Stochastics-based Methods Enabling Testing of Grid-related Algorithms through Simulation

Von der Fakultät für Elektrotechnik und Informationstechnik der Rheinisch-Westfälischen Technischen Hochschule Aachen zur Erlangung des akademischen Grades eines Doktors der Naturwissenschaften genehmigte Dissertation

vorgelegt von
Dipl.-Math. Kanali Togawa
aus
Düsseldorf, Deutschland

Berichter:
Univ.-Prof. Antonello Monti, Ph. D.
Univ.-Prof. Dr.-Ing. Albert Moser

Tag der mündlichen Prüfung: 14.07.2015

Diese Dissertation ist auf den Internetseiten der Hochschulbibliothek online verfügbar.
Abstract

This dissertation presents stochastics-based methods enabling testing related to three different aspects of the transition towards Smart Grids: the overall increase in sources of uncertainty, the need for studying the effects of higher shares of distributed generation on distribution grids, and the focus on single consumers through concepts such as demand side management.

A nonintrusive Polynomial Chaos approach is developed for fast uncertainty analysis. It is shown that by combining Polynomial Chaos and numerical integration, black box use of Polynomial Chaos can be achieved. Additionally, by using a single polynomial basis, the procedure is automated for parameters with arbitrary probability distributions, avoiding adjustments traditionally performed in Polynomial Chaos. It is shown that the results of 10000 Monte Carlo simulations can be achieved by post-processing as little as 6 simulations per random parameter, using deterministic integration points as inputs.

In order to allow for robust testing of distribution grid-related methods with several different topologies, an algorithm based on concepts from Graph Theory is designed for generating random distribution grid models. The algorithm separately generates medium voltage grid and low voltage grid models. A geographical reference is used in order to facilitate the assignment of distances and electrical properties, and through these the construction of admittance matrices for further use. The algorithm is validated by comparing the statistics of real grids with those of generated grids.

Finally, a framework is developed for the random generation of single load profiles for arbitrary types of consumers, based on standard load profiles as a reference for user activity. The generated profiles represent realistic challenges for simulation and testing thanks to the abrupt consumption behaviour, contrary to the smooth standard load profiles which can only be considered realistic for large numbers of consumers. It is shown through an implementation for households that a large number of generated load profiles behave similarly as the original standard load profile, thereby demonstrating their statistical correctness.
Acknowledgments

This dissertation was completed at the E.ON Energy Research Center with the support of many people.

I must thank Prof. Antonello Monti for his motivation over the last five years, beginning with sessions in his office in Mathieustr. 6 in which we experimented on polynomial chaos, and including several discussions in which he was always supportive and accepting. I am also grateful to Prof. Albert Moser for being my second examiner and reviewing my thesis, and to the Professors Michael Vorländer and Christoph Jungemann for agreeing to examine me at short notice.

The implementations of the algorithms presented in this thesis would not have been possible without the diligent work of the students Eran Schweitzer, Julia Hinßen and Sebastian Bolte. Especially the work with Eran was great fun thanks to the countless discussions.

But life in the institute would not have been half as interesting without the experiences I shared with my colleagues. Thanks to Prof. Ferdinanda Ponci for our conversations which made daily life in the institute more colourful. Thanks to Bettina Schäfer for her understanding ways. Thanks to Christoph Molitor and Tim Schlösser for great collaborations and good times. Thanks to Artur Löwen for the geeky jokes and philosophical discussions. Most of all, thanks to all assistants in ACS for their friendship beyond work in the institute, especially (in order of institute age) Andrea Benigni, Junqi Liu, Alexander Helmedag, Marco Lixia, Timo Isermann, Andrea Angioni, Maria Stevic, Simon Pickartz and Baptiste Feron. Thanks also to Nils Soltau from PGS for his positive attitude and for the legendary ACS-PGS crossover activities. I must definitely include Nicole Bielders and Janine Kahn in this list of friends, they made ACS a very special place.

Last but not least, I would like to express my gratitude to my friends and family for their love and care, especially to Stefan for his motivation, humour and patience without which I would not have been able to achieve this goal.
Contents

List of Abbreviations v
Mathematical annotations vii
List of Figures ix
List of Tables xiii

1 Introduction 1
  1.1 Motivation ................................................. 2
    1.1.1 Addressing challenging research questions for future grids:
    a scenario ................................................. 3
    1.1.2 Specifying the need for improved methodologies ............ 4
  1.2 Contributions of this thesis ................................ 7
  1.3 Dissertation outline ...................................... 7

2 A nonintrusive polynomial chaos approach to uncertainty analysis 9
  2.1 Fundamentals ............................................. 11
    2.1.1 Stochastic terms ..................................... 11
    2.1.2 Generalised polynomial chaos .......................... 12
    2.1.3 Traditionally solving stochastic differential equations with
    PC .......................................................... 13
    2.1.4 Gauss quadrature ...................................... 15
  2.2 Non-intrusive single basis polynomial chaos .................. 16
    2.2.1 Prerequisites .......................................... 17
2.2.2 Step 1: expressing continuous random parameters with a single-basis polynomial chaos ........................................... 18
2.2.3 Step 2: selecting deterministic evaluation points for the random parameters ......................................................... 20
2.2.4 Step 3: solving the system equations ................................. 21
2.2.5 Step 4: post-processing the simulation results ..................... 22

2.3 Application examples ............................................................. 24
2.3.1 Approximating nonuniform random variables ..................... 24
2.3.2 Nonintrusive PC simulation of algebraic expressions of random variables .............................................................. 26
2.3.3 NIPC simulation of a nonlinear dynamic system with uncertain parameters ......................................................... 28

2.4 Graphical user interface for NIPC ............................................. 31

2.5 Range of applicability ............................................................. 33
2.5.1 Advantages ........................................................................ 33
2.5.2 Disadvantages .................................................................. 36

2.6 Conclusion ............................................................................. 37

3 Statistical generation of distribution grid models ...................... 39
3.1 Fundamentals ........................................................................ 40
3.1.1 Definitions ...................................................................... 40
3.1.2 Matrix formulation .......................................................... 42
3.1.3 Voronoi diagrams ............................................................ 42
3.1.4 The power grid graph ....................................................... 44
3.1.5 Categorising different graph types ..................................... 44

3.2 Grid generation algorithm ....................................................... 46
3.2.1 Medium voltage grid generation ....................................... 46
3.2.2 Low voltage grid ............................................................. 55
3.2.3 Electrical properties ....................................................... 64

3.3 Validation .............................................................................. 66
3.3.1 Degree distribution and JDD ........................................... 67
3.3.2 Line lengths ................................................................... 71

3.4 Graphical user interface for topology generation .................... 76

3.5 Conclusion ............................................................................. 78
4 Generation of single load profiles for simulation and analysis 81
  4.1 Preliminaries .............................................. 82
    4.1.1 Definitions ........................................... 82
    4.1.2 Data ................................................ 83
  4.2 Development of the generation algorithm ..................... 86
    4.2.1 Generation of consumption events ...................... 86
    4.2.2 Distribution of consumption events .................... 88
  4.3 Performance of the algorithm ................................ 91
  4.4 Graphical user interface for load profile generation ........ 95
  4.5 Conclusion ................................................ 96

5 Conclusion 97

A Comparison of degree distributions of real Dutch grids and 1000 generated grids 101

B Comparison of SLPs with generated profiles 107

Bibliography 111
List of Abbreviations

CDF  Cumulative density function
GUI  Graphical user interface
JDD  Joint degree distribution
LV   Low voltage
MC   Monte Carlo
MV   Medium voltage
NIPC Nonintrusive Polynomial Chaos
PC   Polynomial Chaos
PDF  Probability density function
SLP  Standard load profile
Mathematical annotations

$\mathbb{N}$ The set of natural numbers
$\mathbb{Z}$ The set of integers
$\mathbb{R}$ The set of real numbers
$\mathbb{R}^+, \mathbb{R}^-$ The set of positive or negative real numbers
$\mathbb{R}_0^+, \mathbb{R}_0^-$ The set of positive or negative real numbers including zero
$\mathbb{C}$ The set of complex numbers
$[a,b], (a,b)$ An interval from $a$ to $b$ including (square brackets) or excluding (round brackets) its limits
$||v||$ The Euclidean norm of a vector $v$ in a Euclidean space
$|S|$ The size of a finite set $S$, i.e. the number of its elements
$\text{supp}(f)$ The support of a function $f$
$L^n(D)$ The Lebesgue space of all functions with domain $D$ whose $n$-th power has a finite integral over $D$
$\mathcal{N}(\mu,\sigma)$ The normal distribution with mean value $\mu \in \mathbb{R}$ and standard deviation $\sigma \in \mathbb{R}^+$
$\mathcal{U}(a,b)$ The uniform distribution between $a \in \mathbb{R}$ and $b \in \mathbb{R}$
$[vw]$ The line segment between points $v$ and $w$ in a vector space
List of Figures

2.1 Approximation of $\mathcal{N}(0,1)$ with Legendre PC ................. 25
2.2 Standard normal distribution: comparison of PC samples with samples from the original distribution .................... 26
2.3 Gamma(10,7) distribution: comparison of PC samples with samples from the original distribution .................... 27
2.4 Exponential(3) distribution: comparison of PC samples with samples from the original distribution .................... 27
2.5 Monte Carlo and nonintrusive Polynomial Chaos simulation of an expression with Gamma distributed random parameters .. 28
2.6 Monte Carlo and nonintrusive Polynomial Chaos simulation of an expression with exponentially distributed random parameters 29
2.7 Single-phase rectifier .............................................. 29
2.8 Nonintrusive Polynomial Chaos simulation of a single-phase rectifier 30
2.9 Distribution of load voltage at three time steps ...................... 31
2.10 Schematic of nonintrusive PC analysis with GUIs ..................... 32
2.11 Post-processing GUI for NIPC .................................... 33

