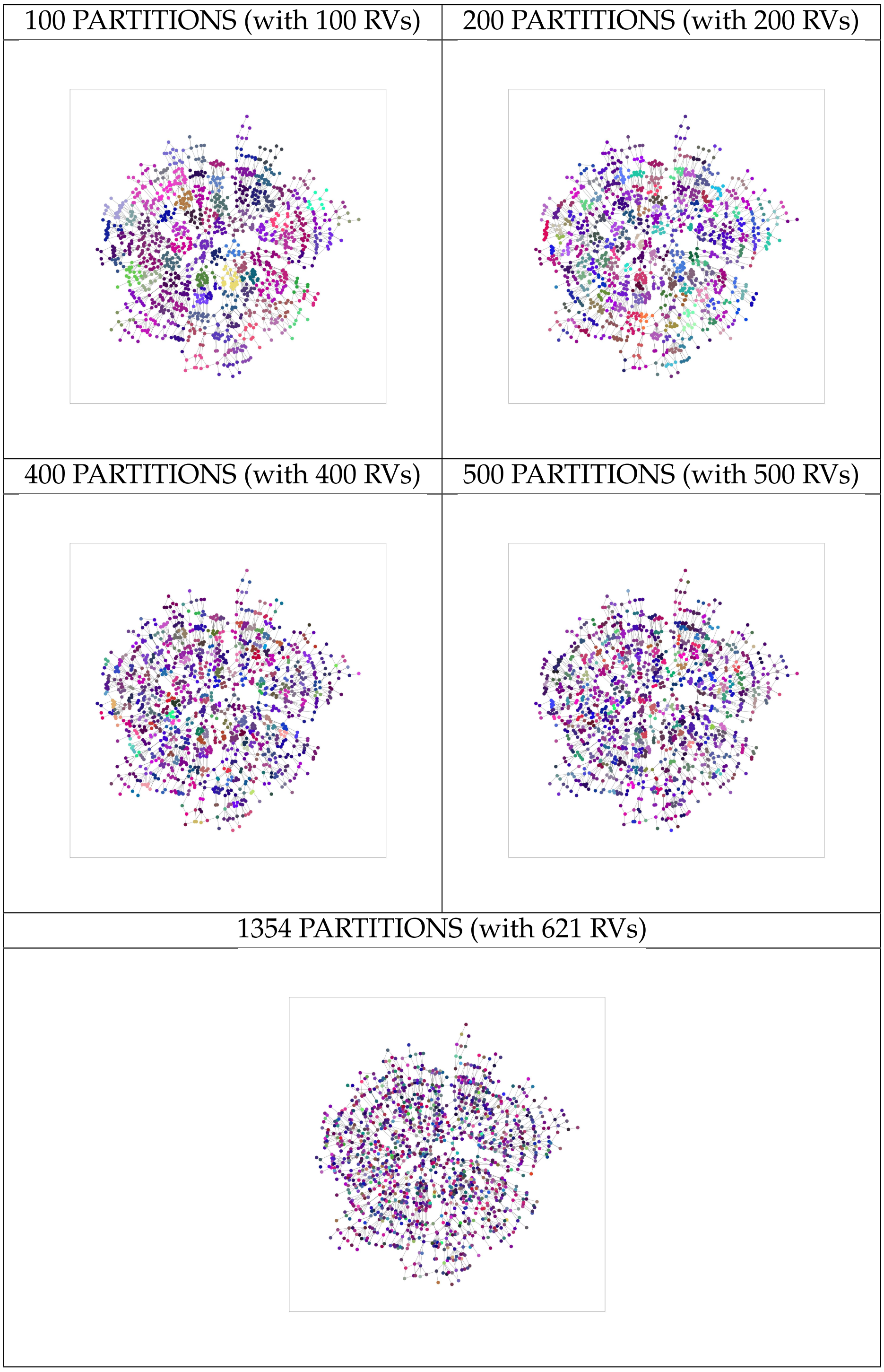
5. Grid with 1354 partitions and $n_{\boldsymbol{\theta}} = 621$: Each load bus of the grid is an independent random variable.

It is crucial to emphasize that the fluctuations (random variables) are considered geographically dependent, being completely correlated within a given partition and utterly independent between different partitions.

We randomly created uniform distributions for each partition with load buses in this study. These uniform distributions were then used to generate (pseudo-)random samples through MC simulations, as described in Section 3.3. This approach was adopted to investigate the behavior of the grid *1354pegase* under different uncertainty scenarios. In practical applications involving real-world power grids, the distributions of random fluctuations can be determined based on experimental data and/or expert professional opinion, from which MC samples can subsequently be generated.

7.2.1. Standard Cases - Partitions

Figure 7.2.: Grid partitions.



100 PARTITIONS (with 100 RVs)

200 PARTITIONS (with 200 RVs)

400 PARTITIONS (with 400 RVs)

500 PARTITIONS (with 500 RVs)

1354 PARTITIONS (with 621 RVs)

Our standard cases are presented in Figure 7.2, where each partition is represented by a distinct color. In each case, all load buses within a partition share the same fluctuation random distribution. Figure 7.2 illustrates the grid *1354pegase* divided into 100, 200, 300, 500, and 1354 partitions. To partition the grid, we consider all buses: load buses and non-load buses.

7.2.2. MC Convergence

We conducted Monte Carlo simulations with 10^4 samples for each standard case. The decision to use this number of samples was informed by the Gelman-Rubin statistics [11], [48], [64], which allows us to monitor the convergence of the MC simulations based on the sample size. The Gelman-Rubin statistics can be calculated as follows:

Mean value of chain $\mathbf{Y}_r$ (r-th row of the matrix $\mathbf{Y} \in \mathbb{R}^{n_{\boldsymbol{Y}} \times N_{\text{samples}}}$):

$$\bar{\mathbf{Y}}_r = \frac{1}{N_{\text{samples}}} \sum_{i=1}^{N_{\text{samples}}} \mathbf{Y}_r^{(i)} \ , \ \forall r = 1, \ldots, n_{\boldsymbol{Y}}. \tag{7.3}$$

Mean of the means of all chains:

$$\bar{\mathbf{Y}}_* = \frac{1}{n_{\boldsymbol{Y}}} \sum_{r=1}^{n_{\boldsymbol{Y}}} \bar{\mathbf{Y}}_r. \tag{7.4}$$

Variance of the means of the chains:

$$B = \frac{N_{\text{samples}}}{n_{\boldsymbol{Y}} - 1} \sum_{j=1}^{n_{\boldsymbol{Y}}} \left(\bar{\mathbf{Y}}_j - \bar{\mathbf{Y}}_*\right)^2. \tag{7.5}$$

Averaged variance of the chains across all chains:

$$W = \frac{1}{n_{\boldsymbol{Y}}} \sum_{j=1}^{n_{\boldsymbol{Y}}} \left(\frac{1}{N_{\text{samples}} - 1} \sum_{i=1}^{N_{\text{samples}}} \left(\mathbf{Y}_i^{(j)} - \bar{\mathbf{Y}}_j\right)^2 \right). \tag{7.6}$$

An estimate of the Gelman-Rubin statistic R then results as

$$R = \frac{\frac{(N_{\text{samples}}-1)}{N_{\text{samples}}} W + \frac{1}{N_{\text{samples}}} B}{W}. \tag{7.7}$$

We can then say that the results of the MC simulations are convergent if the estimator R tends to 1 when N_{samples} tends to infinity. For our study case with 621 random variables, we obtained the following results for the Gelman-Rubin statistics:

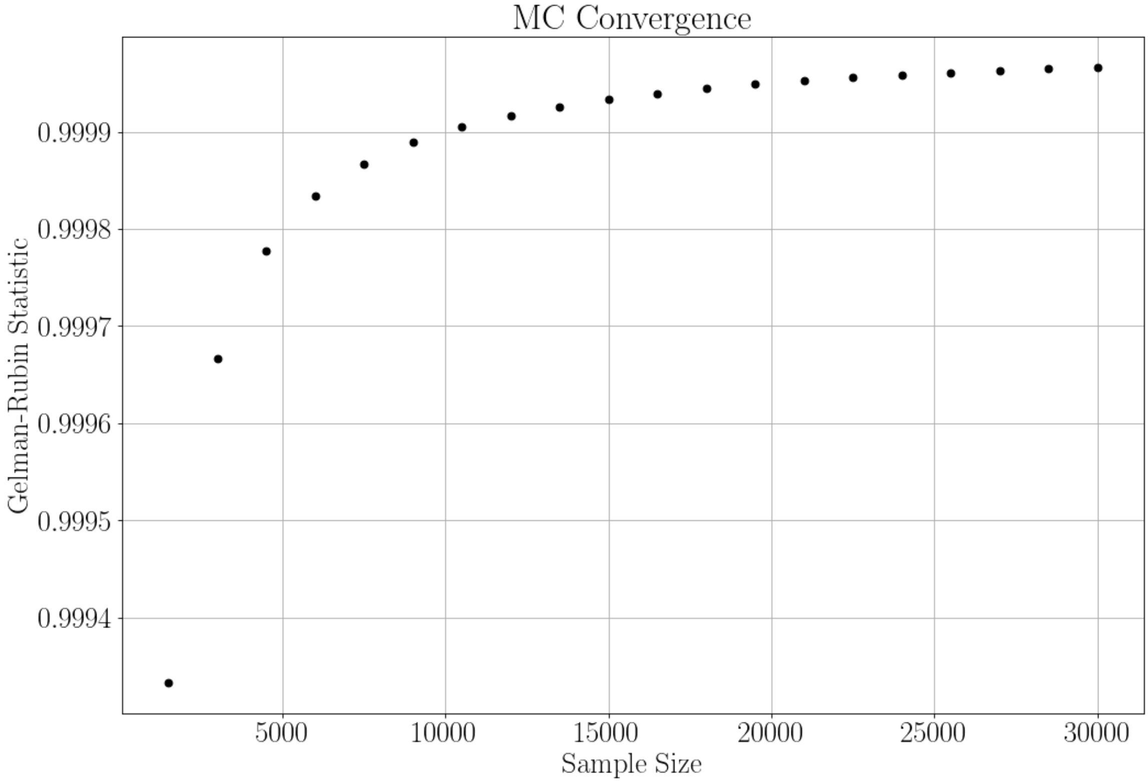


Figure 7.3.: Gelman-Rubin statistics.

From the graph in Figure 7.3, we observe that the rate of change of R decreases monotonically and tends to 0 (with R tending to 1), as the number of samples tend to infinity. From the sample size $\sim 10^4$ onwards, we obtain a value for $R \geq 0.9999$, sufficiently close to 1. This indicates that the MC calculations have good convergence for our case around this sample size and onwards. The MC calculations of the probabilities that critical buses operate outside their voltage range are used as a benchmark for the probability values calculated with PCE and with the Efficient Pipeline (section 7.6).

7.3. Random Fluctuations

The marginal PDFs of the random fluctuation $\boldsymbol{\theta}$ were defined as Uniform distributions with limits $(-a, a)$ (critical scenarios), where a was randomly chosen for each marginal distribution in $\boldsymbol{\theta}$ such that $1 < |a| \leq 2$. This approach ensures variability in the range of fluctuations, reflecting realistic uncertainty levels.

Ultimately, a joint PDF of $\boldsymbol{\theta}$ was created using the OpenTURNS library (see Section A.2). MC samples from the joint distribution were then generated to facilitate running the forward UQ pipeline as outlined in Algorithm 1. This procedure allowed for a comprehensive and accurate analysis of the grid's behavior under the defined uncertainty scenarios and is our benchmark procedure to compare the results calculated with PCE and with the proposed Efficient Pipeline.

7.4. Computations with PCE Alone

Truncation strategies for the generalized Polynomial Chaos (gPC) expansion generally rely on a predetermined polynomial degree p. However, no universal guidelines have been established on how to accurately determine the degree p, which is problem-specific. Some authors suggest that a gPC expansion with $p = 2$ typically yields accurate estimates for the first two statistical moments of a stochastic response [38]. Although this assertion lacks comprehensive validation across diverse problem sets, it can be supported by the fact that the rectangular PF formulation results in quadratic nonlinearities.

For training our PCE cases, that is, to obtain the coefficients as illustrated in the schemes from Section 4.2, we used a minimum training data set of size 1500. After training, we generated a new MC sampling of size 60000 from the corresponding standard case's stochastic input PDF to use as inputs to evaluate and predict our PCE model $\mathbf{Y} = \mathcal{P}(\boldsymbol{\theta})$.

For calculations with PCE alone (different of the case that we use PCE in the Efficient Pipeline's steps), we used polynomials with a degree at most $p = 1$. This choice was made due to the size of our stochastic spaces, which ranged from 100 to 621 dimensions. With $p = 1$, the number of truncated polynomial basis functions is given by:

$$L + 1 = \binom{n_{\boldsymbol{\theta}} + 1}{1} = \frac{(n_{\boldsymbol{\theta}} + 1)!}{n_{\boldsymbol{\theta}}!1!} = n_{\boldsymbol{\theta}} + 1.$$

This means the number of basis functions is of the same order as the dimension size of our stochastic input. For comparison, if we had chosen $p = 2$, the number of basis functions would be:

$$L + 1 = \binom{n_{\boldsymbol{\theta}} + 2}{2} = \frac{(n_{\boldsymbol{\theta}} + 2)!}{n_{\boldsymbol{\theta}}!2!} = \frac{(n_{\boldsymbol{\theta}})^2 + 3n_{\boldsymbol{\theta}} + 2}{2}.$$

This would result in a significantly larger number of basis functions, leading to high computational intensity, especially for our random variables' dimension order range.

To compare the difference in computation time when using PCE with degrees 1 and 2, we ran simulations using the total number of input RVs for each of the considered cases. However, the simulation could not be completed for the case with 621 RVs and $p = 2$ due to a *computational memory error*. Specifically, for the case with 621 RVs and $p = 2$, it is necessary to create $193,753$ basis functions, each multiplied by the dimension of our stochastic output $n_{\boldsymbol{Y}}$, which in our case equals the total number of buses $n_{\boldsymbol{Y}} = 1354$ in our grid, as we consider only the voltage magnitudes $\mathbf{v}$. Even considering the sparsity of the coefficients, which reduces the computational dimension, the number of function evaluations is prohibitive.

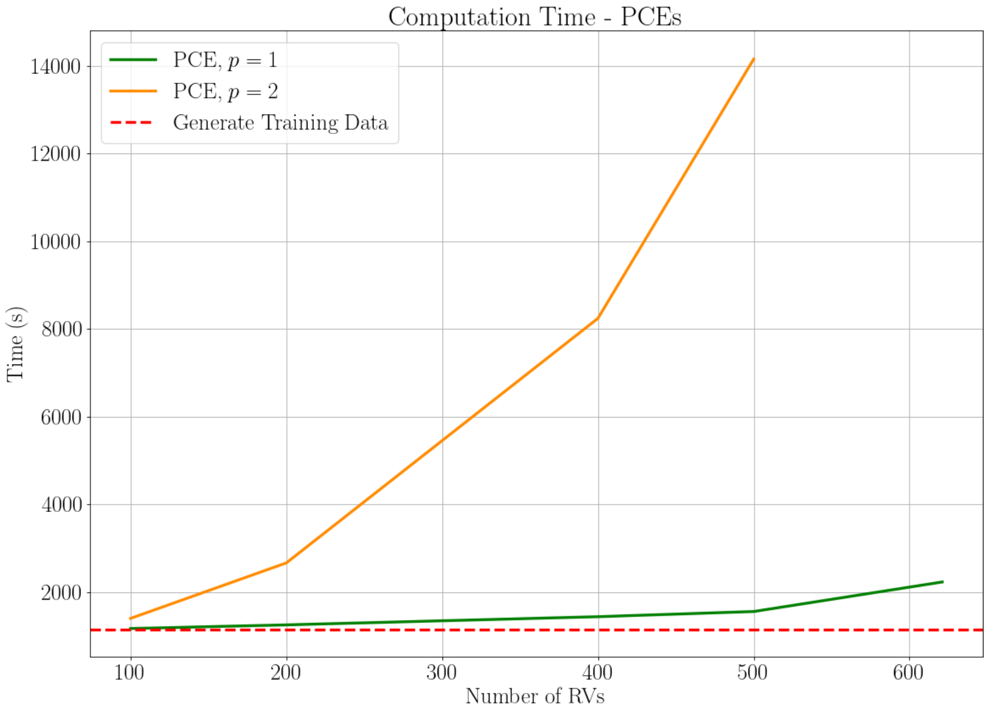


Figure 7.4.: Computation time between PCEs for $p = 1, 2$.