3.1 Voronoi region surrounding a node $v$ with Delaunay triangulation graph ......................................................... 43
3.2 Boundaries for the input parameter $RU$ .............................. 51
3.3 Constructing subregions for different shapes of $\mathcal{V}_v$ ............. 57
3.4 Determining the position of the feeder node $v_1$ starting from the MV node $v$ .................................................. 61
3.5 Branching scheme with ratios of nodes per branch ................... 62
3.6 Simultaneous growth of two branches of a feeder .................... 63
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.7</td>
<td>Comparison of the JDDs of the NL1 grid and generated grids</td>
<td>69</td>
</tr>
<tr>
<td>3.8</td>
<td>Comparison of the degree distribution of the NL1 grid and 1000</td>
<td>70</td>
</tr>
<tr>
<td></td>
<td>generated grids</td>
<td></td>
</tr>
<tr>
<td>3.9</td>
<td>Comparison of the degree distribution of the NL1 grid and one</td>
<td>73</td>
</tr>
<tr>
<td></td>
<td>generated grid</td>
<td></td>
</tr>
<tr>
<td>3.10</td>
<td>Comparison of the line lengths of the NL1 grid and generated grids</td>
<td>75</td>
</tr>
<tr>
<td>3.11</td>
<td>Comparison of 12 Dutch grids’ normalised line lengths with 12</td>
<td>77</td>
</tr>
<tr>
<td></td>
<td>corresponding generated grids’ normalised line lengths</td>
<td></td>
</tr>
<tr>
<td>3.12</td>
<td>Distribution grid generator GUI with a generated MV network</td>
<td>84</td>
</tr>
<tr>
<td>3.13</td>
<td>Distribution grid generator GUI with a generated LV network</td>
<td>77</td>
</tr>
<tr>
<td>4.1</td>
<td>Standard load profiles for German households in different seasons</td>
<td>84</td>
</tr>
<tr>
<td>4.2</td>
<td>Measurement of active power consumption of household devices</td>
<td>85</td>
</tr>
<tr>
<td>4.3</td>
<td>Generated household load profiles for households with an annual</td>
<td>92</td>
</tr>
<tr>
<td></td>
<td>consumption of 1000 kWh</td>
<td></td>
</tr>
<tr>
<td>4.4</td>
<td>Comparison of the average of 5000 single load profiles with the</td>
<td>94</td>
</tr>
<tr>
<td></td>
<td>summer weekend day SLP</td>
<td></td>
</tr>
<tr>
<td>4.5</td>
<td>Difference between the average of 5000 single load profiles and</td>
<td>94</td>
</tr>
<tr>
<td></td>
<td>the summer weekend day SLP</td>
<td></td>
</tr>
<tr>
<td>4.6</td>
<td>GUI for load profile generation</td>
<td>95</td>
</tr>
<tr>
<td>A.1</td>
<td>Comparison of the degree distribution of the NL2 grid and 1000</td>
<td>101</td>
</tr>
<tr>
<td></td>
<td>generated grids</td>
<td></td>
</tr>
<tr>
<td>A.2</td>
<td>Comparison of the degree distribution of the NL3 grid and 1000</td>
<td>102</td>
</tr>
<tr>
<td></td>
<td>generated grids</td>
<td></td>
</tr>
<tr>
<td>A.3</td>
<td>Comparison of the degree distribution of the NL4 grid and 1000</td>
<td>103</td>
</tr>
<tr>
<td></td>
<td>generated grids</td>
<td></td>
</tr>
<tr>
<td>A.4</td>
<td>Comparison of the degree distribution of the NL5 grid and 1000</td>
<td>103</td>
</tr>
<tr>
<td></td>
<td>generated grids</td>
<td></td>
</tr>
<tr>
<td>A.5</td>
<td>Comparison of the degree distribution of the NL6 grid and 1000</td>
<td>103</td>
</tr>
<tr>
<td></td>
<td>generated grids</td>
<td></td>
</tr>
<tr>
<td>A.6</td>
<td>Comparison of the degree distribution of the NL7 grid and 1000</td>
<td>104</td>
</tr>
<tr>
<td></td>
<td>generated grids</td>
<td></td>
</tr>
<tr>
<td>A.7</td>
<td>Comparison of the degree distribution of the NL8 grid and 1000</td>
<td>104</td>
</tr>
<tr>
<td></td>
<td>generated grids</td>
<td></td>
</tr>
</tbody>
</table>
A.8 Comparison of the degree distribution of the NL9 grid and 1000 generated grids ........................................ 105
A.9 Comparison of the degree distribution of the NL10 grid and 1000 generated grids ........................................ 105
A.10 Comparison of the degree distribution of the NL11 grid and 1000 generated grids ........................................ 106
A.11 Comparison of the degree distribution of the NL12 grid and 1000 generated grids ........................................ 106

B.1 Comparison of the average of 5000 single load profiles with the intermediary season weekend day SLP ................ 107
B.2 Difference between the average of 5000 single load profiles and the intermediary season weekend day SLP ................ 108
B.3 Comparison of the average of 5000 single load profiles with the winter weekend day SLP ................................. 108
B.4 Difference between the average of 5000 single load profiles and the winter weekend day SLP ................................. 109
## List of Tables

1.1 Comparison of the traditional grid and the Smart Grid ........................... 2

2.1 Correspondence of continuous distributions and orthogonal polynomials ................................................. 13

2.2 The first 5 Legendre polynomials ......................................................... 18

2.3 Collocation points and weights for Gauss-Legendre quadrature ............... 21

2.4 Input point combinations for simulation scenarios .................................... 22

2.5 Indices of factors for multivariate orthogonal polynomials .................... 23

2.6 Comparison of distribution parameters of PC samples with the original distribution ........................................ 25

2.7 Component values for the single-phase rectifier ..................................... 30

3.1 Dutch grid data ...................................................................................... 67

3.2 ATLANTIDE Italian grid data ................................................................. 67

3.3 Difference between the JDDs of Dutch MV grids and 1000 generated grids with corresponding statistics ........................................................... 71

3.4 Comparison of line length distribution fits .............................................. 74

4.1 Energy consumption of household appliances .......................................... 85
Chapter 1

Introduction

The power grid is one of the largest physical infrastructures built by humans. Having grown from a community level supply for the wealthy in the Industrial Age to a connected network for all homes and industries, it was the foundation for many other technologies which followed, such as telecommunication and electronics [1].

The development of the power grid has however never been as drastic as over the last 15 years. Facing the limit of currently known fossil fuel capacities as well as an impending climate change, the need for alternative and interdis- ciplinary technologies has initiated an evolution towards the so-called Smart Grid. Despite the differences in the definition or the focus of authors, the Smart Grid generally depicts a scenario featuring some of the following technologies or properties [2, 3, 4]:

- a high degree of automation in all voltage levels through intelligence and communication,
- distributed generation,
- renewable energy sources,
- interaction with thermal systems through combined heat and power generation or the usage of waste heat,
- electric storage,
1.1 Motivation

- electric vehicles,

and many more. The introduction of the above mentioned features leads to significant changes in the way the grid is traditionally operated. An overview can be found in the following table [2].

<table>
<thead>
<tr>
<th>Traditional Grid</th>
<th>Smart Grid</th>
</tr>
</thead>
<tbody>
<tr>
<td>unidirectional power flow</td>
<td>bidirectional power flow</td>
</tr>
<tr>
<td>centralized generation</td>
<td>distributed generation</td>
</tr>
<tr>
<td>limited automation</td>
<td>highly automated</td>
</tr>
<tr>
<td>few customer choices</td>
<td>many customer choices</td>
</tr>
</tbody>
</table>

Table 1.1: Comparison of the traditional grid and the Smart Grid

The importance and public interest in these topics can be seen in activities such as the SET-Plan [5] and the many research activities of the European Union, the government of which is aiming to achieve among others 20% of energy from renewable sources and 20% decrease of greenhouse gas emissions compared to 1990 by year 2020, and envisions 27% of energy from renewable sources as well as a 40% decrease of greenhouse gas emissions compared to 1990 by year 2030. These changes strongly influence the role of the distribution grid, which traditionally was designed as a passive system with the sole purpose of delivering energy to end users. Nowadays, generation capabilities are readily available even to single houses thanks to technologies such as photovoltaic panels and combined heat and power units, thereby challenging the flexibility and capacity of the distribution grid. An extensive study on the necessity for additional development in the German distribution grid due to distributed generation has been conducted by the dena (the German Energy Agency) in three studies in 2005 [6], 2010 [7] and in 2012 [8], respectively.

1.1 Motivation

The above mentioned facts illustrate a reality in which the power system is strongly influenced by

- an overall surge in sources of uncertainty due to high shares of intermittent renewable generation;
more specifically the behaviour of the distribution grid, the topology of which was not originally designed for bidirectional power flow;

and, narrowing down the level of detail, the shift from a centralised load-driven system to a system driven by distributed generation, in which it is crucial to take the behaviour of few or even single loads into account.

The following paragraph depicts a scenario in which each of these factors play a large role in developing a new automation concept.

1.1.1 Addressing challenging research questions for future grids: a scenario

Considering the development of a new automation concept for distribution grids, a typical example could be an automation infrastructure offering demand response services for several households.

A first critical question is whether or not realistic grid data is available. In most cases it is not; however, using a single reference grid as a replacement does not provide any assessment of how applicable a proposed solution is in general. If it is suitable for a limited set of grids, it would be helpful to categorise these in order to decide a priori whether it will be useful for a certain grid type. If the solution happens to be applicable for general grids, it is scientifically valuable to be able to prove this through testing on several grid types. A useful automation concept must be able to address a research question not for one grid but at least for a family of statistically similar grids.

A first requirement therefore is:

1. The need for a tool capable of generating a large number of plausible grids incorporating the possibility of varying typical topological parameters.

Supposing that these grids are available, the next point of focus is the fact that the generation portfolio in distribution grids mostly contains large shares of renewable energy sources such as photovoltaic modules. Therefore, the simulation of modern distribution grids is affected by a significant amount of parametric uncertainty, making the assessment of its effect on each candidate grid necessary. Especially in households, where not only electrical but also thermal dynamic behaviour must be taken into account, the uncertainty quantification
must be fast and easily applicable, since every single simulation may take up a long time span.

The second requirement consequently is:

2. The need for a tool capable of speeding up the process of simulation under uncertainty without complex mathematical adjustments.

Last but not least, the behaviour of single users must be included in the testing of the automation concept. While aggregated user behaviour may be known, it does not represent realistic challenges due to its smoothness.

The third requirement in the process of developing and testing the demand response concept may therefore be stated as

3. The need for a tool capable of generating load profiles of single users that are statistically plausible but challenge the grid operation realistically due to the presence of significant discontinuities.

Following each of these validation steps, the automation concept can be considered particularly robust since

- it is shown to be effective for either several types of grids or at least to be useful for a statistically significant number of similar grids;

- the stochastic variations due to uncertain parameters are known including most probable scenarios but also best and worst cases;

- it is capable of handling the abrupt behaviour of single loads.

### 1.1.2 Specifying the need for improved methodologies

Each of the above mentioned requirements must be addressed separately and in detail since they represent different scientific challenges.

**Grid Topologies** The topology of the grid strongly influences its robustness, connectivity, redundancy, etc. Certain events such as blackouts or thermal stress on specific lines can often be explained by the grid’s topological structure [9, 10]. Through the research on topologies of several real networks, the traditional mathematical discipline Graph Theory was revived in the past 20 years as a
new field of research called Complex Network Analysis due to the complexity of modern infrastructures like the Internet [9, 11, 12].

While transmission network topologies may be available from national agencies on demand, it is close to impossible to receive data on the plethora of distribution grids, which mostly belong to a specific regional operator. The synthetic IEEE reference grids often serve as test subjects for various investigations, e.g. in [13, 14]. In order to draw statistically correct conclusions and to evaluate the performance of analysis methods detached from specific topologies, it is crucial to study a large number of realistic grids. In [15], methods were proposed by Wang et.al. to generate statistically correct random grids for the purpose of testing communication strategies. The underlying topological statistics were partly taken from a small number of IEEE reference systems and from two U.S. American grids.

Considering the current evolution towards the grids of the future, this type of analysis will become more and more important in order to ensure that new methods and algorithms are not only suited for a specific topology. There is a need for the possibility of testing the same procedure on several statistically similar grids, sometimes varying certain topological measures.

Uncertainty Analysis In the past, the analysis of power grids relied on the assumption that the system behaviour during normal operation was deterministic. However, the need for stochastic analysis is constantly rising especially in the energy field due to increasing injection of renewable energy sources, some of which are intermittent and not deterministically predictable in nature. For example, photovoltaic generation is in principle deterministic for a certain latitude at perfect weather conditions, but strongly affected by temperature variations and clouds casting shadows on the panels.

Stochastic analysis can also be useful when testing the operation of equipment involving a range of possible parameter values. This situation is often given in the form of relative errors in components which arise due to the presence of uncertainties in the manufacturing process. Especially in complex systems, the interference between such uncertainties may be difficult to assess without sufficient analysis.

A strong motivation comes from the fact that while Monte Carlo simulations are easy to apply and achieve good results, they are not the most sophisticated
1.1 Motivation

methods in terms of computational burden. Oftentimes, Monte Carlo simulations are simply not applicable due to time restrictions, especially in real-time applications or computationally heavy simulation models. Faster approaches include Stochastic Collocation [16] and Polynomial Chaos Theory [17], both accompanied by a higher difficulty of application due to their deep mathematical fundament. Increasing the user-friendliness and analysing the benefits and drawbacks of such methods is a first step to wider usability.

Load Profiles for Testing and Analysis When assessing the potential electrical energy demand for consumers in the grid, standard load profiles (SLP) are often used. A SLP is a normalised curve representing the statistical electrical load of a consumer type for a certain period of time (e.g. a day), aggregated from a large number of consumers. In Germany, many different profile types are provided by the Bundesverband der Energie- und Wasserwirtschaft (BDEW)\textsuperscript{(i)}, the German Association of Energy and Water Industries. These profiles include industrial consumers differentiated by time of operation (9 to 5 as opposed to 24 hours) or specific trades such as bakeries or cinemas, as well as households or other uses like public lighting.

While SLPs are useful for global day ahead demand forecasts and optimisations which may be conducted e.g. by a distribution system operator with large centralised power plants for a high number of consumers, they cannot account for the physical behaviour of smaller sets of consumers for several reasons:

- single load profiles often have abrupt fluctuating characteristics, e.g. in a single household the turning-on of a large load such as a vacuum cleaner will produce a clear peak whereas the SLP is smooth;

- though the values of the SLP may be small at night and large in the early evening since they represent an average, a single consumer may behave completely differently;

- the deviation from reality due to the use of the same SLP for each consumer will be much more significant for a small set than for a statistically relevant number of consumers, following the law of large numbers.

\textsuperscript{(i)}https://www.bdew.de/
Chapter 1 Introduction

These facts imply that single consumer behaviour poses much larger challenges for physical simulation, control or energy balancing, since it is far from being smooth [18, 19, 20].

In order to account for the random behaviour of small sets of consumers, there is a need for methods to generate statistically correct load profiles.

1.2 Contributions of this thesis

This dissertation addresses all three requirements, creating a framework for an advanced process of validation for automation solutions in distribution grids. In particular,

1. an algorithm for generating artificial distribution grid models is presented based on concepts of Graph Theory and surveys on real distribution grids, thus enabling the testing of grid-related methods with several statistically similar grid topologies;

2. in order to tackle the surge of sources of uncertainty, a method is proven which enables the black box application of Polynomial Chaos theory in an automated way, providing a fast means of uncertainty propagation;

3. a theoretical framework for the generation of load profiles is designed which encompasses several types of consumers, exhibiting the abrupt behaviour of single loads while statistically approximating standard load profiles.

Each proposed method is theoretically developed and implemented in a MATLAB\(^{(ii)}\) Graphical User Interface (GUI). The discussed approach surpasses the traditional limited verification process based on the application on one single deterministic scenario.

1.3 Dissertation outline

Chapters 2–4 each present a method tackling one of the above mentioned challenges, starting at the most abstract level from the uncertainty propagation, continuing with the more concrete grid topologies and ultimately narrowing down to single consumption profiles.

\(^{(ii)}\)http://de.mathworks.com/
Chapter 2 deals with the development of the method of uncertainty analysis, following an overview of fundamentals. Major advantages and disadvantages are discussed in order to give a guideline for the application of this approach. The proposed method differs from those introduced in the other chapters due to the fact that despite the grid-related focus of this dissertation, it can be used interdisciplinarily. The following chapters concretely concern electrical grids.

In Chapter 3, the generation algorithm for random realistic network topologies is developed for medium voltage (MV) and low voltage (LV) grids. Electrical parameters are added to define admittance matrices for the developed grids, and a comparison with real grid data is conducted to demonstrate the performance of the method.

Finally, increasing the level of detail, Chapter 4 introduces the algorithm which produces single load profiles for testing purposes from a statistical load profile, for arbitrary types of consumers. Households are used as an example for implementation, and it is shown that an ensemble of generated load profiles matches the large scale statistics.

While specific conclusions are given for each methodology at the end of the respective chapter, a summarising conclusion and outlook is given in Chapter 5.
Chapter 2

A nonintrusive polynomial chaos approach to uncertainty analysis

Approaches often used when assessing the effects of uncertain parameters on a system are best case/worst case evaluations or a simple mean and variance calculation. These methods have in common that they give a feeling for a range of possibilities between certain limits, either the full range as in best/worst case studies or within a certain bandwidth of probability as in mean/variance analyses. However in these methods, there is no information content as to how all possible scenarios are distributed between the mentioned limits.

There are two major types of uncertainty: random (stochastic) behaviour and (statistical) uncertainty. The difference between stochastic variables such as wind speed and measurement or production uncertainty is that the former are subject to volatile behaviour whereas the latter are not intrinsically random:

- measurement uncertainty: a certain quantity has one specific accurate value, which only cannot be determined due to the limitations of measurement devices;

- production uncertainty: for each produced unit, the parameters do not change under unchanging conditions, and could be measured; however in large amounts, statistical deviations arise due to the inaccuracy of the production process.

One way of modelling these types of uncertainty is to interpret them as random
variables with probability density functions (PDF) that match their statistical distribution pattern. Depending on the system’s behaviour and the uncertain input parameters’ PDF, the scenarios may be skewed towards one of the limits instead of being uniformly distributed. These effects can be better understood with Monte Carlo (MC) methods, which were first developed by N. Metropolis and S. Ulam in [21], and have been widely applied in physics and engineering. All MC methods have in common the repetition of a deterministic simulation with input samples generated from the PDFs of the stochastic system parameters.

The principle is based on the law of large numbers, which states that the average of all results obtained from a repeated stochastic experiment converges to the theoretical expected value as the number of repetitions increases. This means that the histogram of output values aggregated from a sufficient number of MC simulations can be regarded as an approximation of the “real” behaviour. However, the fact that MC methods rely on the law of large numbers implies that the convergence rate of the mean value is slow at $\sqrt{n}$ for $n$ samples; i.e. a “sufficient” number of sampling runs may be in the hundred thousands depending on the spread of the probability distribution. Despite the beauty of simulating an uncertain system as though it were deterministic, this is the major drawback of this approach: depending on the system, one single simulation may take days.