Figure 7.4 compares computation time and illustrates how each PCE case behaves. For all cases in this study, we consider computation time as the time interval between the start of generating the training data for our PCE surrogate models and obtaining all the PPF results, that is, the variables $\mathbf{x}$ of our PF system. It is important to note that the computation times obtained from our simulations are not purely analytical but CPU times of our numerical models. These models leverage advanced numerical techniques, such as matrix decomposition and sparsity, to solve the problem efficiently.

For $p = 1$, we observe that the computation time increases monotonically and at a slower rate than for $p = 2$, where the computation time increases non-linearly.

7.5. Computations with MC and PCE

Figure 7.5.: Comparison between *means* of each bus from MC and PCE results.

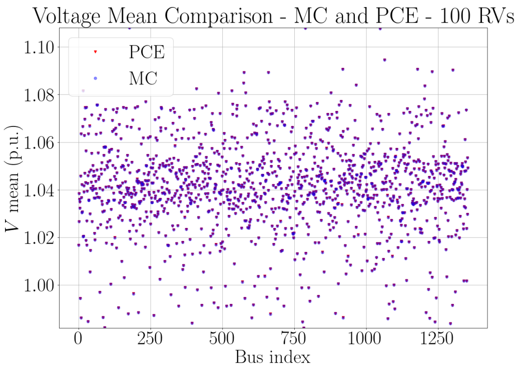
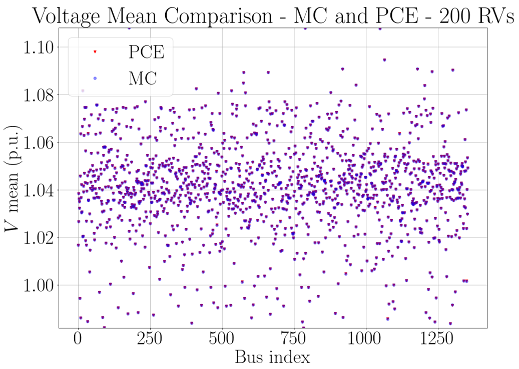
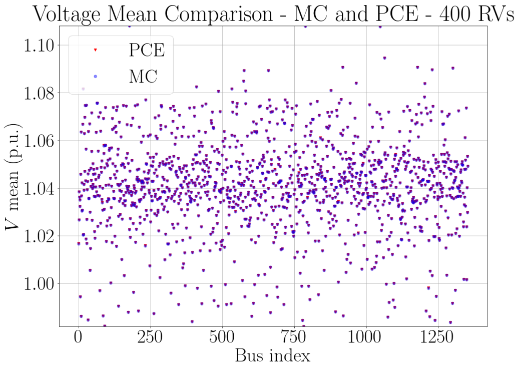
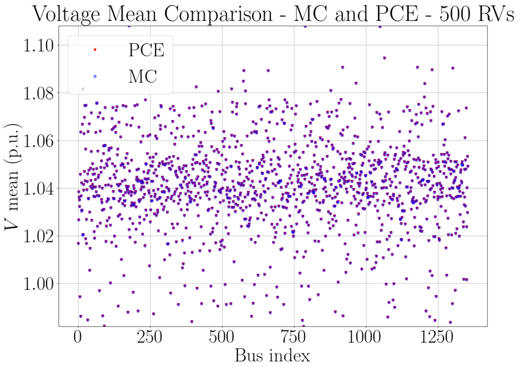
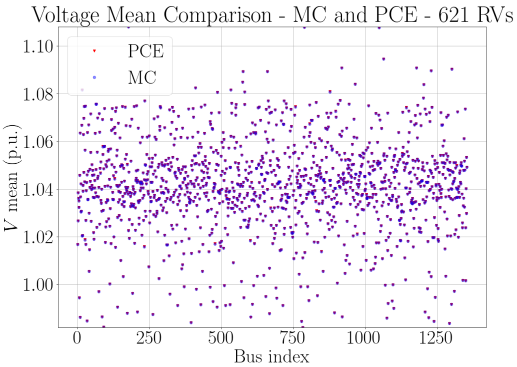
<table>
<tr><th>100 PARTITIONS</th><th>200 PARTITIONS</th></tr>
<tr><td> </td><td> </td></tr>
<tr><th>400 PARTITIONS</th><th>500 PARTITIONS</th></tr>
<tr><td> </td><td> </td></tr>
<tr><th colspan="2">1354 PARTITIONS (621 RVs)</th></tr>
<tr><td colspan="2"> </td></tr>
</table>

Figure 7.6.: Comparison between *standard deviations* of each bus from MC and PCE results.

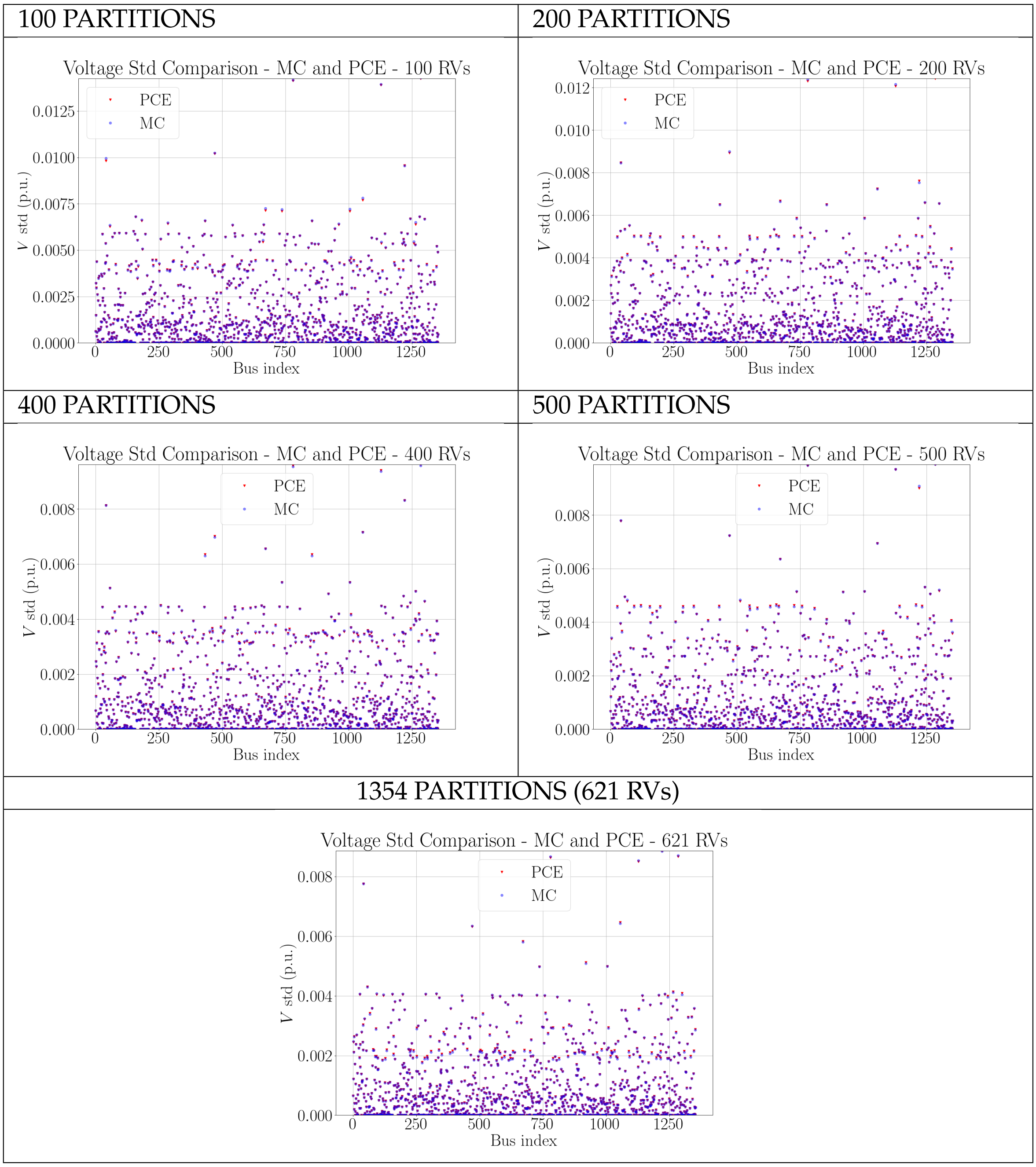
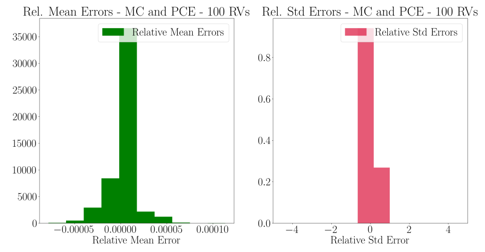
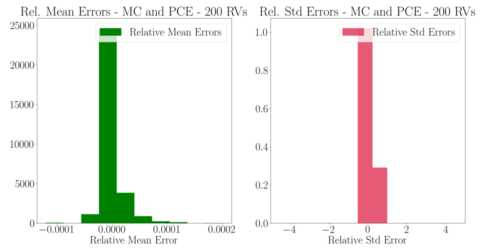
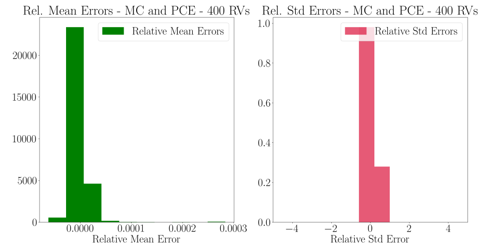
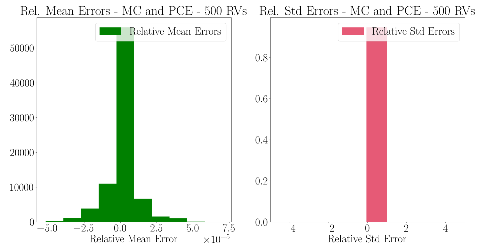
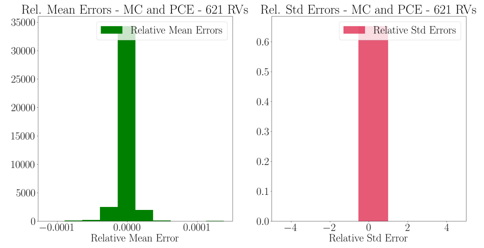


Figure 7.7.: Histogram of relative errors.

<table>
<tr><th>100 PARTITIONS</th><th>200 PARTITIONS</th></tr>
<tr><td> </td><td> </td></tr>
<tr><th>400 PARTITIONS</th><th>500 PARTITIONS</th></tr>
<tr><td> </td><td> </td></tr>
<tr><th colspan="2">1354 PARTITIONS (621 RVs)</th></tr>
<tr><td colspan="2"> </td></tr>
</table>

To evaluate the accuracy of PCE relative to MC simulations for the same grid cases, we compared both the *mean* and *standard deviation* of the results for each bus. These comparisons are shown in Figures 7.5 and 7.6.

Moreover, the histograms of the relative errors of the mean and standard deviation between the results obtained from MC and PCE are given in Figure 7.7 for all standard cases analyzed.

We observe that the results obtained using PCE are accurate for all standard cases compared to those from MC simulations. This attests to PCE as a precise method for performing UQ computations in our standard cases.

7.6. The Efficient Pipeline

In this work, we developed an efficient pipeline to perform PPF analysis and to calculate the probability of buses operating outside the allowed voltage magnitude range in systems characterized by high-dimensional stochastic input random variables. Surrogate modeling with PCE is employed in two of the six pipeline steps to enhance the pipeline's efficiency. For all PCEs considered in these steps, we set the degree to $p = 2$, because the used stochastic inputs are reduced, which will not increase the computation time significantly, and the obtained results are more accurate than when we try with $p = 1$.

The initial training data set comprises 1500 samples ($N_{\text{total-samples}} = 1500$). For the PCE prediction, we utilized MC sampling from the reduced stochastic input created in a respective step, with a size of $40 \times N_{\text{step-sample}}$, where $N_{\text{step-sample}}$ represents the sample size used for training a PCE at a specific step of the pipeline.

The pipeline is composed of the following six steps:

1. **Generate Training Data Set**

   - Generate a small training data set of size $N_{\text{total-samples}}$ with the respective standard case to be computed.

2. **Partition Grid and Identify Critical Buses**

   - Partition the standard case grid into four sections using the K-Means algorithm (see Code B.3).

   - Assign to each new partition a uniform random variable with new limits $(-a, a)$, where $|a|$ is the mean of former distribution limits in that partition area, such that we obtain a new reduced stochastic input $\boldsymbol{\theta}$ with $n_{\boldsymbol{\theta}} = 4$.

   - Train a new PCE surrogate model giving as training inputs a small sample fraction ($N_{\text{step-sample}} \approx 10 \times n_{\boldsymbol{\theta}} = 10 \times 4 = 40$) of the training data set from Step 1.

   - Evaluate/Predict the trained PCE surrogate model $\mathbf{Y} = \mathcal{P}(\boldsymbol{\theta})$ with the realizations $\{\boldsymbol{\theta}^{(i)}\}_{i=1}^{40 \times N_{\text{step-sample}}}$ using MC sampling of the new reduced stochastic input (with $n_{\boldsymbol{\theta}} = 4$).

   - Identify the buses that work out of their voltage magnitude limits with the results $\mathbf{Y}$ of the PCE prediction.

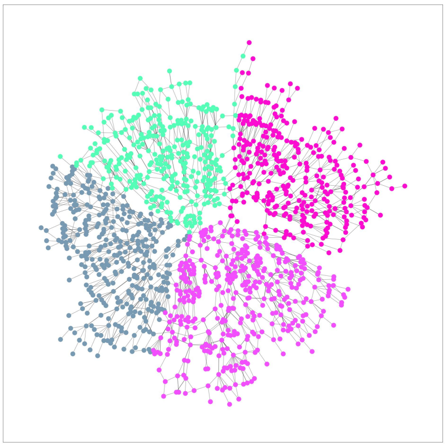


Figure 7.8.: Step 2 – 4 Partitions.

3. **Sensitivity Analysis with Rank-based Estimation**
    - Select the identified critical buses of the previous step.
    - Perform rank-based estimation sensitivity analysis using only the selected critical buses and the original input training data set with $n_{\boldsymbol{\theta}}$ =*original_number_of_RVs* (see Code B.1).
4. **Identify Influential Load Buses (marginal RVs)**
    - Identify and get the load buses (marginal RVs) from Step 3 that genuinely influence the critical buses given by considering the non-zero first order Sobol's indices.
5. **Run Forward UQ with Selected Inputs**
    - Train a new PCE surrogate model giving the entire samples of the training data set created in Step 1 with $N_{\text{step-sample}} = N_{\text{total-samples}} = 1500$, but only with the stochastic inputs identified as influential in Step 3.
    - Evaluate/Predict the trained new PCE surrogate model.
6. **Calculate Probabilities for Critical Buses**
    - Determine the probabilities that the critical buses will operate outside the voltage magnitude range using the results from the previous step.

As a result, we obtained the computation time advantage shown in Figure 7.9.