More sophisticated methods which involve less samples include Polynomial Chaos (PC) and Stochastic Collocation. Both techniques describe stochastic parameters through polynomial expansions. The PC representation captures the randomness of the represented parameter in an orthogonal expansion with well-known polynomials, the coefficients of which are to be found. Stochastic Collocation on the other hand represents a random parameter with known coefficients, the interpolation polynomials are to be determined through the selection of a suitable set of collocation points [16].

This chapter focuses on PC theory. While earlier works concentrated on the further development of its mathematical fundament [22, 23], Monti et al. introduced the concept for applications in electrical engineering [24]. In the following,

- some basic terms of probability theory are recapitulated;
- the fundamentals of PC theory are summarised;
- the concept of nonintrusive PC (NIPC) is introduced, which makes the mathematically sophisticated concept easier to handle;

- the performance of NIPC is compared against MC simulations through examples;

- a developed GUI is presented;

- finally, the advantages and challenges of this concept are discussed in detail.

2.1 Fundamentals

2.1.1 Stochastic terms

The probability space $\Omega$ of a random experiment is defined as the set of all possible outcomes of the experiment. A $\sigma$-algebra $\mathcal{A}$ of $\Omega$ is a system of subsets of $\Omega$ with the following properties:

- $\Omega \in \mathcal{A}$;

- if $A_i$ is a sequence from $\mathcal{A}$, then

$$\bigcup_{i=0}^{\infty} A_i \in \mathcal{A} \text{ and } \bigcap_{i=0}^{\infty} A_i \in \mathcal{A};$$

- if $A \in \mathcal{A}$, then $\bar{A} \in \mathcal{A}$.

A probability $P$ can now be defined as a real-valued function on $(\Omega, \mathcal{A})$ with the properties

- $P(A) \geq 0, \forall A \in \mathcal{A}$;

- $P(\Omega) = 1$;

- for $A_i \in \mathcal{A}$,

$$P\left(\bigcup_{i=0}^{\infty} A_i\right) = \sum_{i=0}^{\infty} P(A_i).$$
2.1 Fundamentals

For a continuous random variable $X$ defined on $I \subseteq \mathbb{R}$, the *probability distribution function* (PDF) is defined as

$$g_X : x \mapsto P[X \leq x].$$

The PDF can now be used to define the *cumulative distribution function* (CDF)

$$G_X : x \mapsto \int_{-\infty}^{x} g_X(t)dt.$$

The first two moments of a distribution are known as the mean value and the variance, defined as

$$E[X] := \int_{I} xg_X dx,$$

$$Var := E[(X - E[X])^2] = \int_{I} x^2g_X dx - (E[X])^2.$$

These and other fundamental concepts of Stochastics can be found e.g. in [25].

2.1.2 Generalised polynomial chaos

Let $X$ be a continuous random variable with finite variance. Then, according to [23], $X$ can be represented as

$$X = \sum_{i=0}^{\infty} a_i \Phi_i(\xi). \quad (2.1)$$

Here, $\{\Phi_i\}_{i=0}^{\infty}$ is a set of orthogonal polynomials corresponding to the PDF of an artificial random variable $\xi$ and its support. An optimal matching in terms of type of distribution and polynomial basis was established by D. Xiu and G. Karniadakis in [23] and is listed in Table 2.1. The coefficient $a_i$ in (2.1) is the projection of $X$ onto the polynomial $\Phi_i$ with the PDF $g(\xi)$ as a weight:

$$a_i = \frac{1}{\langle \Phi_i^2 \rangle} \int_{\text{supp}(\xi)} X\Phi_i(\xi)g(\xi)d\xi. \quad (2.2)$$

The stochastic behaviour of $X$ is fully specified by the coefficients $a_i$, as in other orthogonal expansions like the Fourier series. Incidentally, the first coefficient
Chapter 2 A nonintrusive polynomial chaos approach to uncertainty analysis

$a_0$ is equal to the mean value of the respective distribution by definition because the first polynomial $\Phi_0$ is always a constant, reducing the computational effort of determining the PC coefficient since formulas are known for the mean values of most distributions. The main benefit of the PC representation can be interpreted in the following way: while the randomness of $X$ has been captured by the general terms $\Phi_i(\xi)$ with $i \in \mathbb{N}_0$ which are easily sampled without additional efforts, the coefficients determining $X$ specifically are now deterministic. This fact is highly important for uncertainty propagation and will be explored later in this chapter. When $X$ is a quantity depending on several random parameters $\xi_1, \ldots, \xi_n$, the PC expansion becomes

$$X = \sum_{i=0}^{\infty} a_i \Psi_i(\xi_1, \ldots, \xi_n), \quad (2.3)$$

with multivariate polynomials $\Psi_i$ constructed from each of the orthogonal polynomial sets which correspond to the single random variables $\xi_j$, and coefficients with multiple integrations

$$a_i = \frac{1}{\langle \Psi_i^2 \rangle} \int_{\text{supp}(\xi_1)} \cdots \int_{\text{supp}(\xi_n)} X \Psi_i(\xi_1, \ldots, \xi_n) \left( \prod_{j=1}^{n} g_j(\xi_j) \right) d\xi_1 \cdots d\xi_n. \quad (2.4)$$

The construction of multivariate polynomials will be detailed in Section 2.2.5.

<table>
<thead>
<tr>
<th>Distribution</th>
<th>Polynomials</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>Hermite</td>
<td>$(-\infty, \infty)$</td>
</tr>
<tr>
<td>Uniform</td>
<td>Legendre</td>
<td>$[-1, 1]$</td>
</tr>
<tr>
<td>Beta</td>
<td>Jacobi</td>
<td>$[-1, 1]$</td>
</tr>
<tr>
<td>Exponential</td>
<td>Laguerre</td>
<td>$[0, \infty)$</td>
</tr>
<tr>
<td>Gamma</td>
<td>Generalized Laguerre</td>
<td>$[0, \infty)$</td>
</tr>
</tbody>
</table>

Table 2.1: Correspondence of continuous distributions and orthogonal polynomials

2.1.3 Traditionally solving stochastic differential equations with PC

The application of PC classically involves a re-formulation of the system equations due to the PC expansion of all variables and parameters. This principle is
described in the following with the simple equation
\[
\frac{dy}{dt} = ky, \quad y(0) = y_0,
\]
where \( y \) is the system variable and \( k \) is a random parameter. The aim is to find out the stochastic behaviour of \( y \) through uncertainty propagation.

In a first step, all random quantities are represented with their PC expansion as described in Section 2.1.2:

\[
y(t) \approx \sum_{i=0}^{K} y_i(t) \Psi_i(\xi),
\]
\[
k \approx \sum_{i=0}^{M} k_i \Phi_i(\xi).
\]

Since only one random parameter \( k \) is involved, the polynomial expansion bases are identical and will be denoted as \( \Phi_i \) in the following. For the same reason, the truncation indices \( K, M \in \mathbb{N} \) are equal in this case, whereas \( K \) would be larger if more dimensions of random parameters were involved. The coefficients \( k_i \) can be calculated as in Equation 2.2 with the corresponding integration weight \( g(\xi) \), however the \( y_i \) are unknown. Once they have been determined, the random behaviour of \( y \) can be sampled from its expansion with the artificial random variable \( \xi \).

Substituting the PC expansions into the original equation, the system description becomes the set of differential equations
\[
\sum_{i=0}^{M} \frac{dy_i(t)}{dt} \Phi_i(\xi) \approx \sum_{i=0}^{M} \sum_{j=0}^{M} k_i y_j(t) \Phi_i(\xi) \Phi_j(\xi).
\]

Now, a projection onto the \( l \)-th polynomial dimension is performed on the equation for \( l = 1, \ldots, M \), i.e. the scalar product \( \langle \cdot, \Phi_i \rangle \) is applied for all \( l \), leading to
\[
\frac{dy_i(t)}{dt} \langle \Phi_i^2 \rangle \approx \sum_{i=0}^{M} \sum_{j=0}^{M} k_i y_j(t) \langle \Phi_i \Phi_j, \Phi_l \rangle, \quad l = 1, \ldots, M.
\]

thanks to the orthogonality of the polynomials \( \Phi_i \). It is noteworthy that the random variable \( \xi \) dissapears from this equation, as all polynomials are part of scalar products which produce deterministic numerical values.
Solving for the coefficients $y_i$ yields the explicit form of the PC expansion of the target variable $y$ including its stochastic behaviour, and in the bigger picture, a deterministic way of solving stochastic equations. This method is called the Galerkin projection [23, 26].

### 2.1.4 Gauss quadrature

For an $m \in \mathbb{N}$, a well-proven theorem says that there exist collocation points $c_0, \ldots, c_m \in (a,b) \subset \mathbb{R}$ and positive weights $w_0, \ldots, w_m$ such that

$$(b - a) \sum_{i=0}^{m} w_i f(c_i) \approx \int_{a}^{b} f(x) \omega(x) dx.$$ 

Here, $\omega(x)$ is a positive weighting function. Proofs of this theorem can be found e.g. in [27, 28]. This method of numerical integration is called Gauss quadrature. The choice of the collocation points $c_i$, the weights $w_i$ and the weighting function $\omega(x)$ are strongly related to sets of orthogonal polynomials. In particular, given a specific set of orthogonal polynomials $\{\Phi_i(x)\}_{i=0}^{\infty}$,

- the weighting function $\omega(x)$ corresponds to the integration weight which ensures that the polynomials are orthogonal:

  $$\langle \Phi_i, \Phi_j \rangle_{\omega} = \int_{a}^{b} \Phi_i(x) \Phi_j(x) \omega(x) dx = C \cdot \delta_{ij}, \quad C = \text{const.};$$

- for a quadrature rule of the $m$-th order, i.e. with $m + 1$ summands as above, the collocation points $c_0, \ldots, c_m$ are equal to the roots of $\Phi_{m+1}$;

- in the same setting, the weights $w_i$ can be calculated from the $\{c_j\}_{j=0}^{m}$ and $\omega(x)$.

At the same time, orthogonal polynomial bases are also related to stochastic distributions through the fact that their PDFs can serve as the weighting function $\omega(x)$ which establishes the orthogonality of the polynomials.

This essentially explains the correspondence of distributions and polynomials listed in Table 2.1, defining generalised PC in [23]. In this chapter, the interest lies specifically in a special case of Gauss quadrature, the so-called Gauss-Legendre quadrature which corresponds to the uniform distribution on $[-1,1]$. 

15
2.2 Non-intrusive single basis polynomial chaos

Here, the polynomial basis is the set of Legendre polynomials, the weighting function is \( \omega(x) = 1 \) and the evaluation interval is \([-1, 1]\):

\[
\int_{-1}^{1} f(x) \, dx \approx \sum_{i=0}^{m} w_i f(c_i).
\] (2.5)

Integrals defined on other intervals can be approximated with the same method by simply transforming the integrand’s argument.

The simplicity of Gauss-Legendre quadrature thanks to the constant weighting function is one of the main reasons why in the further course of this chapter all numerical integrations will concentrate on this specific formulation in order to put more focus on the aspect of usability.

2.2 Non-intrusive single basis polynomial chaos

In Section 2.1.3, it was shown that uncertainty propagation from stochastic inputs to output variables of a system can be quantified through the combination of PC expansions and Galerkin projections. The goal was to find the PC expansion coefficients of each target variable, as these determine its representation uniquely. While the method is elegant in the sense that it transforms stochastic system equations into an analytical deterministic problem, full knowledge of the mathematical system behaviour is necessary. Additionally, the Galerkin projection changes the formulation of the system equations, calling for a customised solver.

Non-intrusive polynomial chaos (NIPC) derives its name from the fact that it enables the use of PC while treating the evaluation of the system equations as a black box, thus not intruding on the solving process. All variations of NIPC have in common the numerical evaluation of the integrals needed to determine the PC expansion of target variables, instead of analytically solving them as in the Galerkin projection method. Different choices of collocation points and numerical integration methods are presented in [29], where this concept was introduced first, to the author’s best knowledge.

The numerical approximation of the integrals essentially splits the process of integration into two separate parts, one being the evaluation of the integrand at certain points and the other being the weighted summation of these values. The application of NIPC involves four main steps, where the separation of steps
3 and 4 is possible due to this split:

1. the PC expansion of the uncertain system parameters;
2. the selection of deterministic evaluation points;
3. the process of solving the system equations;
4. the post-processing to obtain the PC expansions and PDFs of system variables of interest.

The main point is that the “integration”, now a summation of weighted function values, can be conducted after deterministically simulating the system equations a number of times, depending on the number of collocation points.

### 2.2.1 Prerequisites

In the following, let the system behaviour over time depend on uncorrelated continuous random parameters $X_1, \ldots, X_n$ as follows:

\[
\begin{align*}
f : \Omega \times T &\rightarrow \Lambda, \\
(X_1, \ldots, X_n; t) &\mapsto f(X_1, \ldots, X_n; t) \\
&= (Y_1(t), \ldots, Y_N(t)),
\end{align*}
\]

where $\Omega := \Omega_1 \times \ldots \times \Omega_n$ is the probability space of the parameters, $T$ is the range of time and $\Lambda$ is the codomain of $f$. When the parameters are correlated, they must be transformed to a set of uncorrelated random variables before applying the steps in the following section. Additionally, it is assumed that the CDFs of all $X_i$ are invertible. For a dynamic system, $f$ can be a differential expression or a function which depends explicitly on the time $t \in T$; a time independent system is represented through a constant function $f$ which simply determines the relation between the system outputs $Y_1, \ldots, Y_N \in \Lambda$, $N \in \mathbb{N}$, and the random input parameters.

The overall goal is the same as in Section 2.1.3. In particular, for each $i \in \{1, \ldots, N\}$, the coefficients $y_{ij}$ of the PC expansion of $Y_i$,

\[
Y_i(t) \approx \sum_{j=0}^{K} y_{ij}(t) \Psi(\xi_1, \ldots, \xi_n),
\]
must be found, which determine its stochastic behaviour. Here, \( K \) is a truncation index and \( \{ \Psi_j \}_{j=0}^K \) is a set of multivariate polynomials which are constructed from the single polynomial bases corresponding to the single random variables \( \xi_l, l \in \{1, \ldots, n\} \). The single Legendre polynomials can be constructed recursively:

\[
P_0(x) = 1, \quad P_1(x) = x, \quad (k + 1)P_{k+1}(x) = (2k + 1)xP_k(x) - kP_{k-1}(x).
\]

Proofs can be found e.g. in [30].

<table>
<thead>
<tr>
<th>Index ( i )</th>
<th>Legendre polynomial ( \Phi_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>( x )</td>
</tr>
<tr>
<td>2</td>
<td>( \frac{1}{2}(3x^2 - 1) )</td>
</tr>
<tr>
<td>3</td>
<td>( \frac{1}{2}(5x^3 - 3x) )</td>
</tr>
<tr>
<td>4</td>
<td>( \frac{1}{8}(35x^4 - 30x^2 + 3) )</td>
</tr>
</tbody>
</table>

Table 2.2: The first 5 Legendre polynomials

Without loss of generality, it can be assumed that \( N = 1 \), i.e. there is only one output variable

\[
f(X_1, \ldots, X_n; t) = Y(t),
\]

since the procedure of determining the PC expansion for each of the \( Y_i \) is decoupled. The subsequent four sections will present the details of the four steps under this assumption.

### 2.2.2 Step 1: expressing continuous random parameters with a single-basis polynomial chaos

While one branch of polynomial chaos, the so-called arbitrary polynomial chaos, has evolved from generalised polynomial chaos to optimally match customized polynomial bases for arbitrary probability distributions [31], the use of a single basis is especially valuable in terms of user-friendliness. The approach developed in the following is based on the latter, while taking into account its advantages and challenges. Let \( n \) be the number of stochastic parameters in the system of interest. Given the joint probability space \( \Omega \) of the \( n \) random
variables $X_1, \ldots, X_n$ of the system, the following mapping can be defined:

$$h : \mathcal{U}(-1,1)^n \rightarrow \Omega,$$

$$\begin{pmatrix} \xi_1 \\ \vdots \\ \xi_n \end{pmatrix} \mapsto \begin{pmatrix} X_1(\xi_1) \\ \vdots \\ X_n(\xi_n) \end{pmatrix} := \begin{pmatrix} \sum_{j_1=0}^{\infty} a_{j_1} \Phi_{j_1}(\xi_1) \\ \vdots \\ \sum_{j_n=0}^{\infty} a_{j_n} \Phi_{j_n}(\xi_n) \end{pmatrix}. \tag{2.8}$$

Assuming that $X_1, \ldots, X_n$ are independent, this mapping represents their PC expansion with $n$ independent uniform random variables $\xi_1, \ldots, \xi_n \sim \mathcal{U}(-1,1)$. Following the matching of polynomial sets and continuous distributions in 2.1, $\{\Phi_i\}_{i=0}^{\infty}$ are the Legendre polynomials which form an orthogonal basis of the Hilbert space $L^2(-1,1)$. In the following, the $X_i$ will be approximated by the truncated PC expansions, so it is assumed that with a fixed $M \in \mathbb{N}$

$$h\left(\begin{pmatrix} \xi_1 \\ \vdots \\ \xi_n \end{pmatrix}\right) = \begin{pmatrix} \sum_{j_1=0}^{M} a_{j_1} \Phi_{j_1}(\xi_1) \\ \vdots \\ \sum_{j_n=0}^{M} a_{j_n} \Phi_{j_n}(\xi_n) \end{pmatrix}. \tag{2.9}$$

It is important to note that $\xi_1, \ldots, \xi_n$ and $\{\Phi_i\}_{i=0}^{\infty}$, respectively, need not correspond to the distributions of $X_1, \ldots, X_n$; expansions with other polynomials converge as well, albeit with a slightly impaired convergence rate [32]. The choice of one single polynomial basis for all stochastic parameters leads to a significant simplification of the application of PC, since an adjustment to each single random variable can be avoided. However, when the distribution of $X$ does not match $\mathcal{U}(-1,1)$, the coefficients

$$a_{ij} = \frac{1}{\Phi_i^2} \int_{-1}^{1} X_i \Phi_i(x) d\xi_i,$$

cannot be determined straightforwardly due to differing probability spaces. This problem is solved by applying the inverse probability integral transform theorem which was proven e.g. in [33]. In particular, the probability integral transform theorem states that for any continuous random variable $X$ with CDF $G_X$, the random variable

$$\hat{\xi} = G_X(X)$$
2.2 Non-intrusive single basis polynomial chaos

is uniformly distributed on [0,1]. The inverse probability transform theorem equivalently states the inverse implication: given a uniform random variable \( \hat{\xi} \sim \mathcal{U}(0,1) \) and a CDF \( G_X \) of an arbitrary continuous random variable \( X \), \( G_X^{-1}(\hat{\xi}) \) has the same distribution as \( X \). Therefore, uniform random variables can be linked to arbitrary continuous distributions as long as their CDF is invertible.