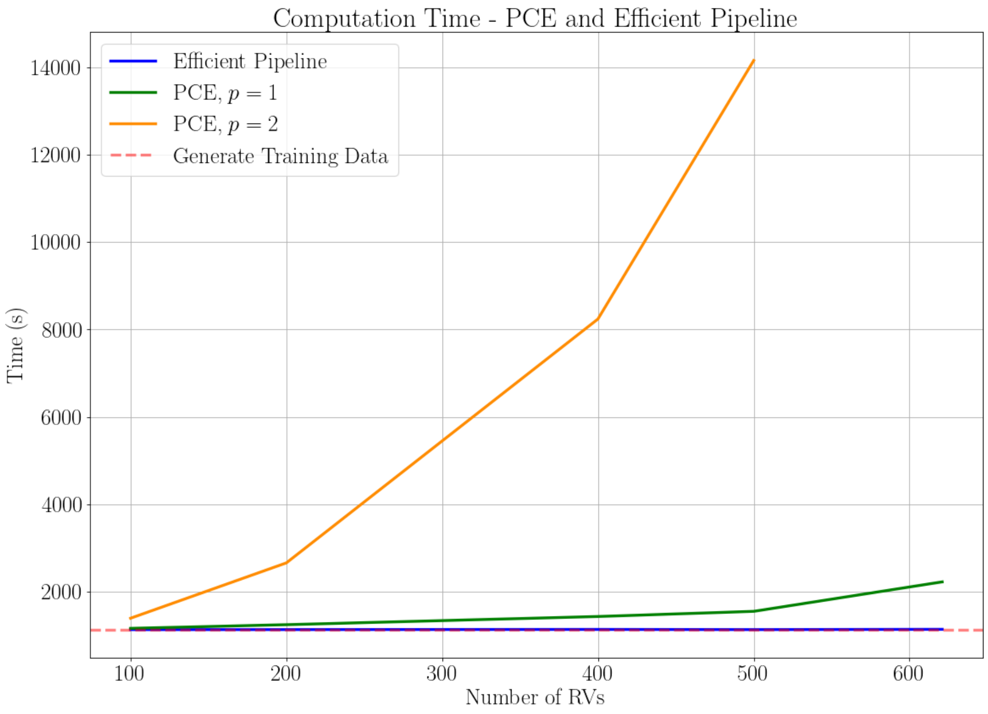


Figure 7.9.: Computation time between Efficient Pipeline and PCEs alone.

From Figure 7.9, we observe that for a given training size, the computation time of the efficient pipeline does not scale up when the number of RVs increases. This behavior contrasts with the performance of using PCE alone, where computation time increases with the number of RVs. This characteristic endows the efficient pipeline with a significant advantage for calculating systems with high-dimensional stochastic spaces, making it a more practical and advantageous approach than using PCE alone.

The efficient pipeline is particularly competitive for performing PPF analysis of large power grids, such as those of mid-sized cities or networks of connected small cities within a regional ring. This makes it a tool for analyzing complex power systems where traditional methods would become computationally prohibitive.

The main reason for the efficient pipeline's computation time not scaling up with the increasing number of RVs lies in step 3, **Sensitivity Analysis with Rank-based Estimation**. In this step, we reduce the number of input random variables to only those genuinely influencing the critical buses. This reduction allows for a significantly smaller stochastic input space in step 5, **Run Forward UQ with Selected Inputs**. By focusing only on the relevant RVs, the efficient pipeline can maintain a manageable computation time even as the potential number of RVs increases, ensuring an optimized and efficient calculation process.

The flowchart in Figure 7.10 shows the efficient pipeline in a compact form.

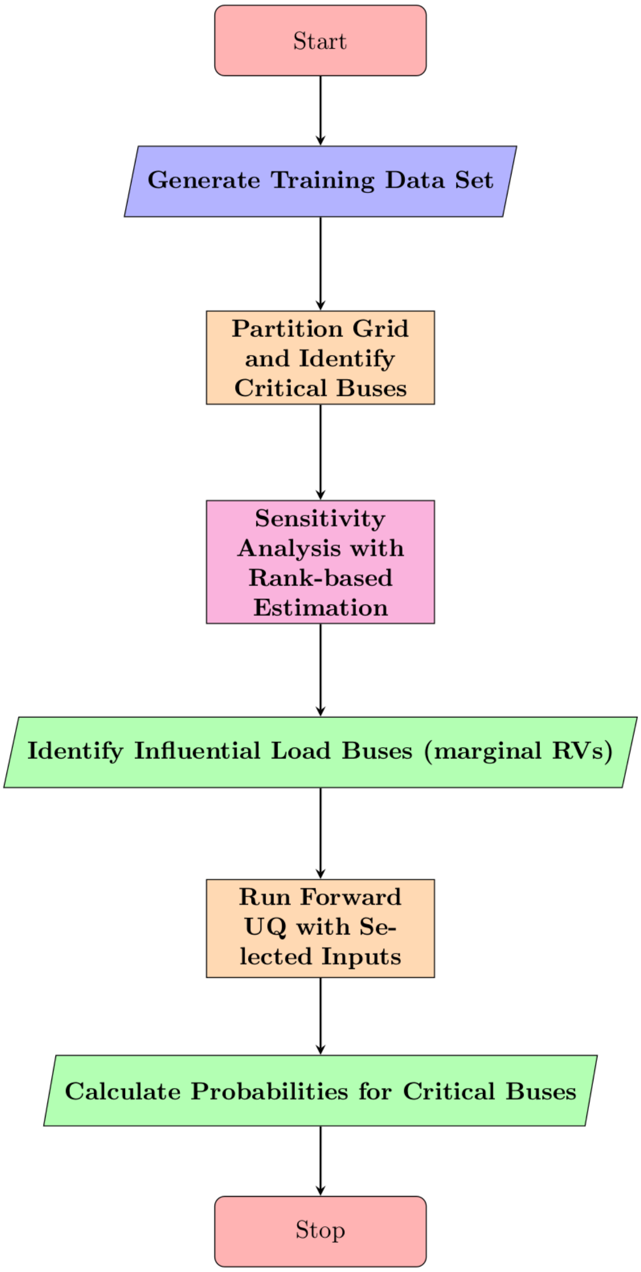

Start
Generate Training Data Set
Partition Grid and Identify Critical Buses
Sensitivity Analysis with Rank-based Estimation
Identify Influential Load Buses (marginal RVs)
Run Forward UQ with Selected Inputs
Calculate Probabilities for Critical Buses
Stop

Figure 7.10.: Flowchart of Efficient Pipeline.

Part III.

Results and Conclusion

8. Results

In this chapter, we present and compare the results of our case study using the *1354pegase* power network for all considered standard cases.

Section 8.1 shows the probability values of all identified critical buses operating outside their specified voltage magnitude range, calculated with different approaches (MC, Efficient Pipeline, and PCE alone), and the compared histograms of voltage magnitudes from the results of the respective calculation approaches for the respective critical buses.

Section 8.2 presents the critical buses and their most influential load buses, which are given by the rank-based estimation sensitivity analysis, on the grid map.

Finally, in Section 8.3, we compare the computation times of all computation schemes used.

8.1. Failure Risk of Critical Buses

The results for the probabilities of the critical buses obtained using the Efficient Pipeline, PCE, and MC simulations are presented for all standard cases from Subsection 8.1.1 to 8.1.5.

The way the critical buses are identified, and the risk of failure probabilities are calculated is the following:

- For MC: we count the number of voltage magnitudes from the MC simulations that exceed the voltage limits by iterating over the result array. We can then divide this number by the total result array size to obtain the probabilities/risk of failure in a frequentist fashion.

- For PCE and the Efficient Pipeline: by generating a new and large sample from the considered input PDF and then inputting these samples into the trained PCE surrogate model $\mathbf{Y} = \mathcal{P}(\boldsymbol{\theta})$, we obtain the predicted results. We can then identify the critical buses and calculate their probabilities of failure in the same manner as in the MC case. The Efficient Pipeline uses PCE in Step 2 with a reduced stochastic space to identify the critical buses.

Our observations indicate that the accuracy of probability calculations using PCE and the Efficient Pipeline is highly satisfactory compared to the probability values obtained from MC simulations (our benchmark simulations). This suggests that both PCE and the proposed Efficient Pipeline provide accurate and realistic results in identifying the probabilities of critical buses operating outside their voltage magnitude range.

8.1.1. Standard Case – 100 RVs

In Figure 8.1, we show the probabilities of buses working outside their voltage magnitude range. For the standard case of 100 RVs, four buses were identified to work out of their magnitude voltage range: buses 919, 1179, 1230, and 1301. Table 8.1 provides more detailed information on the probabilities of the critical buses and the respective schemes from which they were computed.

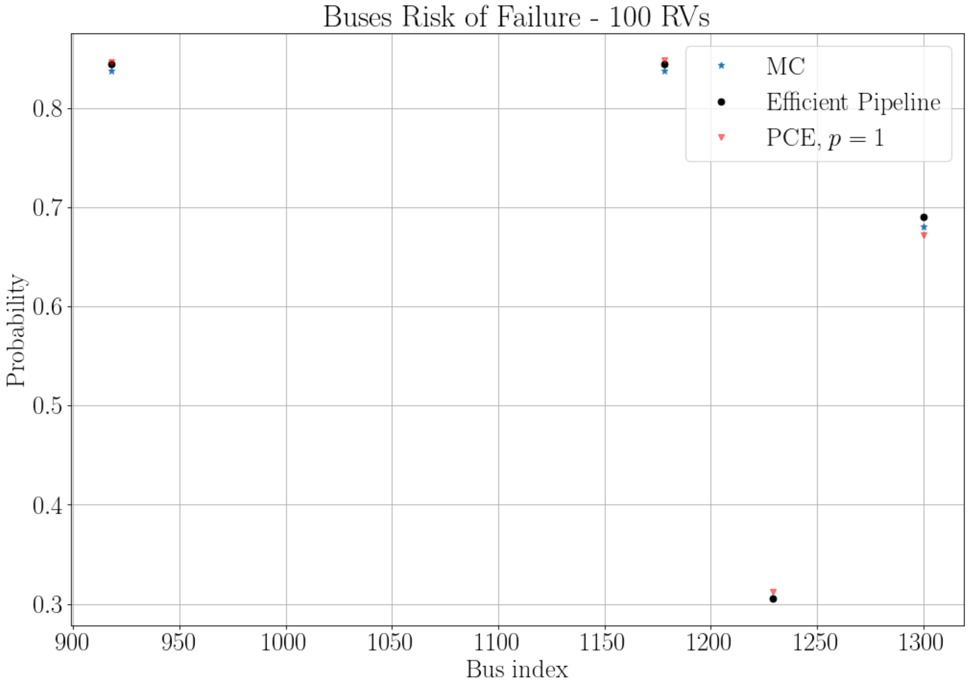


Figure 8.1.: Failure Risk of Critical Buses – Standard Case with 100 RVs.

Method name	Failure risk bus 919	Failure risk bus 1179	Failure risk bus 1230	Failure risk bus 1301	Computational cost (s)
MC	0.84	0.84	0.31	0.68	10^4
Eff. Pipeline	0.85	0.85	0.31	0.69	10^2
PCE, $p = 1$	0.85	0.85	0.31	0.67	10^2

Table 8.1.: Details on the risk of failure of critical buses.

Figures 8.2 and 8.3 show the compared histograms of the critical buses' voltage magnitudes obtained from the different calculation schemes' results. In Figure 8.2, we compare the histograms from the Efficient Pipeline and MC; in Figure 8.3, we compare the histograms from the Efficient Pipeline, PCE, and MC. The histograms are superimposed, and we used translucid colors for better visualization.

For the case with 100 RVs, we observe a strong similarity between the different histograms, what indicates our proposed Efficient Pipeline method is accurate and correct when compared to the benchmark MC simulations.

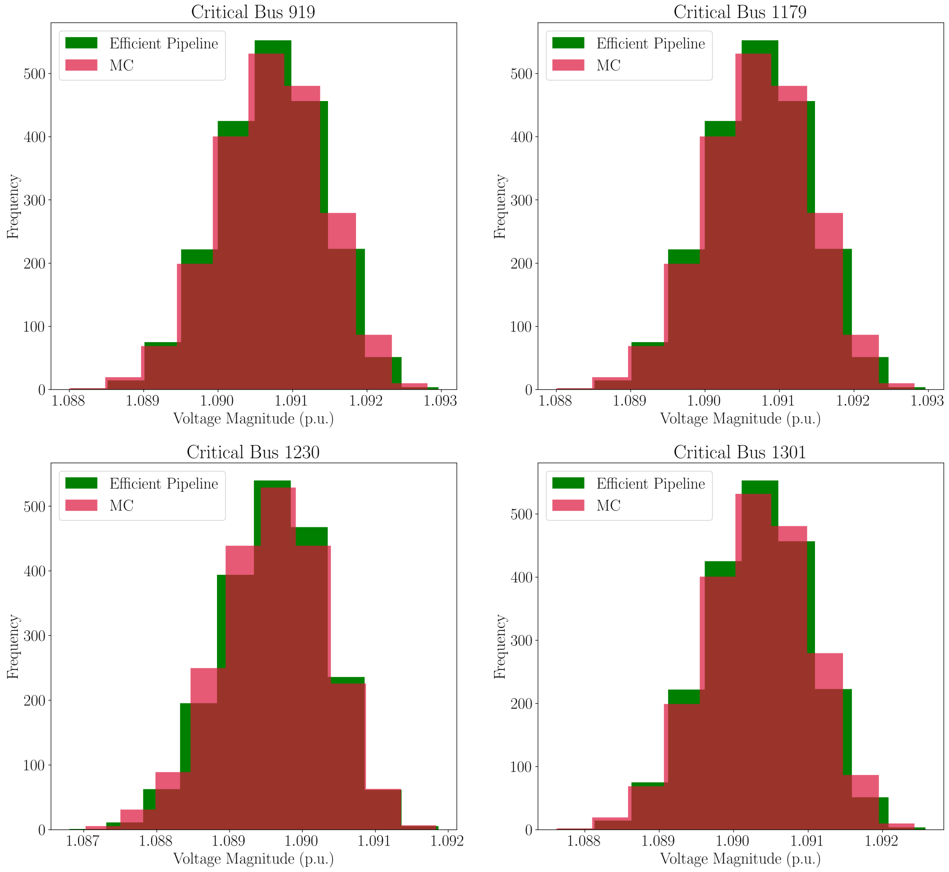


Figure 8.2.: Comparison between histograms from MC and Efficient Pipeline.

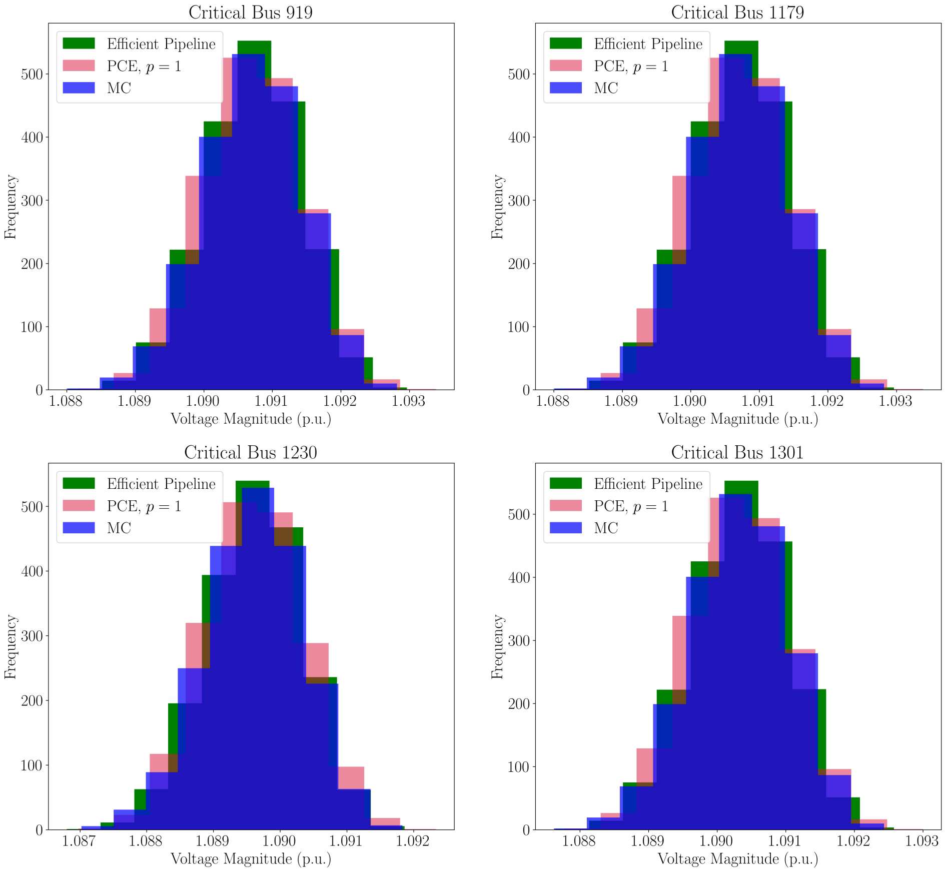


Figure 8.3.: Comparison between histograms from MC, Efficient Pipeline, and PCE.

8.1.2. Standard Case – 200 RVs

In Figure 8.4, we show the probabilities of buses working outside their voltage magnitude range. For the standard case of 200 RVs, four buses were identified to work out of their magnitude voltage range: buses 919, 1179, 1230, and 1301. Table 8.2 provides more detailed information on the probabilities of the critical buses and the respective schemes from which they were computed.

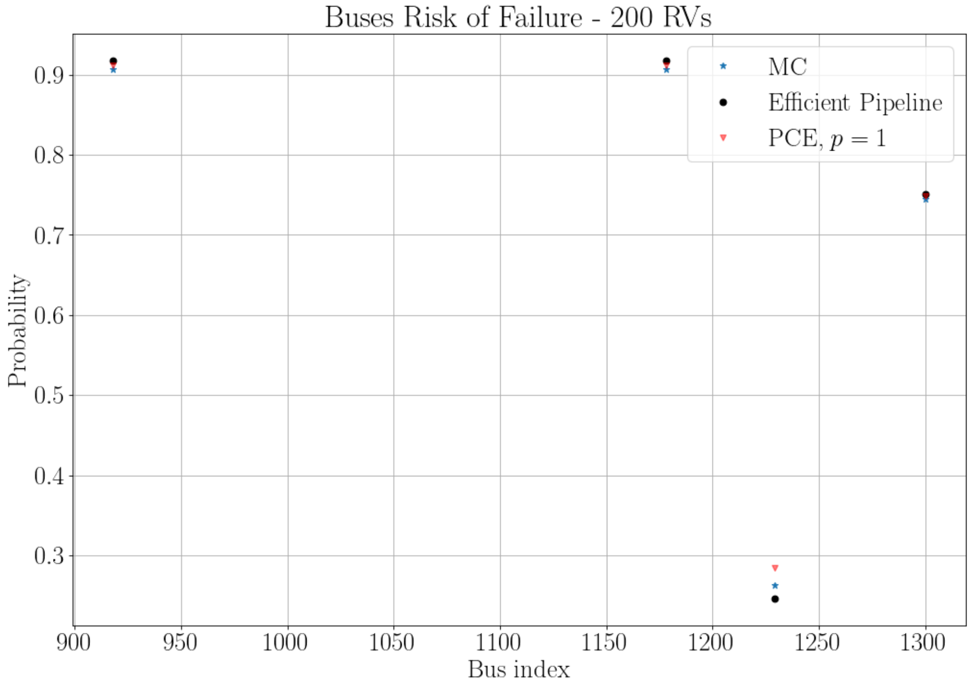


Figure 8.4.: Failure Risk of Critical Buses – Standard Case with 200 RVs.

Method name	Failure risk bus 919	Failure risk bus 1179	Failure risk bus 1230	Failure risk bus 1301	Computational cost (s)
MC	0.91	0.91	0.26	0.74	10^4
Eff. Pipeline	0.92	0.92	0.25	0.75	10^2
PCE, $p = 1$	0.91	0.91	0.28	0.75	1.3×10^3

Table 8.2.: Details on the risk of failure of critical buses.

Figures 8.5 and 8.6 show the compared histograms of the critical buses' voltage magnitudes obtained from the different calculation schemes' results. In Figure 8.5, we compare the histograms from the Efficient Pipeline and MC; in Figure 8.6, we compare the histograms from the Efficient Pipeline, PCE, and MC. The histograms are superimposed, and we used translucid colors for better visualization.

For the case with 200 RVs, we also observe a substantial similarity between the different histograms, indicating our proposed Efficient Pipeline method is accurate and correct when compared to the benchmark MC simulations.

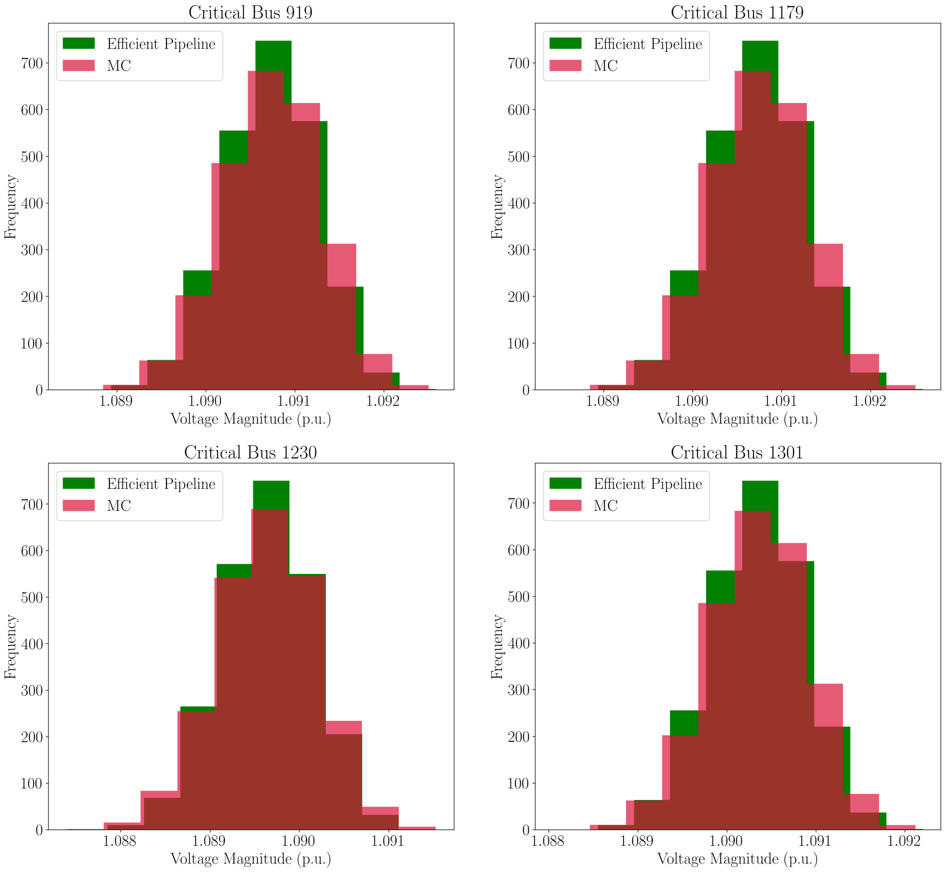


Figure 8.5.: Comparison between histograms from MC and Efficient Pipeline.

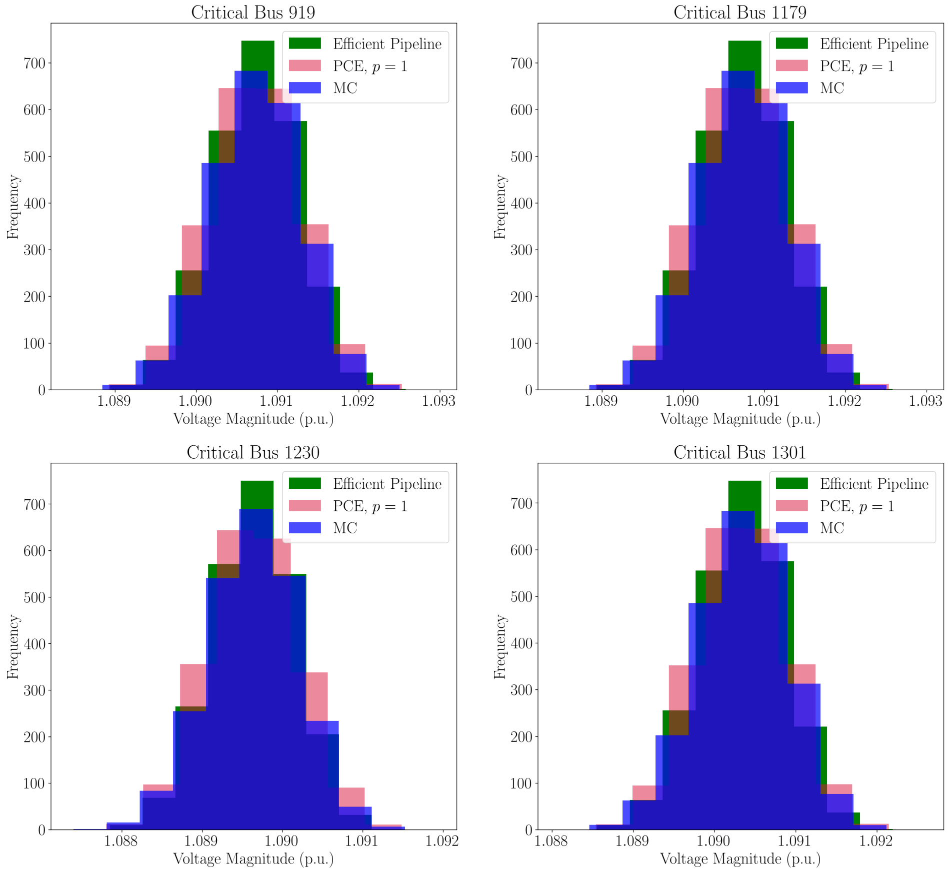


Figure 8.6.: Comparison between histograms from MC, Efficient Pipeline, and PCE.

8.1.3. Standard Case – 400 RVs

In Figure 8.7, we show the probabilities of buses working outside their voltage magnitude range. For the standard case of 400 RVs, four buses were identified to work out of their magnitude voltage range: buses 919, 1179, 1230, and 1301. Table 8.3 provides more detailed information on the probabilities of the critical buses and the respective schemes from which they were computed.

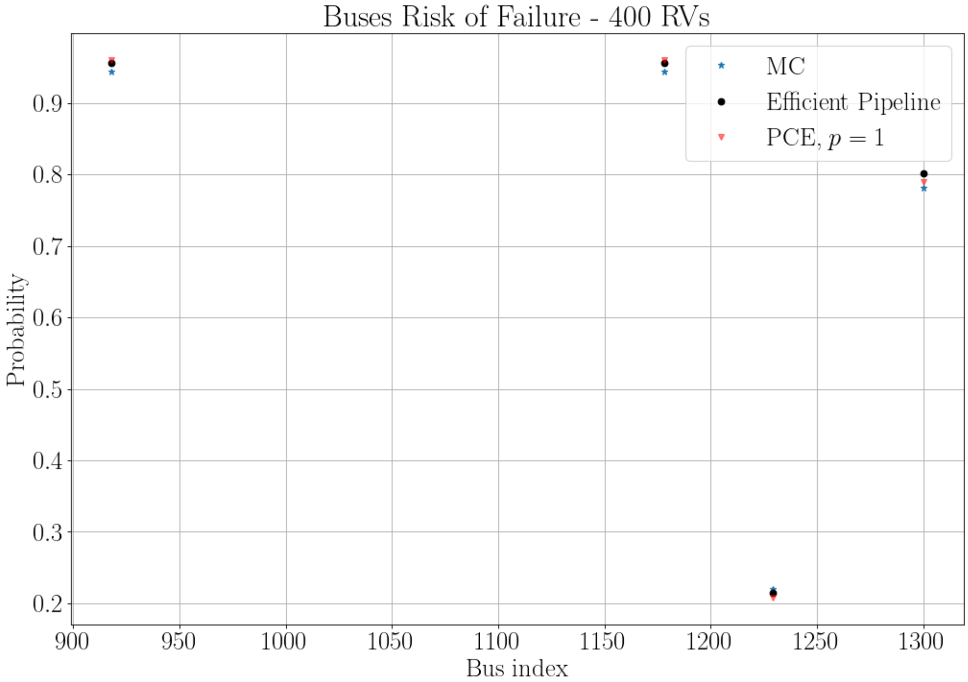


Figure 8.7.: Failure Risk of Critical Buses – Standard Case with 400 RVs.

Method name	Failure risk bus 919	Failure risk bus 1179	Failure risk bus 1230	Failure risk bus 1301	Computational cost (s)
MC	0.94	0.94	0.22	0.78	10^4
Eff. Pipeline	0.96	0.96	0.21	0.80	10^2
PCE, $p = 1$	0.96	0.96	0.21	0.79	1.4×10^3

Table 8.3.: Details on the risk of failure of critical buses.

Figures 8.8 and 8.9 show the compared histograms of the critical buses' voltage magnitudes obtained from the different calculation schemes' results. In Figure 8.8, we compare the histograms from the Efficient Pipeline and MC; in Figure 8.9, we compare the histograms from the Efficient Pipeline, PCE, and MC. The histograms are superimposed, and we used translucid colors for better visualization.

For the case with 400 RVs, we start observing that the similarities between the different histograms are lower than in the previous cases. However, the proposed Efficient Pipeline method is still accurate and correct compared to the benchmark MC simulations.

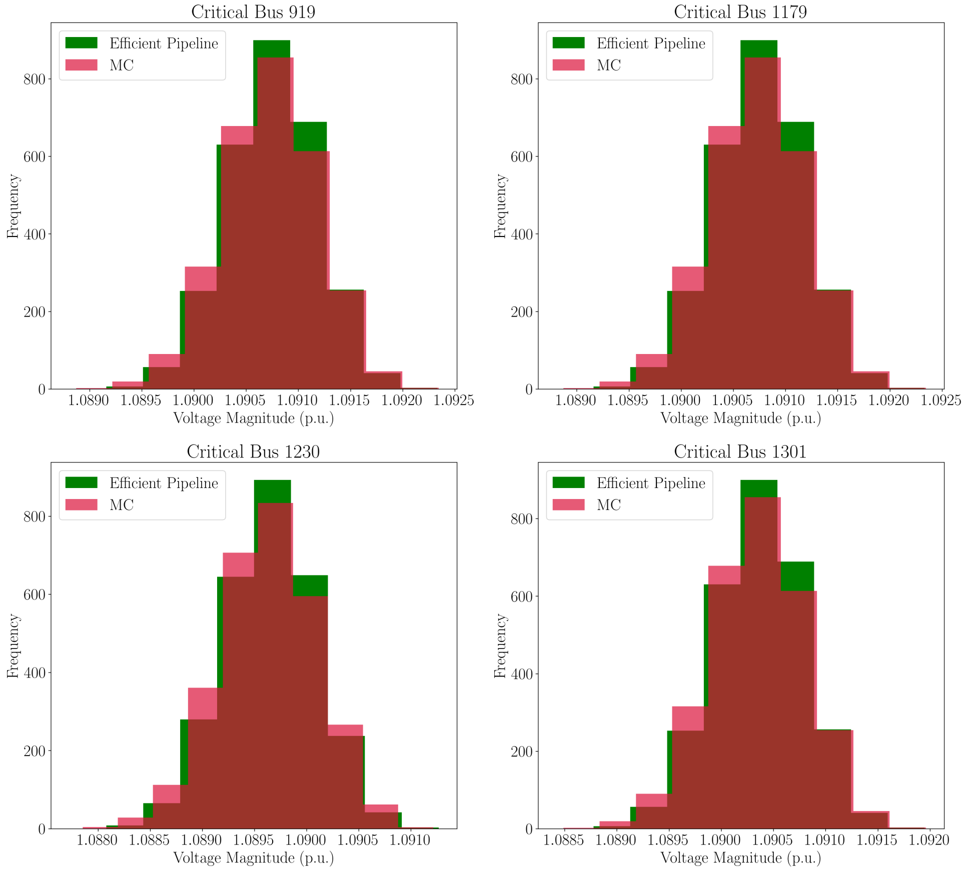


Figure 8.8.: Comparison between histograms from MC and Efficient Pipeline.