So in the case where \( X \) is not uniformly distributed on \([-1,1]\), it can be substituted by the identically distributed random variable \( G_X^{-1}(\xi) \), slightly modifying the statement above with

\[
\xi = \frac{1}{2} \hat{\xi} + \frac{1}{2}
\]

in order to account for the shift from \( \mathcal{U}(0,1) \) to \( \mathcal{U}(-1,1) \). In this way, the PC expansion with uniform random variables and Legendre polynomials can be computed for any continuous random variables which fulfill the above mentioned criteria.

Given the function \( h \) in (2.8) and (2.9), the system function \( f \) as defined in (2.6) can now directly be linked to the random variables \( \xi_1, \ldots, \xi_n \):

\[
f(X_1, \ldots, X_n; t) = f(h(\xi_1, \ldots, \xi_n); t).
\] (2.10)

2.2.3 Step 2: selecting deterministic evaluation points for the random parameters

According to the PC mapping detailed in the last section, it is known that

\[
X_i(\xi_i) \approx \sum_{j=0}^{M} a_{ij} \Phi_i(\xi_i), \quad i = 1, \ldots, n.
\]

As stated earlier, the goal is to determine the coefficients

\[
y_i = \frac{1}{\langle \Psi_i^2 \rangle} \int_{[-1,1]^n} Y \Psi_i(\xi_1, \ldots, \xi_n) d(\xi_1, \ldots, \xi_n)
\]

of the PC expansion of the target variable \( Y = f(X_1, \ldots, X_n; t) \). Using (2.10) and Gauss-Legendre quadrature with the collocation points \( (c_1, \ldots, c_m) \) and
Chapter 2 A nonintrusive polynomial chaos approach to uncertainty analysis

<table>
<thead>
<tr>
<th>Order</th>
<th>Collocation points</th>
<th>Weights</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>-0.5773502692, 0.5773502692</td>
<td>1, 1</td>
</tr>
<tr>
<td>3</td>
<td>-0.7745966692, 0, 0.7745966692</td>
<td>0.5, 0.5</td>
</tr>
<tr>
<td>4</td>
<td>-0.8611363116, -0.3399810436, 0.8611363116</td>
<td>0.3478548451, 0.6521451549</td>
</tr>
</tbody>
</table>

Table 2.3: Collocation points and weights for Gauss-Legendre quadrature up to $m = 4$ (extracted from [34])

weights $(w_1, \ldots, w_m)$ as defined in Section 2.1.4, this becomes

$$y_i(t) = \frac{1}{\langle \Psi_i^2 \rangle} \int_{[-1,1]^n} (f(h(\xi_1, \ldots, \xi_n); t)\Psi_i(\xi_1, \ldots, \xi_n)d(\xi_1, \ldots, \xi_n)$$

$$\approx \frac{1}{\langle \Psi_i^2 \rangle} \sum_{l_1=0}^{m} \cdots \sum_{l_n=1}^{m} f(X_1(c_{l_1}), \ldots, X_n(c_{l_n}); t)w_{l_1} \cdots w_{l_n}\Psi_i(c_{l_1}, \ldots, c_{l_n}).$$

(2.11)

In the second statement of this equation, all quantities are known except for the function values (or solutions of system equations) $f(X_1(c_{l_1}), \ldots, X_n(c_{l_n}); t)$. Fortunately, the easily obtained points

$$p_{ij} := X_i(c_j) = \sum_{l=0}^{M} a_{il}\Phi_l(c_j), \quad i = 1, \ldots, n, \quad j = 1, \ldots, m$$

(2.12)

are the only input required to solve the problem. These points $p_{ij}$ serve as input for the system equations in the next step, the set $\{p_{i1}, \ldots, p_{im}\}$ substituting the random parameter $X_i$.

2.2.4 Step 3: solving the system equations

The solving process, i.e. the evaluation of the system rule $f$, stays untouched by the uncertainty analysis in contrast to the classical procedure described in
2.2 Non-intrusive single basis polynomial chaos

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Parameter</th>
<th>$X_1$</th>
<th>$X_2$</th>
<th>...</th>
<th>$X_n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>$p_{11}$</td>
<td>$p_{21}$</td>
<td>...</td>
<td>$p_{n1}$</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>$p_{11}$</td>
<td>$p_{21}$</td>
<td>...</td>
<td>$p_{n2}$</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td>...</td>
<td>...</td>
<td></td>
<td>...</td>
</tr>
<tr>
<td>$m$</td>
<td></td>
<td>$p_{1m}$</td>
<td>$p_{2m}$</td>
<td>...</td>
<td>$p_{nm}$</td>
</tr>
<tr>
<td>$m^n$</td>
<td></td>
<td>$p_{1m}$</td>
<td>$p_{2m}$</td>
<td>...</td>
<td>$p_{nm}$</td>
</tr>
</tbody>
</table>

Table 2.4: Input point combinations for simulation scenarios

Section 2.1.3, and therefore is a step which is to be conducted externally.

The main difference from a purely deterministic solving procedure is the number of times the system must be solved in order to obtain all integration points for the output $Y$. In particular, the system must be simulated with all $m^n$ combinations of input points $p_{ij}$ as shown in Table 2.4. The output of this step is an $m^n \times 1$-vector of solutions of $f$ for the $m^n$ input scenarios. In dynamic systems, each solution will be time varying, as opposed to constant values in time invariant systems.

2.2.5 Step 4: post-processing the simulation results

When constructing the stochastic behaviour of the target variable $Y$, the polynomial basis must be adapted to multivariate polynomials, as introduced in Equation 2.7, in order to account for the number of parameter dimensions which $Y$ depends on. These polynomials can be constructed as tensor products of the original Legendre polynomials, in which each of the $n$ factors corresponds to one of the $X_i$:

$$
\Psi_i(\xi_1, \ldots, \xi_n) = \Phi_{i1}(\xi_1) \otimes \ldots \otimes \Phi_{in}(\xi_n).
$$

(2.13)

The truncation index $K$ of the PC expansion of $Y$ depends on the truncation index $M$ of the single parameter PC expansions as well as the number $n$ of parameters:

$$
K = \frac{(n + M)!}{n!M!}.
$$

(2.14)
Table 2.5 shows an example for the multivariate polynomials $\Psi_i$ for $n = 2$ and $M = 3$. In this case, it can be seen that e.g.

$$
\Psi_6(\xi_1,\xi_2) = \Phi_1(\xi_1) \otimes \Phi_2(\xi_2) = \frac{1}{2} \xi_1(3\xi_2^2 - 1),
\Psi_7(\xi_1,\xi_2) = \Phi_2(\xi_1) \otimes \Phi_0(\xi_2) = \frac{1}{2}(3\xi_1^2 - 1).
$$

<table>
<thead>
<tr>
<th>Index $i$ of polynomial</th>
<th>$i_2$</th>
<th>$i_1$</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$0 + 0 \leq K$</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>$0 + 1 \leq K$</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>2</td>
<td>$0 + 2 \leq K$</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>3</td>
<td>.</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0</td>
<td>.</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>.</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>2</td>
<td>.</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>0</td>
<td>$1 + 3 \geq K \Rightarrow$ skip</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>1</td>
<td>$2 + 1 \leq K$</td>
</tr>
<tr>
<td>9</td>
<td>3</td>
<td>0</td>
<td>$2 + 2 &gt; K$ and $2 + 3 &gt; K \Rightarrow$ skip</td>
</tr>
</tbody>
</table>

Table 2.5: Indices of factors for polynomials with $n = 2$ and $K = 3$

The PC coefficients

$$
y_i(t) \approx \frac{1}{\langle \Psi_i^2 \rangle} \sum_{l_1=0}^{m} \cdots \sum_{l_n=1}^{m} f(X_1(c_{l_1}),\ldots,X_n(c_{l_n});t)w_{l_1} \cdots w_{l_n} \Psi_i(c_{l_1},\ldots,c_{l_n})
$$

can now be obtained from Table 2.3, Section 2.2.4 and (2.13). Finally, the stochastic behaviour of $Y$ is known:

$$
Y(t) = \sum_{i=0}^{M-1} y_i(t) \cdot \Psi_i(\xi_1,\ldots,\xi_n).
$$

(2.15)

The distribution can easily be visualised by sampling $\xi_1,\ldots,\xi_n$. 

23
2.3 Application examples

Having introduced the theoretical aspects of nonintrusive PC, this chapter proceeds to present the application. The first two sections, focusing on the representation of unprocessed random parameters and some algebraic expressions thereof, purely demonstrate the performance of nonintrusive PC with direct sampling or MC simulations as references. The third and final group of examples then proceeds to show applications with a dynamic and nonlinear electrical system.

2.3.1 Approximating nonuniform random variables

In order to demonstrate that meaningful samples of nonuniform random variables can be made with the non-matching Legendre polynomials, the standard normal (Gaussian) distribution $\mathcal{N}(0,1)$, Gamma distribution $\text{Gamma}(10,7)$ and exponential distribution $\text{Exponential}(3)$ are chosen as examples.

In a way, the normal distribution represents the most extreme case compared to the uniform distribution: while the uniform distribution has a finite support, the normal distribution has an infinite support and long tails; and while the uniform distribution is constant, the normal distribution has a strong peak. The Gamma distribution shows the approximation of a similar distribution which is more skewed towards one side, whereas the exponential distribution in contrast has one extreme peak and a long tail.

Figure 2.1 shows how higher truncation orders of the Legendre PC expansion add to the tail for the approximated normal distribution, though as expected a strong tail cannot be achieved fast. For each of these approximations, the PC expansion was calculated with the respective truncation order, followed by a sampling with 100000 uniform random numbers as arguments to the Legendre polynomial basis. For an order 6 approximation, the data exhibits a typical peak for a normal random variable.

While the generated sample displays untypical behaviour at the boundaries, the distribution fit presented in Figure 2.2 shows nearly identical statistical behaviour compared to a data set, also with 100000 samples, generated directly from the standard normal PDF. Similar results could be achieved for a Gamma distribution and an exponential distribution, as seen in Figures 2.3, 2.4 and Table 2.6, where a small relative error is presented for the fitted distribution.
parameters compared to the original distribution. The distribution fitting was performed with the Statistics Toolbox of MATLAB. Due to the sufficient number of samples, the statistical behaviour hardly varies, the parameters only changing in the range of 0.1%. While the sampled data may not behave perfectly at the theoretical boundaries, the fit is comparable to the actual statistical behaviour.

<table>
<thead>
<tr>
<th>Theoretical distribution</th>
<th>Distribution fitted to sampling from PC expansion</th>
<th>Relative error of parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal $\mu = 0$, $\sigma = 1$</td>
<td>$\mu = 0.00059$, $\sigma = 0.97232$)</td>
<td>$\varepsilon_\mu \approx 0%$, $\varepsilon_\sigma \approx 3%$</td>
</tr>
<tr>
<td>Gamma $a = 10$, $b = 7$</td>
<td>$a = 10.59710$, $b = 6.60674$</td>
<td>$\varepsilon_a \approx 6%$, $\varepsilon_b \approx 6%$</td>
</tr>
<tr>
<td>Exponential $\mu = 3$</td>
<td>$\mu = 3.02667$</td>
<td>$\varepsilon_\mu \approx 0.8%$</td>
</tr>
</tbody>
</table>

Table 2.6: Comparison of distribution of PC samples with the original distribution
2.3 Application examples

Figure 2.2: Standard normal distribution: comparison of PC samples with samples from the original distribution, truncation order $M = 6$

2.3.2 Nonintrusive PC simulation of algebraic expressions of random variables

Though differential equations are more common in describing the behaviour of physical systems, algebraic expressions play a large role as well, e.g. Betz’s law for the calculation of wind power, or more fundamentally, Kirchhoff’s laws. In order to show both summations and multiplications, the expressions

\[
Y_T = X_{\gamma_1} - X_{\gamma_2} \cdot X_{\gamma_3}
\]
\[
Y_E = X_{e_1} + 2X_{e_2} \cdot X_{e_3}
\]

are simulated, where

\[
X_{\gamma_1}, X_{\gamma_2}, X_{\gamma_3} \sim Gamma(10,7),
\]
\[
X_{e_1}, X_{e_2}, X_{e_3} \sim Exponential(3).
\]

For this purpose, each of the random parameters are represented with 6 collocation points each as described in Section 2.2.3. Figures 2.5 and 2.6 show that the
Figure 2.3: Gamma(10,7) distribution: comparison of PC samples with samples from the original distribution, truncation order $M = 6$

Figure 2.4: Exponential(3) distribution: comparison of PC samples with samples from the original distribution, truncation order $M = 6$
results are statistically comparable with those of a MC simulation with 100000 samples per expression, though for both of them only 216 calculations each had to be computed. Furthermore, though the random parameters are represented with the same logic as in the last section, the problem of cut-off tails has been canceled due to coupling effects. This can be explained through the fact that the probability of the values at the cut off boundaries of the PC approximation are already considerably small, so the multiplication of as little as two such random variables produces extreme values less probably, resulting in longer tails comparable to those of the real behaviour.

![Figure 2.5: MC and NIPC simulation of $Y_T$](image)

2.3.3 NIPC simulation of a nonlinear dynamic system with uncertain parameters

In order to highlight the ability of NIPC to handle highly nonlinear system behaviour, the single-phase rectifier depicted in Figure 2.7 is simulated, assuming that the capacitor and resistor have a Gaussian manufacturing error with the parameters shown in Table 2.7. The target variable is chosen to be the load voltage over the resistor $R_1$. 
Chapter 2  A nonintrusive polynomial chaos approach to uncertainty analysis

Figure 2.6: MC and NIPC simulation of $Y_E$

Figure 2.7: Single-phase rectifier
2.3 Application examples

<table>
<thead>
<tr>
<th>Component</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Voltage source</td>
<td>100 V, 50 Hz, resistance 0.01 Ω</td>
</tr>
<tr>
<td>Inductor</td>
<td>1 μH, resistance 0.1 Ω</td>
</tr>
<tr>
<td>Diodes</td>
<td>On-resistance 0.001 Ω, off-resistance 100k Ω</td>
</tr>
<tr>
<td>Capacitor</td>
<td>0.01 F±10%</td>
</tr>
<tr>
<td>Load resistor</td>
<td>10 Ω ± 10%</td>
</tr>
</tbody>
</table>

Table 2.7: Component values for the single-phase rectifier

Since two uncertain parameters are identified, the total number of nonintrusive PC simulations will be $m^2$ where $m$ is the number of collocation points per parameter. Choosing $m = 6$, it suffices to run the simulation with just 36 different input scenarios. The external system simulation of $V_{R_1}$ for $t \in [0,0.1\text{s}]$ in VTB$^{(i)}$ produced the 36 corresponding solutions which are shown in Figure 2.8, exhibiting a nonlinear zigzag behaviour after the initial transient phase from $t = 0$ until about $t = 0.005\text{s}$.

![Voltage over Resistor R1](http://vtb.engr.sc.edu/vtbwebsite/)

Figure 2.8: Thirty-six scenario solutions for the single-phase rectifier

Once in steady state, the voltage bounces periodically between local maxima and minima, and even with only the plot of 36 solutions a tendency can be seen that uncertainty increases as the voltage heads towards a minimum. Choosing to analyse the uncertainty for $t = 0.01\text{s}$ to account for the first fall after transient phase, at $t = 0.15\text{s}$ for a maximum and at $t = 0.023\text{s}$ for a minimum, the PC expansion of $V_{R_1}$ is calculated for each of these steps, yielding the sampled

$^{(i)}$http://vtb.engr.sc.edu/vtbwebsite/
distributions presented in Fig 2.9. While the resulting uncertainty is relatively small in the maxima of $V_{R_1}$, it is noteworthy in the minima that not only is the bandwidth of possible voltages much larger, but there is a longer tail towards lower values.

As the uncertainty considered in this example is based on production errors, it needs to be taken into account that the resulting distributions would not be valid for a single circuit, once its component values have been measured in detail. However, this type of analysis is especially interesting if either the component values for a specific circuit are unknown and an assessment is needed, or if the single-phase rectifier is mass-produced with the same type of capacitors and resistors and therefore cannot be measured for every single instance.

![Figure 2.9: Distribution of $V_{R_1}$ at $t = 0.01$ s, $t = 0.015$ s and $t = 0.023$ s](image)

**2.4 Graphical user interface for NIPC**

In order to enhance the usability aspect of NIPC, a GUI was developed in MATLAB. The GUI makes use of the fact that the actual system simulations can be fully independent of the uncertainty analysis, providing only the necessary simulation input for the random parameters and the possibility to generate the distribution of selected simulation outputs. It is separated into two parts as
shown in Figure 2.10: the pre-processing of random parameters (Steps 1–2 in Section 2.2) and the post-processing of simulation results (Step 4).

![Diagram showing the process of nonintrusive PC analysis with GUIs]

Figure 2.10: Schematic of nonintrusive PC analysis with GUIs

The pre-processing GUI allows the user to select several random variables and their parameters from the list of all distributions provided by MATLAB, e.g. the beta, gamma, normal or uniform distribution, among others. It then calculates the PC coefficients for each random variable with the inverse probability transform. Next, a file is produced containing all necessary scenarios according to the selected random parameters and number of collocation points per parameter, which can then be used as input to the external simulator.

The post-processing GUI loads the simulation outputs corresponding to a single target variable to calculate its PC expansion via numerical integration, the output being a histogram of the target variable behaviour. The user can adjust the sampling number. The process is repeated for all target variables of interest.

Furthermore, another connector GUI was developed as an interface between the NIPC GUI and an external simulation. While this GUI was developed to match random parameters in the NIPC GUI with simulation results from the Modelica\(^{(ii)}\)-based simulation environment SimulationX\(^{(iii)}\), it can be extended for any simulation software which has a COM interface.

The functionalities of all three GUIs were described in detail in [35].

\(^{(ii)}\text{https://www.modelica.org/}\)
\(^{(iii)}\text{http://www.simulationx.com/de/}\)
Chapter 2 A nonintrusive polynomial chaos approach to uncertainty analysis

2.5 Range of applicability

When applying single-basis NIPC, some considerations need to be made about its advantages and disadvantages. While the advantages are all attributed to aspects of usability, the disadvantages need special attention since they are associated with the applicability of the method. A comparison of some of these aspects was also presented in [36].

2.5.1 Advantages

This section summarises the six main benefits of the NIPC method introduced in Section 2.2.

Number of simulation runs The main advantage of NIPC compared to MC simulations is the small number of required simulation runs, for systems with few random parameters. While the accuracy of MC simulations increases with the number of repetitions, NIPC always requires the same number of system solutions for a certain level of accuracy. In a system with only one random parameter this may mean that the same results can be achieved with less than 10 simulation runs instead of a MC simulation with 100000 samples. On the
other hand, the number of required samples is independent of the number of random parameters in MC simulations, whereas this relationship plays a great role in NIPC, as will be discussed in Section 2.5.2.

In contrast, the system only has to be solved once for classical PC methods. However, this comparison is not meaningful, as the focus in classical PC is not on simulation runs but the computational effort rather goes into the mathematically complicated re-formulation of system equations, leading to the next advantage of NIPC.

**Black box application** The possibility of applying the concept of PC in black boxes solves a major drawback for the wider application of classical PC concepts: the necessity to adjust the system formulation and thereby to meddle with the solving process.