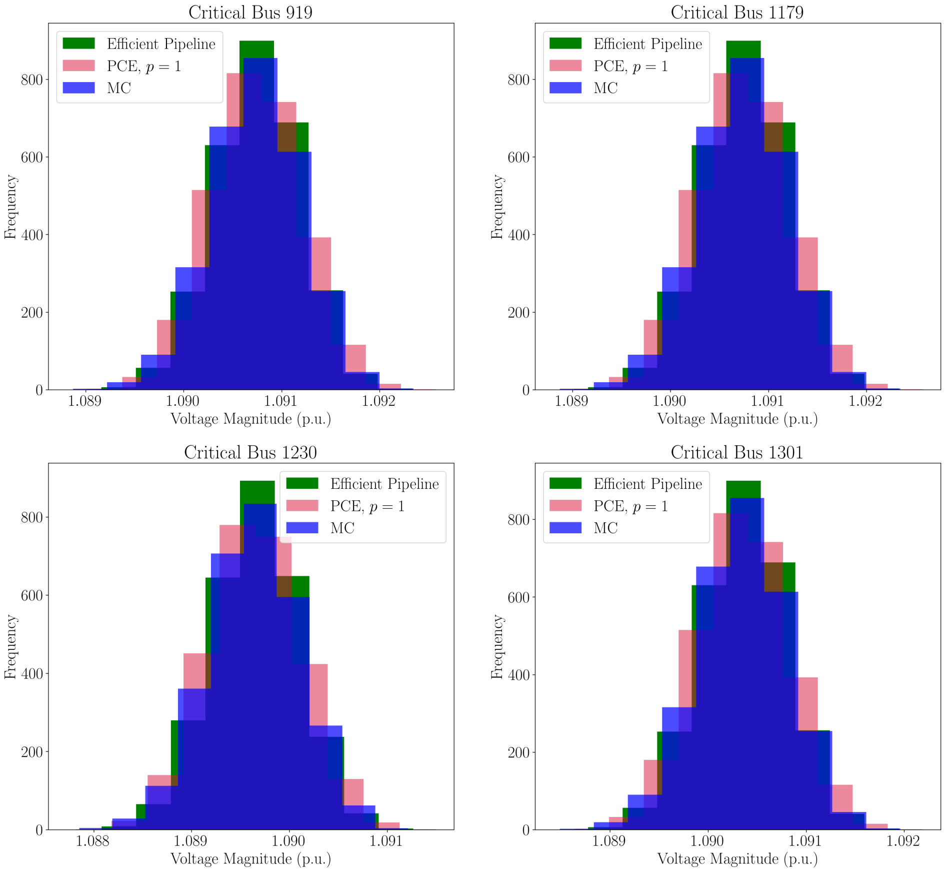


Figure 8.9.: Comparison between histograms from MC, Efficient Pipeline, and PCE.

8.1.4. Standard Case – 500 RVs

In Figure 8.10, we show the probabilities of buses working outside their voltage magnitude range. For the standard case of 500 RVs, four buses were identified to work out of their magnitude voltage range: buses 919, 1179, 1230, and 1301. Table 8.4 provides more detailed information on the probabilities of the critical buses and the respective schemes from which they were computed.

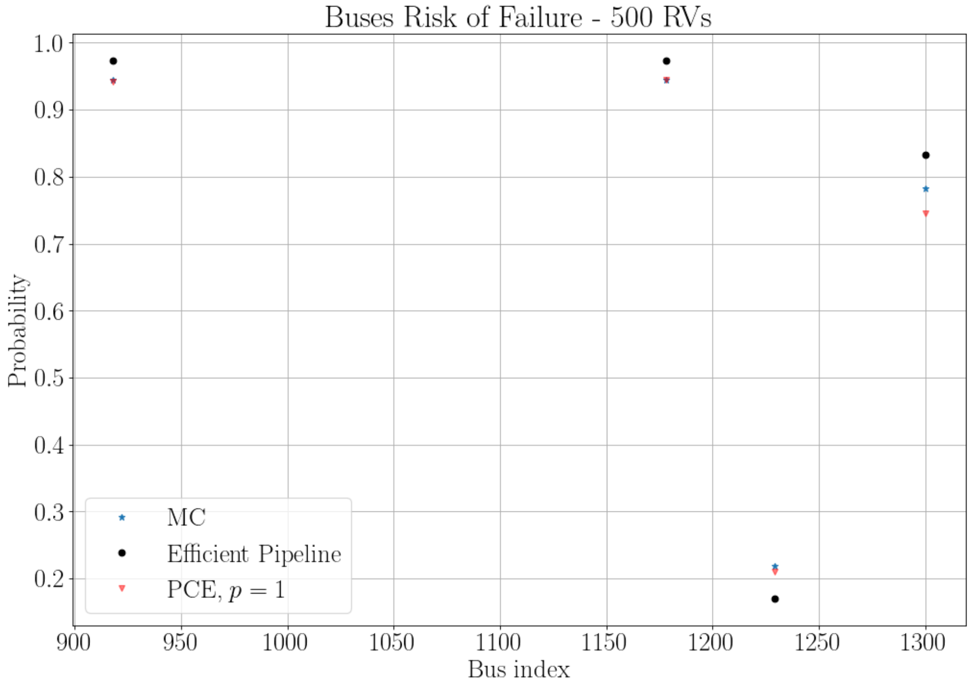


Figure 8.10.: Failure Risk of Critical Buses – Standard Case with 500 RVs.

Method name	Failure risk bus 919	Failure risk bus 1179	Failure risk bus 1230	Failure risk bus 1301	Computational cost (s)
MC	0.95	0.95	0.22	0.78	10^4
Eff. Pipeline	0.97	0.97	0.17	0.83	10^2
PCE, $p = 1$	0.94	0.94	0.21	0.75	1.6×10^3

Table 8.4.: Details on the risk of failure of critical buses.

Figures 8.11 and 8.12 show the compared histograms of the critical buses' voltage magnitudes obtained from the different calculation schemes' results. In Figure 8.11, we compare the histograms from the Efficient Pipeline and MC; in Figure 8.12, we compare the histograms from the Efficient Pipeline, PCE, and MC. The histograms are superimposed, and we used translucid colors for better visualization.

For the case with 500 RVs, we see a decrease in the similarities between the different histograms. Now, the similarities are lower than in the previous cases. This is explained by the fact that we used the same training data set size for all standard cases, even when the number of RVs increase. Nevertheless, the proposed Efficient Pipeline method shows good accuracy compared to the benchmark MC simulations.

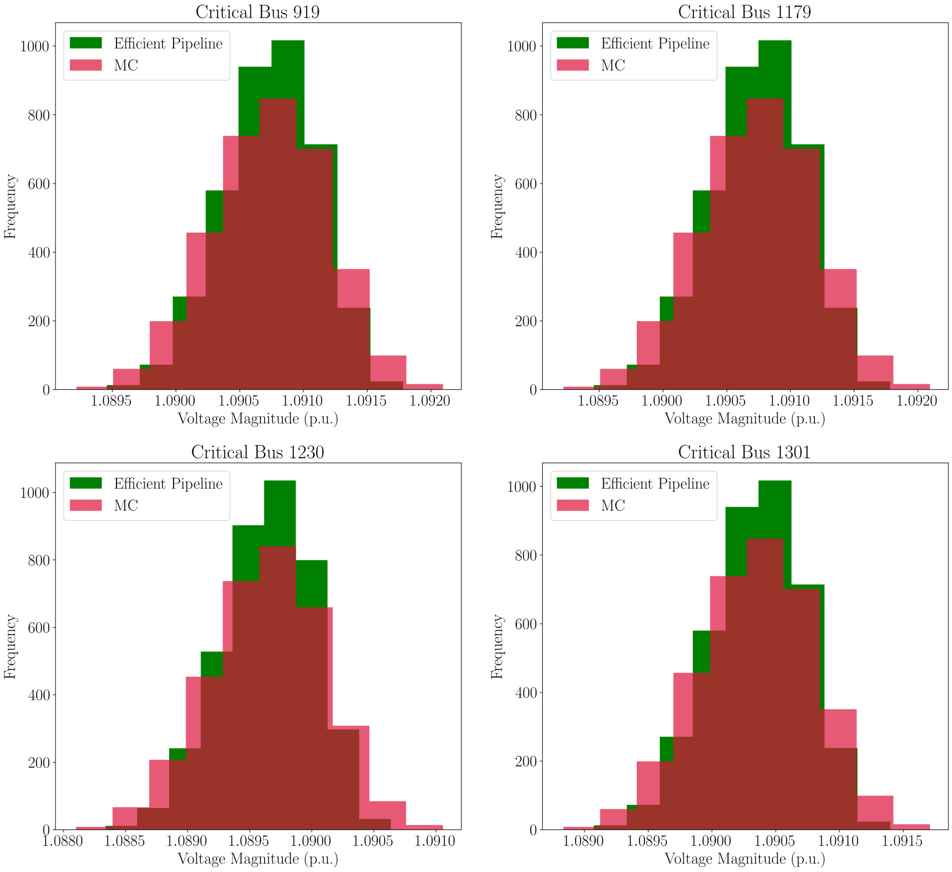


Figure 8.11.: Comparison between histograms from MC and Efficient Pipeline.

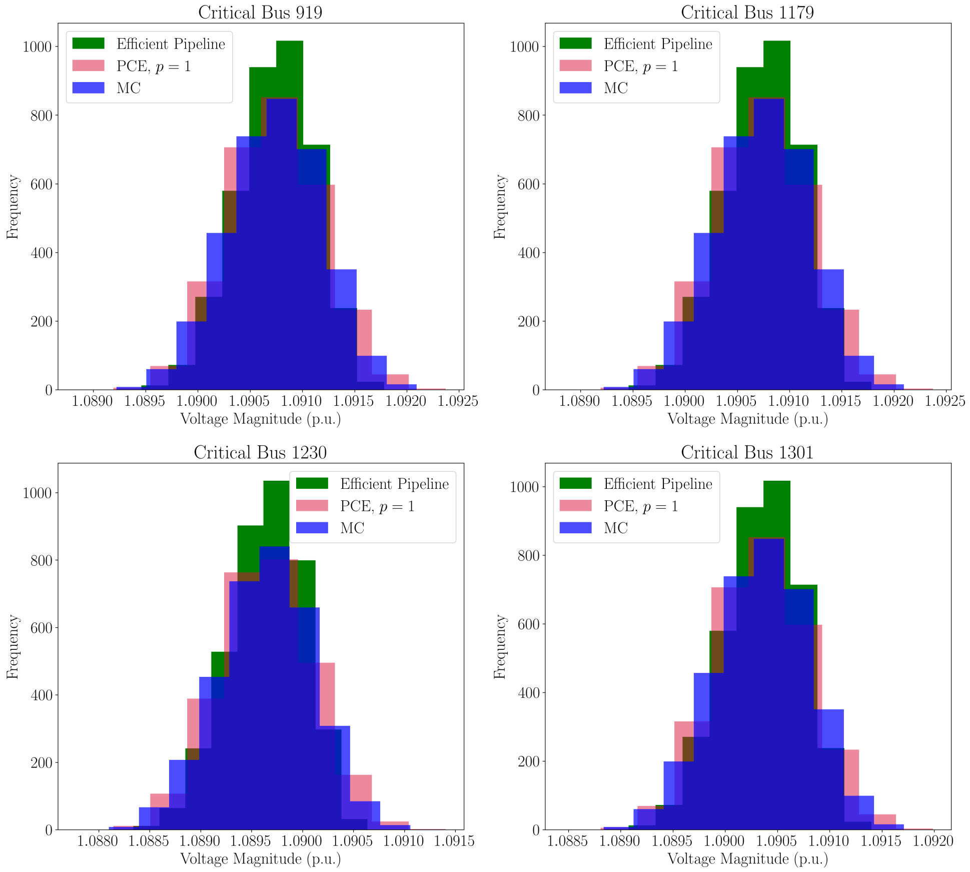


Figure 8.12.: Comparison between histograms from MC, Efficient Pipeline, and PCE.

8.1.5. Standard Case – 621 RVs

In Figure 8.13, we show the probabilities of buses working outside their voltage magnitude range. For the standard case of 621 RVs, four buses were identified to work out of their magnitude voltage range: buses 919, 1179, 1230, and 1301. Table 8.5 provides more detailed information on the probabilities of the critical buses and the respective schemes from which they were computed.

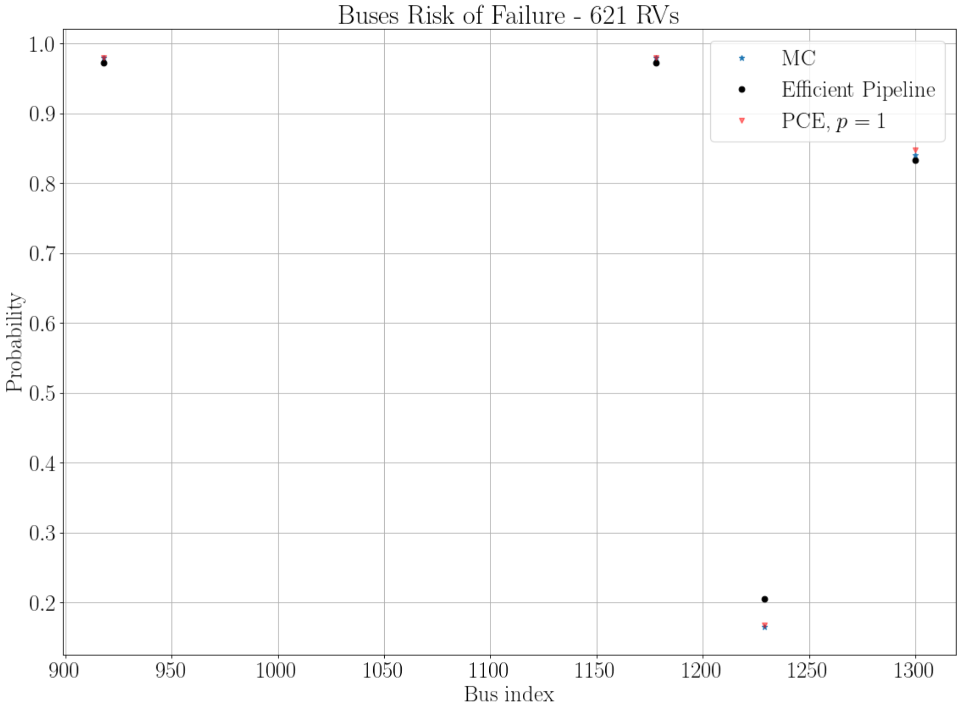


Figure 8.13.: Failure Risk of Critical Buses – Standard Case with 621 RVs.

Method name	Failure risk bus 919	Failure risk bus 1179	Failure risk bus 1230	Failure risk bus 1301	Computational cost (s)
MC	0.98	0.98	0.17	0.84	10^4
Eff. Pipeline	0.97	0.97	0.20	0.83	10^2
PCE, $p = 1$	0.98	0.98	0.17	0.85	2.2×10^3

Table 8.5.: Details on the risk of failure of critical buses.

For this case, the similarity between the histograms from PCE and the Efficient Pipeline and those from MC decreases compared to the former standard cases. This is because we used the same training data set size for all standard cases. As the number of input RVs increases, the training data set size should also increase to ensure the surrogate models can accurately capture the uncertainties of the high-dimensional stochastic space. Nevertheless, for the sake of time comparison and to assess the accuracy of the standard cases, we kept the training data set size the same for all cases.

In Figures 8.14 and 8.15, we present the compared histograms, given from results obtained from different schemes, of the voltage magnitudes for each critical bus.

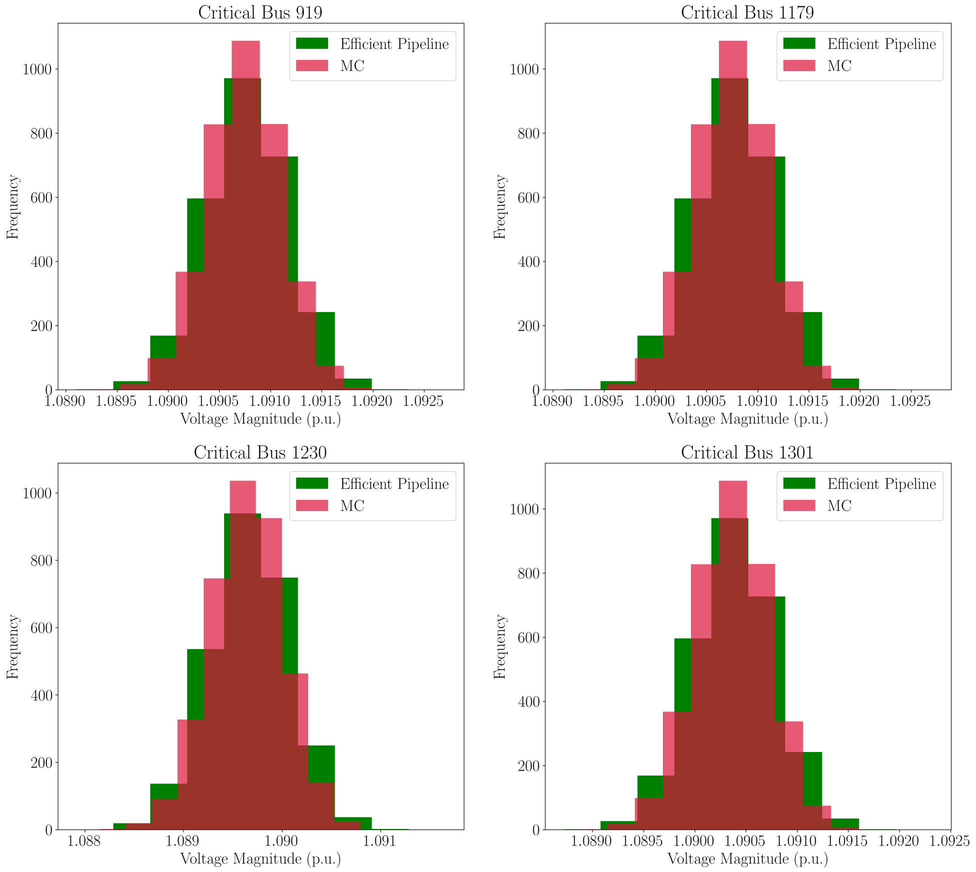


Figure 8.14.: Comparison between histograms from MC and Efficient Pipeline.

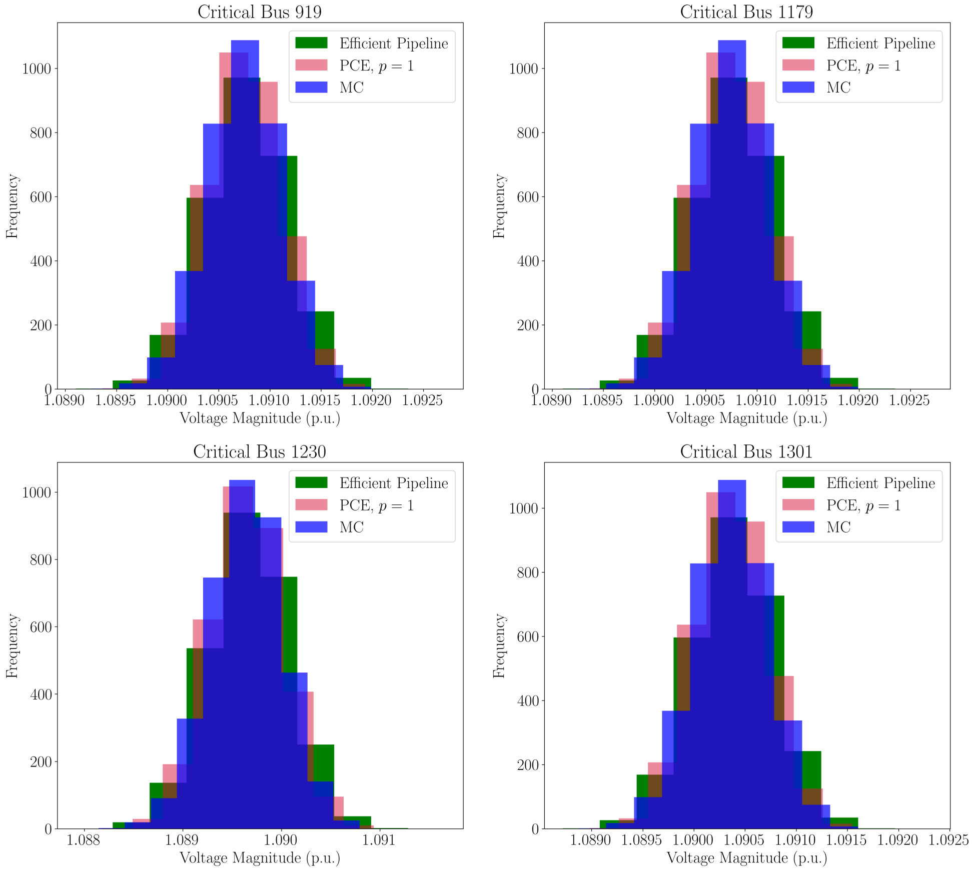


Figure 8.15.: Comparison between histograms from MC, Efficient Pipeline, and PCE.

8.2. Critical Buses and Influential RVs on the Map

In this section we present the critical buses (highlighted as large red dots) on the *1354pegase* grid map for each standard case. The most influential buses are determined by the rank-based estimation sensitivity analysis method.

In the maps, influential load buses within the same partition are represented with the same color, indicating they share the same probability distribution. Different partitions are shown in different colors. However, in the case of 621 RVs, all load buses with different probability distributions are represented by the same color, blue.

8.2.1. Standard Case – 100 RVs

Table 8.6.: Grid map with critical buses and their influential RVs - Standard Case 100 RVs.

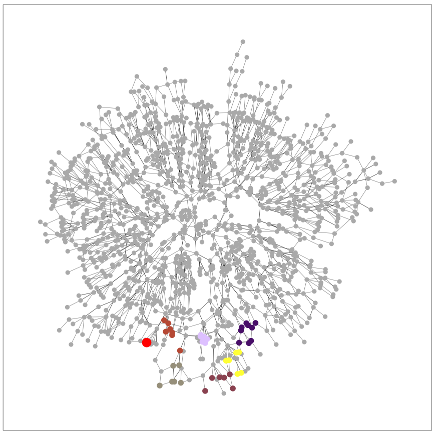
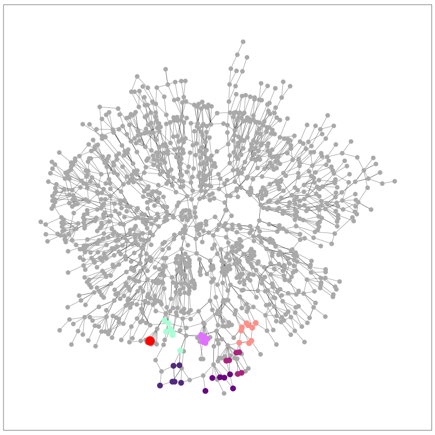
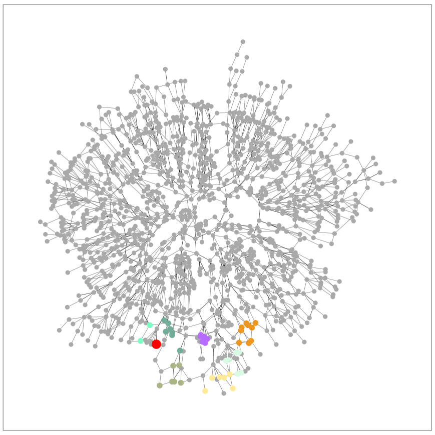
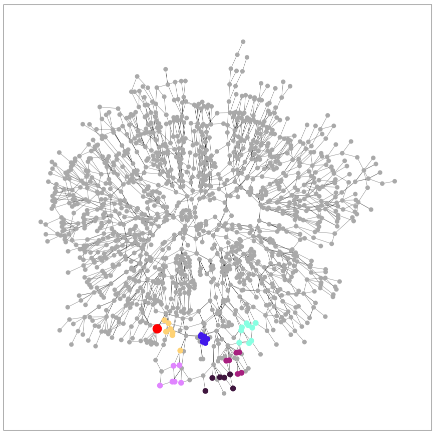
Critical Bus 919 (big red one)	Critical Bus 1179 (big red one)
	

Table 8.7.: Grid map with critical buses and their influential RVs - Standard Case 100 RVs.

Critical Bus 1230 (big red one)	Critical Bus 1301 (big red one)
	

8.2.2. Standard Case – 200 RVs

Table 8.8.: Grid map with critical buses and their influential RVs - Standard Case 200 RVs.

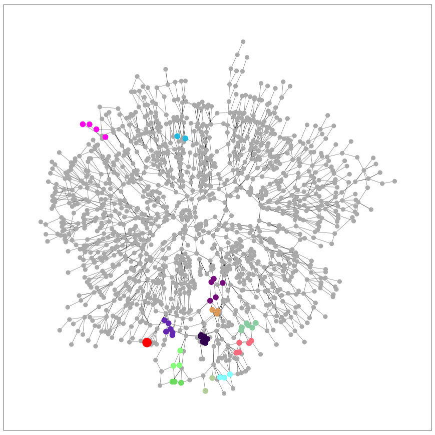
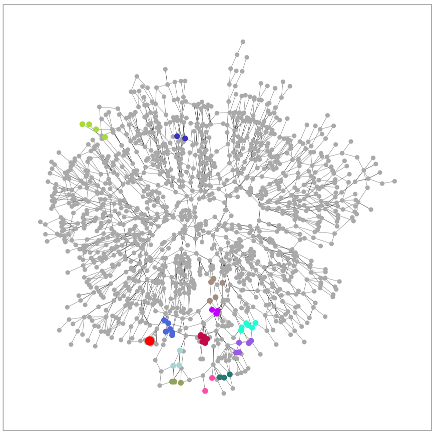
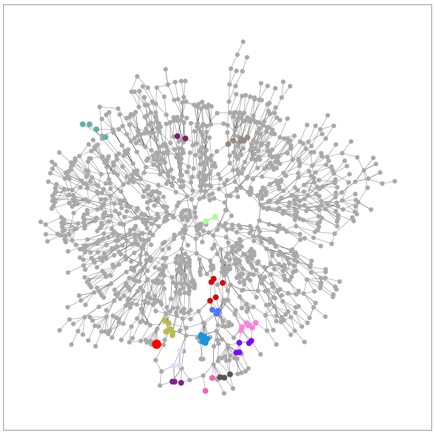
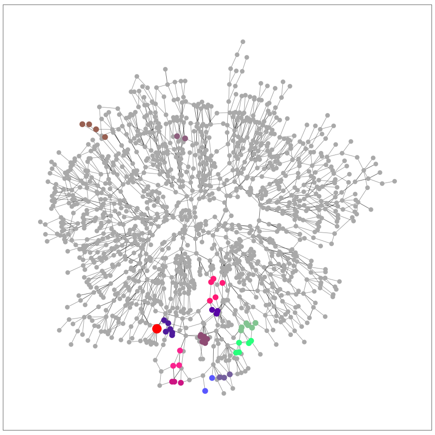
Critical Bus 919 (big red one)	Critical Bus 1179 (big red one)
	

Table 8.9.: Grid map with critical buses and their influential RVs - Standard Case 200 RVs.

Critical Bus 1230 (big red one)	Critical Bus 1301 (big red one)
	

8.2.3. Standard Case – 400 RVs

Table 8.10.: Grid map with critical buses and their influential RVs - Standard Case 400 RVs.

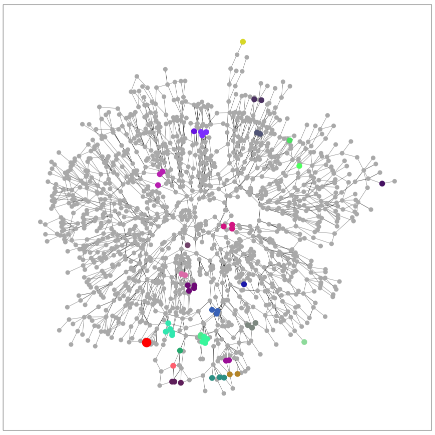
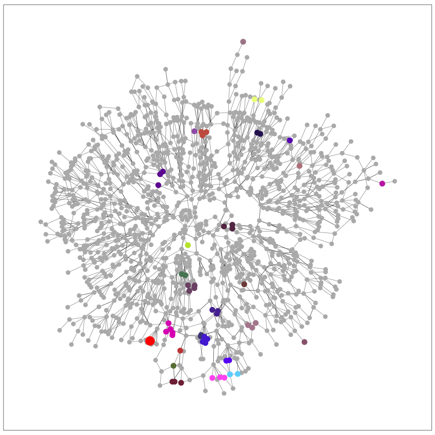
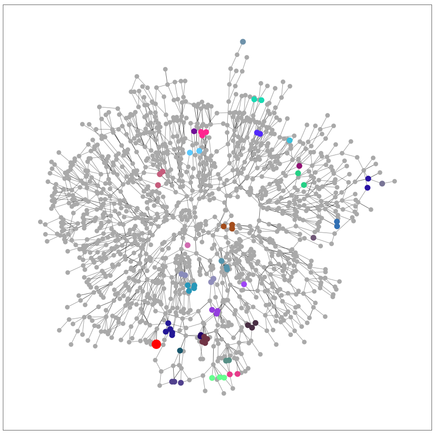
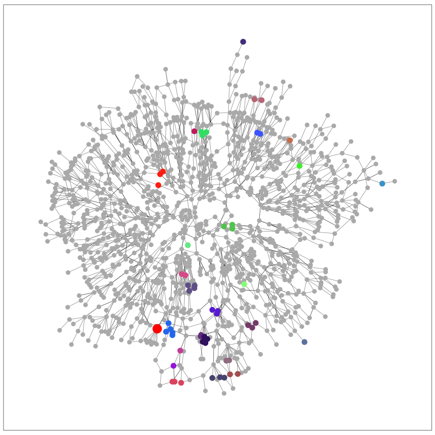
Critical Bus 919 (big red one)	Critical Bus 1179 (big red one)
	

Table 8.11.: Grid map with critical buses and their influential RVs - Standard Case 400 RVs.

Critical Bus 1230 (big red one)	Critical Bus 1301 (big red one)
	

8.2.4. Standard Case – 500 RVs

Table 8.12.: Grid map with critical buses and their influential RVs - Standard Case 500 RVs.

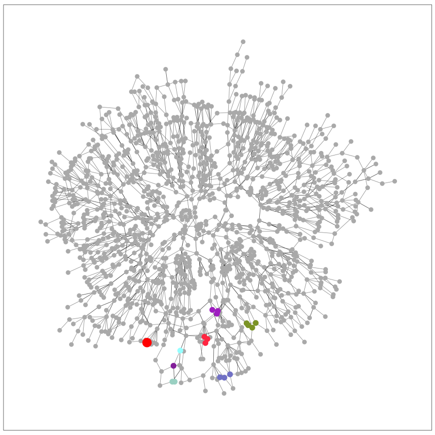
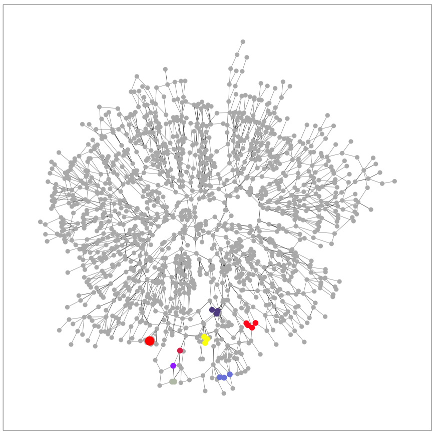
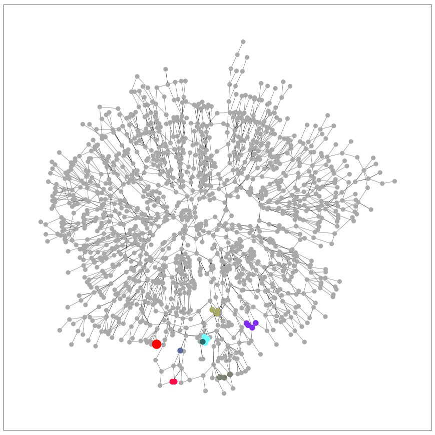
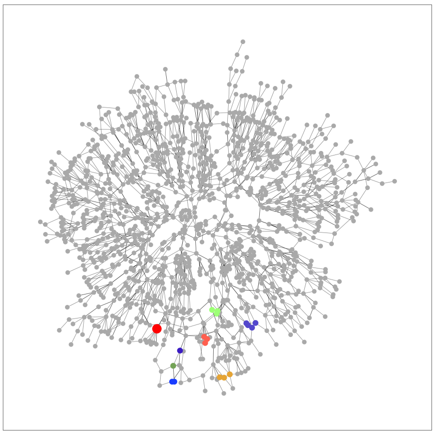
Critical Bus 919 (big red one)	Critical Bus 1179 (big red one)
	

Table 8.13.: Grid map with critical buses and their influential RVs - Standard Case 500 RVs.

Critical Bus 1230 (big red one)	Critical Bus 1301 (big red one)
	

8.2.5. Standard Case – 621 RVs

Table 8.14.: Grid map with critical buses and their influential RVs - Standard Case 621 RVs.

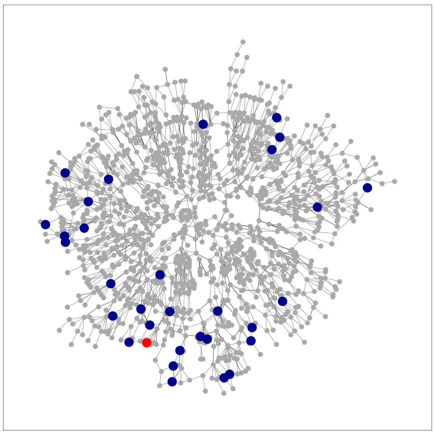
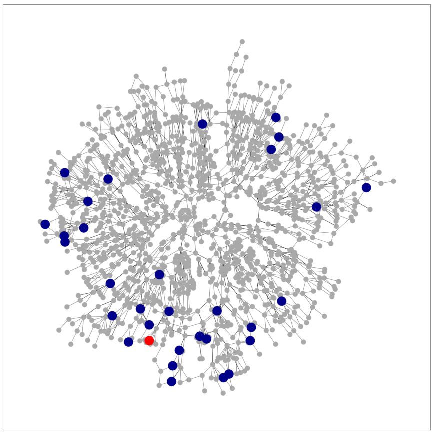
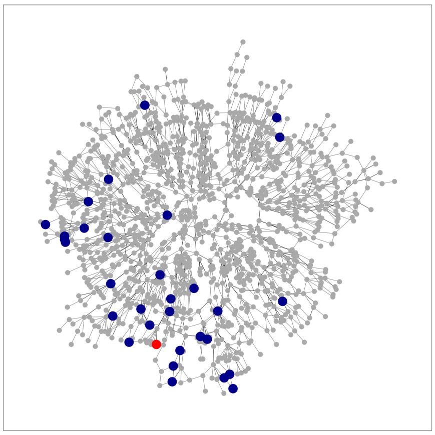
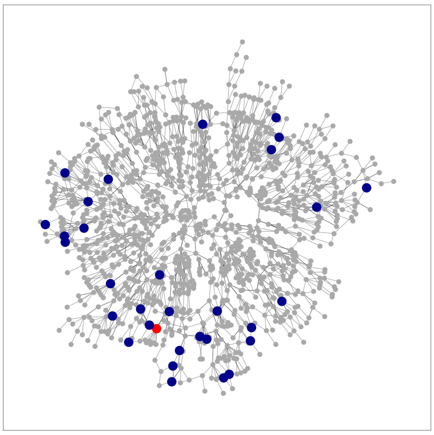
Critical Bus 919 (big red one)	Critical Bus 1179 (big red one)
	

Table 8.15.: Grid map with critical buses and their influential RVs - Standard Case 621 RVs.

Critical Bus 1230 (big red one)	Critical Bus 1301 (big red one)
	

8.3. Comparison of Computation Times

The comparisons of computation time between the three computation schemes used — Monte Carlo, Polynomial Chaos Expansion with degrees $p = 1, 2$, and the Efficient Pipeline — is shown in Figure 8.16.

The figure illustrates the performance of the Efficient Pipeline relative to the other methods. From the graph, it is evident that the Efficient Pipeline demonstrates an advantage in computation time over all the compared schemes. The Monte Carlo method is the least efficient among these.

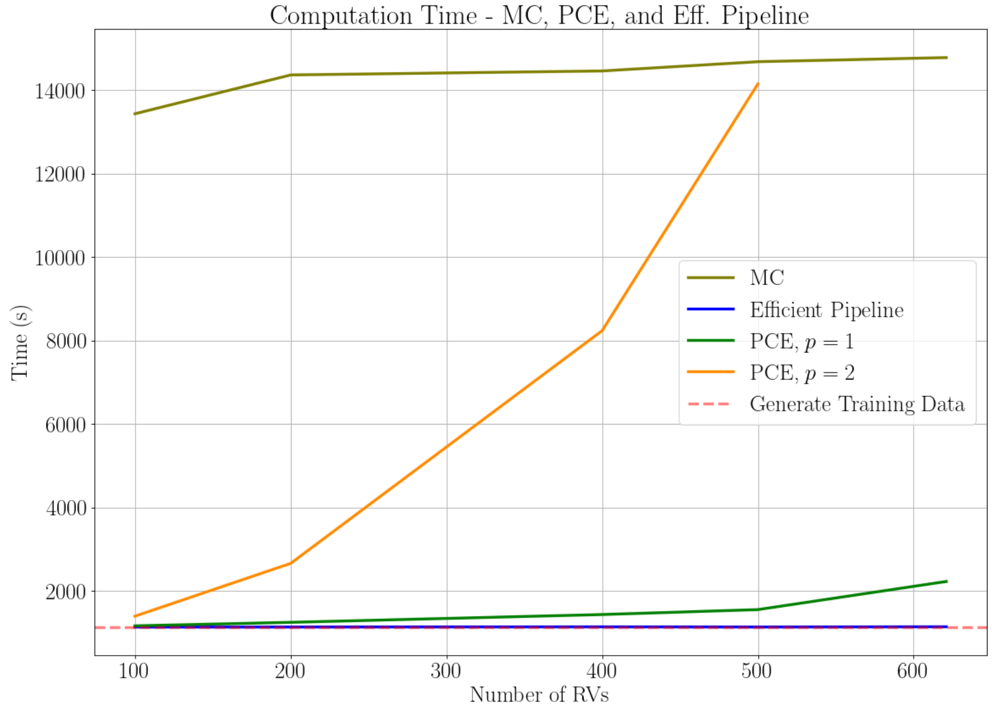


Figure 8.16.: Computation time comparison between the Efficient Pipeline, PCEs and MC.

9. Conclusion

In this work, we explained the theory behind the PPF analysis and explored various approaches for applying forward Uncertainty Quantification to solve the PPF problem. MC simulations were significantly less efficient than the surrogate models using PCE implemented in this study. Additionally, we proposed an Efficient Pipeline that leverages the reduction of random variables through clustering, a novel rank-based sensitivity analysis technique, and PCE to solve the PPF problem, ultimately determining the probabilities of critical buses operating outside their voltage magnitude range.

Our Efficient Pipeline, as demonstrated in our case study, proved to be more efficient than MC simulations and standalone PCE modeling (with degrees 1 or 2). Importantly, this pipeline does not scale adversely when the number of random variables are increased, given an initial training data set size, ensuring its adaptability to varying stochastic computational needs.

The new Efficient Pipeline has broad applicability and can be a game-changer in various domains where PPF analysis is crucial. From cost optimization of power grids to security analysis and monitoring of large-scale power systems, this pipeline can be applied to a wide range of scenarios, inspiring new possibilities in power system management and optimization. Moreover, the methods presented in this work can be enhanced through advancements in machine learning approaches and parallel computing using General-Purpose Graphics Processing Units (GPGPUs) and distributed memory systems. Specifically, the generation of training data sets with MC simulations is highly parallelizable, which would significantly improve the efficiency of surrogate modeling both with PCE alone and our proposed Efficient Pipeline.

In summary, the proposed Efficient Pipeline offers a promising solution for efficient and scalable PPF analysis with high-dimensional stochastic space, with potential applications across various power system management and optimization domains. Future work can further refine these methods and explore their integration with emerging computational technologies to enhance their applicability and performance.

Appendix

A. Software Tools

A.1. PANDAPOWER

The power network utilized in the case study of this thesis was developed using the open-source library Pandapower[1] [63], which builds upon the PANDAS[2] library and the power systems analysis toolbox PYPOWER[3].

Pandapower offers an extensive range of electric components for element-based power network calculations. The primary objective of the library is to facilitate static analysis of three-phase power systems, enabling the examination of *three-phase distribution systems* (commonly found in Europe), as well as *transmission* and *subtransmission systems*.

The program employs a tabular-based data structure where all elements are represented by tables containing their respective parameters. Users select these parameters depending on various analysis methods to address their specific problems. Upon completion of the analysis, a new table of results is generated, with the results' parameters depending on the analysis type conducted.

A.1.1. Power Grid Creation

Pandapower facilitates the creation of customized power grid networks from scratch through its APIs. Users intending to develop their power networks can utilize a set of commands to initialize an empty power grid and incrementally add buses, lines, and other components to construct a complete power grid[4].

In addition to this capability, Pandapower offers synthetic and benchmark networks via the networks module, which can be accessed using the command *Pandapower.network*. In this work, we will used the power system test case *Case 1354pegase*, which has 1354 buses and 621 loads. More on this test case will be elaborated in the next chapters.

A.1.2. Power Flow Analysis with Pandapower

The power flow system of equations defined in Chapter 2 is solved by Pandapower with the command *pandapower.runpp(net, algorithm='list')*[5], where *net* is an specific power grid

[1] https://www.Pandapower.org/about/

[2] https://pandas.pydata.org/

[3] https://pypi.org/project/PYPOWER/

[4] https://www.Pandapower.org/start/

created by Pandapower (with buses, lines and other elements characteristic of the grid), and 'list' refers to one of the algorithms listed below:

1. *'nr'* Newton-Raphson (pypower implementation with numba accelerations),
2. *'iwamoto_nr'* Newton-Raphson with Iwamoto multiplier,
3. *'bfsw'* backward/forward sweep,
4. *'gs'* gauss-seidel (pypower implementation),
5. *'fdbx'* fast-decoupled (pypower implementation),
6. *'fdxb'* fast-decoupled (pypower implementation).

A.2. OPENTURNS

OpenTURNS[6] (Open source Treatment of Uncertainty, Risk'N Statistics) [2] is an open-source software library dedicated to uncertainty quantification in numerical simulations. Developed by a consortium of industrial and academic partners, OpenTURNS provides a comprehensive framework for probabilistic modeling, statistical analysis, and uncertainty propagation.

Openturns supports various probability distributions and copulas for modeling complex dependencies between random variables and allows users to define custom probability distributions and empirical data-based models. Moreover, the library also offers tools for descriptive statistics, hypothesis testing, and parameter estimation, providing methods for fitting distributions to data and performing goodness-of-fit tests.

In uncertainty propagation, it is possible to implement various methods with OpenTURNS for propagating uncertainties through mathematical models, including Monte Carlo simulation, Latin Hypercube Sampling, and Polynomial Chaos Expansion (PCE).

A.2.1. MC Sampling with OpenTURNS

We use OpenTURNS to create the multivariate random variable distributions for our stochastic input and to generate random samples from these distributions. The method by which the stochastic fluctuations are created and assigned to the partitions of our test grid is detailed in Section B.2.

A.2.2. PCE with OpenTURNS

In this work, we use OpenTURNS to construct Polynomial Chaos Expansion (PCE) to create the surrogate model of our power grid test case. A customized class *PolynomialChaosExpansion()* was utilized.

[6] https://openturns.github.io/www/index.html#

A.3. NETWORKX

The NetworkX[7] [26] package is a versatile and powerful network analysis tool developed in Python. NetworkX provides fundamental network data structures that represent various graphs, including simple graphs, directed graphs, and graphs with self-loops and parallel edges. Notably, NetworkX supports using (almost) arbitrary objects as nodes and associating arbitrary objects with edges. This flexibility allows for the seamless integration of network structures with custom objects and data structures, complementing any pre-existing code and facilitating network analysis in diverse application settings without significant software development.

Once a network is represented as a NetworkX object, a range of standard algorithms can be employed to analyze its structure. These algorithms include those for determining degree distributions (the number of edges incident to each node), clustering coefficients (the number of triangles each node is part of), shortest paths, spectral measures, and community detection. This comprehensive suite of tools enables detailed and sophisticated analysis of network properties and behaviors, making NetworkX an essential tool for researchers and practitioners in network science.

A.3.1. Creation of Graphs

Transforming a Pandapower network into a NetworkX Multigraph can be easily accomplished. The *Pandapower.topology* module provides the method *create_nxgraph*, which facilitates the conversion of a Pandapower network into a Multigraph[8]. Our application used the Python function detailed in A.1 to create a NetworkX Multigraph from the network object *net*.

```python
import Pandapower.topology as top
def create_graph(net):

    G = top.create_nxgraph(net)

    return G

```

Source Code A.1.: NetworkX graph creation from Pandapower network (*net*)

This function simplifies the integration of power network analysis with NetworkX's

[7] https://networkx.org/documentation/stable/

[8] https://Pandapower.readthedocs.io/en/v2.0.0/topology/create_graph.html

graph-based analysis capabilities, enhancing the flexibility and functionality of our research tools such as the application of clustering algorithms.

A.3.2. Graph Partitioning

Once the power grid graph is created, we can utilize it for further analysis. In this thesis, the graph of the power grid network is partitioned into regions based on the geographical proximity of the buses. We assume buses within the same partition exhibit identical fluctuations, meaning the same marginal random variable describes their fluctuations. Conversely, buses from different partitions have completely independent fluctuations, with their fluctuations described by i.i.d. random variables.

The K-Means algorithm is employed to obtain the partitions of the power grid, as we will explain in next section, and a Python function was developed to assign the random fluctuation distributions respective to each partition as describes in the subsetion A.2.1 and in the code B.2. Scikit-learn provides a standard package for implementing the K-Means algorithm, among other clustering algorithms. The next chapter will present Scikit-learn and the Python function used for the graph partitioning.

A.4. SCIKIT-LEARN

Scikit-learn[9] is a Python library that provides a standard interface for implementing machine learning algorithms [7]. It encompasses a variety of ancillary functions integral to the machine learning pipeline, including data preprocessing steps, data resampling techniques, evaluation metrics, and search interfaces for tuning and optimizing an algorithm's performance. This comprehensive suite of tools allows for efficient development, evaluation, and refinement of machine learning models, facilitating robust and reliable predictive analytics.

A.4.1. Grid Partition with K-Means

The K-Means algorithm, described in Algorithm **??**, was implemented using the Scikit-learn package *sklearn.cluster* and the *KMeans* method[10]. The dataset chosen for clustering comprised the geographical coordinates (x, y) of the buses in our power grid test case. This selection is based on the assumption that bus fluctuations are significantly influenced by their geographical locations, owing to the connections between buses and geographically-dependent conditions of renewable energy generation, such as sunlight intensity (for solar energy sources) and wind speed (for wind turbine generators). More details on the implementation of the K-Means algorithm is show in B.3

[9] `https://scikit-learn.org/stable/getting_started.html`

[10] `https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans`

B. Source Codes

B.1. Rank-based Estimation

```python
import numpy as np

def rank_estimation(Y_r, X):

        dim = X.shape[1]
        N = X.shape[0]

        if len(Y_r.shape) == 2:
                Y_r = Y_r.ravel()
        mean = np.mean(Y_r)
        var = np.var(Y_r, ddof=1)
        sobol = np.zeros(dim)
        px = X.argsort(axis=0)

        pi_j = px.argsort(axis=0) + 1

        argpiinv = (pi_j % N) + 1
        for i in range(dim):
                N_j = px[argpiinv[:, i] - 1, i]
                YN_j = Y_r[N_j]
                sobol[i] = (np.mean(Y_r*YN_j) - mean**2)/var

        return sobol

```

Source Code B.1.: Rank-estimation function[1]

[1] https://github.com/elizqian/mfgsa/tree/master

B.2. Creation of Stochastic Fluctuations

```python
import openturns as ot
def set_fluctuation(bus_active_areas):

        active_fluctuation = []

        dist_active = {}

        for key in bus_active_areas.keys():
        # Active areas are the partitioned areas with active loads

                a = np.random.uniform(1,2,size=1)

                if a < 0.0:
                        a = abs(a)
                elif a == 0.0:
                        a = 0.1
                dist_active[key] = a[0]

        for key in bus_active_areas.keys():

                a = dist_active[key]
                active_fluctuation.append(ot.Uniform(-a,a))

        dist_partition_a = ot.ComposedDistribution(active_fluctuation)

        return dist_partition_a

```

Source Code B.2.: Creation of stochastic fluctuations

B.3. K-Means Implementation

```

from sklearn.cluster import KMeans
def partition_kmeans(pos, n_clusters=16, random_state=0):

        ## Array of coordinates
        C = []
        Carray = np.zeros([len(pos),2])
        for key in pos.keys():
                C.append(np.array(pos[key]))

        for i, val in enumerate(C):
                Carray[i,0] = val[0]
                Carray[i,1] = val[1]


        ##  K-Means Algorithm
        algo = KMeans(n_clusters=n_clusters, \
                random_state=random_state, n_init=3)
        algo.fit(Carray)
        centers = algo.cluster_centers_

        L = algo.labels_

        P = {} # Dictionary with partitions
        for key in L:
                P[key]=[]

        for i, labels in enumerate(L):
                P[labels].append(i)

        return P, centers

```

Source Code B.3.: K-Means Implementation

glossaries

Bibliography

[1] Morteza Aien, Masoud Rashidinejad, and Mahmud Fotuhi-Firuzabad. On possibilistic and probabilistic uncertainty assessment of power flow problem: A review and a new approach. *Renewable and Sustainable energy reviews*, 37:883–895, 2014.

[2] G Andrianov, S Burriel, S Cambier, A Dutfoy, I Dutka-Malen, E De Rocquigny, B Sudret, P Benjamin, R Lebrun, F Mangeant, et al. Open turns, an open source initiative to treat uncertainties, risks' n statistics in a structured industrial approach. In *Proceedings of the ESREL'2007 Safety and Reliability Conference, Stavenger: Norway*, 2007.

[3] Richard Askey and James Arthur Wilson. *Some basic hypergeometric orthogonal polynomials that generalize Jacobi polynomials*, volume 319. American Mathematical Soc., 1985.

[4] Florian Augustin, A Gilg, M Paffrath, P Rentrop, and U Wever. Polynomial chaos for the approximation of uncertainties: Chances and limits. *European Journal of Applied Mathematics*, 19(2):149–190, 2008.

[5] J Bada, AD Vidal, Y Komazawa, N Ledanois, H Yaqoob, A Brown, JL Sawin, H Abdelnabi, H Couzin, A El Guindy, et al. Renewables 2023 global status report. *Report REN21*, 2023.

[6] Jordan Bell. The cameron-martin theorem. 2015.

[7] Ekaba Bisong and Ekaba Bisong. Introduction to scikit-learn. *Building machine learning and deep learning models on google cloud platform: a comprehensive guide for beginners*, pages 215–229, 2019.

[8] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, oct 2008.

[9] Adam Bobrowski. *Functional analysis for probability and stochastic processes: an introduction*. Cambridge University Press, 2005.

[10] Hans-Hermann Bock. Clustering methods: a history of k-means algorithms. *Selected contributions in data analysis and classification*, pages 161–172, 2007.

[11] Stephen P Brooks and Andrew Gelman. General methods for monitoring convergence of iterative simulations. *Journal of computational and graphical statistics*, 7(4):434–455, 1998.

[12] John Burkardt. K-means clustering. *Virginia Tech, Advanced Research Computing, Interdisciplinary Center for Applied Mathematics*, 2009.

[13] Defu Cai, Dongyuan Shi, and Jinfu Chen. Probabilistic load flow computation using copula and latin hypercube sampling. *IET Generation, Transmission & Distribution*, 8(9):1539–1549, 2014.

[14] Robert H Cameron and William T Martin. The orthogonal development of non-linear functionals in series of fourier-hermite functionals. *Annals of Mathematics*, pages 385–392, 1947.

[15] Sourav Chatterjee. A new coefficient of correlation. *Journal of the American Statistical Association*, 116(536):2009–2022, 2021.

[16] Aaron Clauset, M. E. J. Newman, and Cristopher Moore. Finding community structure in very large networks. *Physical Review E*, 70(6), December 2004.

[17] Julian Clausnitzer and Andreas Kleefeld. Comparing intrusive and non-intrusive polynomial chaos for a class of exponential time differencing schemes. *arXiv preprint arXiv:2311.16921*, 2023.

[18] Gonzalo Esteban Constante-Flores and Mahesh S Illindala. Data-driven probabilistic power flow analysis for a distribution system with renewable energy sources using monte carlo simulation. *IEEE Transactions on Industry Applications*, 55(1):174–181, 2018.

[19] Bradley Efron and Charles Stein. The jackknife estimate of variance. *The Annals of Statistics*, pages 586–596, 1981.

[20] Brendan J Frey and Delbert Dueck. Clustering by passing messages between data points. *science*, 315(5814):972–976, 2007.

[21] Fabrice Gamboa, Pierre Gremaud, Thierry Klein, and Agnès Lagnoux. Global sensitivity analysis: A novel generation of mighty estimators based on rank statistics. *Bernoulli*, 28(4), 2022.

[22] Fabrice Gamboa, Thierry Klein, and Agnès Lagnoux. Sensitivity analysis based on cramér–von mises distance. *SIAM/ASA Journal on Uncertainty Quantification*, 6(2):522–548, 2018.

[23] Roger G Ghanem and Pol D Spanos. *Stochastic finite elements: a spectral approach*. Courier Corporation, 2003.

[24] Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826, 2002.

[25] Michael Griebel and Markus Holtz. Dimension-wise integration of high-dimensional functions with applications to finance. *Journal of Complexity*, 26(5):455–489, 2010.

[26] Aric Hagberg, Pieter J Swart, and Daniel A Schult. Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Laboratory (LANL), Los Alamos, NM (United States), 2008.

[27] Mahdi Hajian, William D Rosehart, and Hamidreza Zareipour. Probabilistic power flow by monte carlo simulation with latin supercube sampling. *IEEE Transactions on Power Systems*, 28(2):1550–1559, 2012.

[28] W Keith Hastings. Monte carlo sampling methods using markov chains and their applications. 1970.

[29] Wassily Hoeffding. A class of statistics with asymptotically normal distribution. *Breakthroughs in statistics: Foundations and basic theory*, pages 308–334, 1992.

[30] Jonas Hörsch, Fabian Hofmann, David Schlachtberger, and Tom Brown. Pypsa-eur: An open optimisation model of the european transmission system. *Energy strategy reviews*, 22:207–215, 2018.

[31] Anil K Jain and Richard C Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.

[32] Katiana Kontolati, Dimitrios Loukrezis, Ketson RM Dos Santos, Dimitrios G Giovanis, and Michael D Shields. Manifold learning-based polynomial chaos expansions for high-dimensional surrogate models. *International Journal for Uncertainty Quantification*, 12(4), 2022.

[33] Dirk P Kroese, Thomas Taimre, and Zdravko I Botev. *Handbook of monte carlo methods*. John Wiley & Sons, 2013.

[34] Chun Sing Lai, Youwei Jia, Malcolm D McCulloch, and Zhao Xu. Daily clearness index profiles cluster analysis for photovoltaic system. *IEEE Transactions on Industrial Informatics*, 13(5):2322–2332, 2017.

[35] Christiane Lemieux. Quasi–monte carlo constructions. *Monte Carlo and Quasi-Monte Carlo Sampling*, pages 1–61, 2009.

[36] Quan Li and Nan Zhao. Probabilistic power flow calculation based on importance-hammersley sampling with eigen-decomposition. *International Journal of Electrical Power & Energy Systems*, 130:106947, 2021.

[37] Youguo Li and Haiyan Wu. A clustering method based on k-means algorithm. *Physics Procedia*, 25:1104–1109, 2012.

[38] Teems E Lovett, Ferdinanda Ponci, and Antonello Monti. A polynomial chaos approach to measurement uncertainty. *IEEE transactions on instrumentation and measurement*, 55(3):729–736, 2006.

[39] Manjarini Mallik, Ayan Kumar Panja, and Chandreyee Chowdhury. Paving the way with machine learning for seamless indoor–outdoor positioning: A survey. *Information Fusion*, 94:126–151, 2023.

[40] Steffen Meinecke, Annika Klettke, Dzanan Sarajlic, Jörg Dickert, Matthias Hable, Franziska Fischer, Martin Braun, and Albert Moser. General planning and operational principles in german distribution systems used for simbench. 2019.

[41] David Métivier, Marc Vuffray, and Sidhant Misra. Efficient polynomial chaos expansion for uncertainty quantification in power systems. 189:106791.

[42] Nicholas Metropolis and Stanislaw Ulam. The monte carlo method. *Journal of the American statistical association*, 44(247):335–341, 1949.

[43] Tillmann Mühlpfordt, Timm Faulwasser, and Veit Hagenmeyer. Solving stochastic ac power flow via polynomial chaos expansion. pages 70–76, 2016.

[44] Tillmann Mühlpfordt, Line Roald, Veit Hagenmeyer, Timm Faulwasser, and Sidhant Misra. Chance-constrained ac optimal power flow: A polynomial chaos approach. *IEEE Transactions on Power Systems*, 34(6):4806–4816, 2019.

[45] M. E. J. Newman. Equivalence between modularity optimization and maximum likelihood methods for community detection. *Phys. Rev. E*, 94:052315, Nov 2016.

[46] Fei Ni, Phuong H Nguyen, and Joseph FG Cobben. Basis-adaptive sparse polynomial chaos expansion for probabilistic power flow. *IEEE Transactions on Power Systems*, 32(1):694–704, 2016.

[47] Parikshit Pareek and Hung D Nguyen. Gaussian process learning-based probabilistic optimal power flow. *IEEE Transactions on Power Systems*, 36(1):541–544, 2020.

[48] Roger D Peng. Advanced statistical computing. *Work in progress*, page 121, 2018.

[49] Elizabeth Qian, Benjamin Peherstorfer, Daniel O'Malley, Velimir V Vesselinov, and Karen Willcox. Multifidelity monte carlo estimation of variance and sensitivity indices. *SIAM/ASA Journal on Uncertainty Quantification*, 6(2):683–706, 2018.

[50] Abbas Rabiee, Alireza Soroudi, and Andrew Keane. Information gap decision theory based opf with hvdc connected wind farms. *IEEE Transactions on Power Systems*, 30(6):3396–3406, 2014.

[51] Herschel Rabitz and Ömer F Aliş. General foundations of high-dimensional model representations. *Journal of Mathematical Chemistry*, 25(2):197–233, 1999.

[52] Sharif Rahman. A generalized anova dimensional decomposition for dependent probability measures. *SIAM/ASA Journal on Uncertainty Quantification*, 2(1):670–697, 2014.

[53] Andrea Saltelli, Paola Annoni, Ivano Azzini, Francesca Campolongo, Marco Ratto, and Stefano Tarantola. Variance based sensitivity analysis of model output. design and estimator for the total sensitivity index. *Computer physics communications*, 181(2):259–270, 2010.

[54] Andrea Saltelli and Ilya M Sobol. About the use of rank transformation in sensitivity analysis of model output. *Reliability Engineering & System Safety*, 50(3):225–239, 1995.

[55] Andrea Saltelli and Stefano Tarantola. On the relative importance of input factors in mathematical models: safety assessment for nuclear waste disposal. *Journal of the American Statistical Association*, 97(459):702–709, 2002.

[56] Andrea Saltelli, Stefano Tarantola, and Francesca Campolongo. Sensitivity analysis as an ingredient of modeling. *Statistical science*, pages 377–395, 2000.

[57] Andrea Saltelli, Stefano Tarantola, and KP-S Chan. A quantitative model-independent method for global sensitivity analysis of model output. *Technometrics*, 41(1):39–56, 1999.

[58] Ilya M Sobol. Global sensitivity indices for nonlinear mathematical models and their monte carlo estimates. *Mathematics and computers in simulation*, 55(1-3):271–280, 2001.

[59] Ilya M Sobol'. Theorems and examples on high dimensional model representation. *Reliability Engineering and System Safety*, 79(2):187–193, 2003.

[60] IM Soboĺ. Sensitivity estimates for nonlinear mathematical models. *Math. Model. Comput. Exp.*, 1:407, 1993.

[61] Bruno Sudret, Géraud Blatman, and Marc Berveiller. Response surfaces based on polynomial chaos expansions. *Construction reliability: safety, variability and sustainability*, pages 147–167, 2013.

[62] Yonghui Sun, Yan Zhou, Sen Wang, Rabea Jamil Mahfoud, Hassan Haes Alhelou, George Sideratos, Nikos Hatziargyriou, and Pierluigi Siano. Nonparametric probabilistic prediction of regional pv outputs based on granule-based clustering and direct optimization programming. *Journal of Modern Power Systems and Clean Energy*, 11(5):1450–1461, 2023.

[63] L. Thurner, A. Scheidler, F. Schäfer, J. Menke, J. Dollichon, F. Meier, S. Meinecke, and M. Braun. pandapower — an open-source python tool for convenient modeling, analysis, and optimization of electric power systems. *IEEE Transactions on Power Systems*, 33(6):6510–6521, Nov 2018.

[64] Dootika Vats and Christina Knudson. Revisiting the gelman–rubin diagnostic. *Statistical Science*, 36(4):518–529, 2021.

[65] Norbert Wiener. The homogeneous chaos. *American Journal of Mathematics*, 60(4):897–936, 1938.

[66] Xu Wu, Tomasz Kozlowski, Hadi Meidani, and Koroush Shirvan. Inverse uncertainty quantification using the modular bayesian approach based on gaussian process, part 1: Theory. *Nuclear Engineering and Design*, 335:339–355, 2018.

[67] Dongbin Xiu. *Numerical methods for stochastic computations: a spectral method approach*. Princeton university press, 2010.

[68] Dongbin Xiu and George Em Karniadakis. The wiener–askey polynomial chaos for stochastic differential equations. *SIAM journal on scientific computing*, 24(2):619–644, 2002.

[69] Taiyou Yong and RH Lasseter. Stochastic optimal power flow: formulation and solution. In *2000 Power Engineering Society Summer Meeting (Cat. No. 00CH37134)*, volume 1, pages 237–242. IEEE, 2000.

[70] Han Yu, CY Chung, KP Wong, HW Lee, and JH Zhang. Probabilistic load flow evaluation with hybrid latin hypercube sampling and cholesky decomposition. *IEEE Transactions on Power Systems*, 24(2):661–667, 2009.