In both classical and nonintrusive PC, the goal is to determine the PC coefficients of system variables. The complication in classical PC methods arises from the need to substitute all random quantities in the system equations with their PC expansion, resulting in systems of equations involving more variables. By decoupling the simulation from the process of calculating the unknown PC coefficients, NIPC renders unnecessary the interference with system equations and solvers.

This ultimately frees the user from the necessity to study the fundamentals of PC theory: each of Steps 1–4 can be automated, and furthermore, the preprocessing of random parameters (Steps 1 and 2) as well as the post-processing of system solutions (Step 4) can be completely separated from the actual system simulations.

The advantage of nonintrusive stochastic analysis is also shared by MC simulations, albeit with a significantly larger number of repetitions, depending on the number of random variables.

**Independent input scenarios** A benefit shared with MC simulations is the fact that the single system solutions can be parallelised, since the input scenarios are fully independent of each other. This leads to a further enhancement of the already significantly smaller amount of time needed compared to MC analysis, and may be an important plus factor especially in computationally heavy simulations.
Deterministic representation  The application of NIPC essentially results in representing random variables with a small set of deterministic collocation points for further processing. Once these have been calculated for a specific PDF, they never have to be computed again, meaning that all necessary information on something as complex as the stochastic behaviour of parameters can be stored simply in a vector with a small number of representative points.

Single polynomial basis  Limiting the PC representation of all random quantities to expansions based on a single polynomial basis further simplifies the overall process compared to the optimal matching of polynomial bases and random parameters in generalised PC. In this way, the handling of random parameters becomes more standardised. Especially in the context of NIPC, the choice of Legendre polynomials and uniform random variables to represent all uncertainties comes in handy for three reasons:

1. it is easy to relate any continuous random variable with an invertible CDF to a uniformly distributed random variable through the inverse probability integral transformation;

2. limiting the probability space to a standardised finite interval simplifies the application of numerical integration, as the most simple Gauss-Legendre quadrature is defined on a finite interval;

3. using uniform random variables in polynomial expansion basis enables a straightforward sampling of target variables once their PC coefficients are known, thanks to the ease of generating uniform random numbers.

However, while in principle the PC expansion with any orthogonal polynomial basis will converge [37], the fact that a mismatch between random variables and polynomial basis calls for a higher truncation order must not be neglected.

Nevertheless, for several non-uniform distributions, PC expansions with Legendre polynomials and uniform variables truncated at $M = 6$ already produced results with reasonable fits compared to the respective PDFs, as seen in Section 2.3. Taking this into account, the choice of a single polynomial basis can be justified also for suboptimal matches with random variables.

When not only the accuracy of the PDF fit but also the accuracy of the sampled data itself is an important focus, the process can be adjusted simply
by using the matching polynomial sets for each random parameter and using different numerical integration formulae.

**Compatibility with nonlinear systems** A feature of the presented NIPC method which is useful for applications on modern power systems is the ability to propagate uncertainty even in highly nonlinear systems, as seen using the example of the single phase rectifier. This fact is especially important since power electronics are omnipresent in power systems, accompanied by their inherently nonlinear behaviour due to switching.

### 2.5.2 Disadvantages

While the benefits of NIPC are obvious, it must first be determined whether or not the method is suitable for the system of interest, and whether or not the application of NIPC pays off against MC simulations. The following points detail the limits of applicability to this approach of uncertainty analysis.

**Curse of dimensionality** The probably most problematic downside is not theoretical in nature, but rather a numerical challenge: as seen in (2.4), the more uncertain parameters are involved, the more nested integrals must be solved when determining the PC coefficients. However, the numerical integration process with product rules such as the Gauss quadrature is expensive, with a computational burden growing exponentially for larger numbers of uncertainties [38]. This also results in the necessity to consider a tradeoff between accuracy of the numerical integration in NIPC as well as the number $s$ of simulation runs compared to a Monte Carlo type analysis:

$$s = m^n,$$

where $m$ is the order of the quadrature rule and $n$ is the number of random system parameters. This means that it pays off to apply this NIPC method while the number of desired MC simulations is larger than $s$. The curse of dimensionality can be slightly contained by using other numerical integration approaches which are more suitable for high-order integrals, e.g. Smolyak sparse grids [16].
**Type of random parameters**  This approach cannot be used to analyse systems with discrete uncertain parameters. This is due to the fact that the black box application is established by substituting analytical integration with numerical integration, i.e. can only be defined when all uncertainties are continuous. Classical PC methods can be applied nevertheless, since the PC expansion can be defined for discrete random variables as well with analogous definitions with sums instead of integrals for the PC coefficients [39].

**Choice of quadrature rule**  One effect which must be considered is the fact that the Legendre polynomials appear in the PC representations as expansion polynomials while at the same time the numerical integration procedure uses the roots of Legendre polynomials as collocation points. When the truncation order \( n \) of the expansion is lower than the chosen collocation order \( m \), the numerical integration causes artificial zeros in the \( m+1 \)-th summand of the PC representation which contains \( \Phi_m \), since the collocation points are the roots of \( \Phi_m \). For example, the integral

\[
\int_{-1}^{1} \Phi_4(x) \cdot (x^4 - x^3) \, dx = \frac{16}{315}
\]

would produce a false value when integrated numerically with Gauss-Legendre quadrature with \( m = 4 \):

\[
\int_{-1}^{1} \Phi_4(x) \cdot (x^4 - x^3) \, dx \approx \sum_{i=1}^{4} \Phi_4(x_i)(x_i^4 - x_i^3)w_i = 0.
\]

This problem however can easily be circumvented by either setting \( n > m \) or by using another type of quadrature rule on finite intervals, e.g. the Gauss-Tchebychev quadrature.

### 2.6 Conclusion

In this chapter it was shown that NIPC provides an efficient method for the stochastic analysis of systems with uncertainties, cutting down the number of necessary samples from the range of \( 10^5 \) for typical MC simulations to a significantly smaller number of simulations when the number of random parameters is sufficiently small.
The results produced by NIPC were shown to be comparable to large MC simulations which may be considered as the real behaviour of the investigated variables. The application of Polynomial Chaos theory was significantly simplified by substituting Galerkin projections by the use of numerical integration, therefore creating means to automate the procedure. A tool was developed providing such an automation, making the in-depth knowledge of the mathematical foundations unnecessary for most users.

It will be beneficial to conduct stochastic analyses especially in complex systems to account for unpredictable interferences between several random parameters, both stochastic (e.g. wind or solar energy) and statistical (components with manufacturing errors). An interesting field of research extending from this study may be the design of systems which limit the overall effect of stochastic behaviour, combining the traditional method of simulation based design [40] with NIPC theory.
Chapter 3

Statistical generation of distribution grid models

While many studies have been published focusing purely on the statistical properties of existing and artificial power grid topologies, these statistics and the variety of topologies are seldomly considered in the testing of new analysis methods such as control algorithms. Additionally, the focus of topological studies has always been mainly on the transmission network. Several topological studies arose following major catastrophic events such as the U.S. and Italian black outs in 2003. In studies such as [41, 42, 43], statistical metrics of the American and Italian power grid were analysed especially under the aspect of vulnerability under attack, showing how great a role the actual topologies of the studied grids played in the considered scenarios. Pagani et. al. list over 30 similar studies by different groups on high voltage grids in their survey [9], which mostly try to find a consensus on universal properties in grid structures. The same group of researchers have published a study on real low and medium voltage grids in the north of Netherlands [10]. Major previous efforts to generate grid models based on principles of Graph Theory can be found in [15, 44, 45], which are based on spatial point processes in order to include a distance reference.

The focus of the distribution grid generation algorithm presented in this chapter differs from that of the above mentioned topological studies in the sense that the goal is not to define “the” characteristic power grid topology whose existence
is arguable [44]. Instead, the aim is to enable the generation of several types of grids, especially in the MV level, while controlling relevant statistical properties. While the developed approach also utilises spatial point processes, the main novelty is the inclusion of the so-called joint degree distribution (JDD) which is a distinguishing topological feature and is manipulated by encouraging specific types of connections during the process of constructing. In order to facilitate the assignment of various distances and electrical properties, the unit square $[0, 1] \times [0, 1]$ was selected for the geographical definition of the topology.

3.1 Fundamentals

In this section, some of the main principles of Complex Network Analysis and its underlying academic field, Graph Theory, are summarised. Most of these concepts are documented in detail in several fundamental books on Graph Theory and Network Science, e.g. [11, 46].

3.1.1 Definitions

A graph is a 2-tuple $(V, E)$ with a set $V$ of vertices or nodes and a set $E \subseteq V \times V$ of edges or links, equipped with an embedding mapping function defining the connection between nodes:

$$f : V \times V \rightarrow E, \{v_1, v_2\} \mapsto e_{v_1, v_2} := (v_1, v_2),$$

in the case of a directed graph in which the orientation of the links is important, or

$$f : V \times V \rightarrow E, \{v_1, v_2\} \mapsto e_{v_1, v_2} = \{v_1, v_2\},$$

for an undirected graph in which $e_{v_1, v_2}$ denotes the same link as $e_{v_2, v_1}$. It is assumed that no links are duplicate or loops, i.e. links which connect a vertex with itself. The number of nodes will be denoted as $N := |V|$ and the number of edges as $M := |E|$.

It is quite common for the links of a graph to carry some information. This is captured in the so-called weight function $W : E \rightarrow \mathbb{R}$ or $\mathbb{C}$ which assigns values to all links. The weights may represent distance or cost, or in this chapter,
Chapter 3 Statistical generation of distribution grid models

electrical properties of the links.

**Nodes, Links and Degrees**  Two vertices $v_1, v_2 \in V$ are called *adjacent* if there exists a link $e_{v_1, v_2} \in E$, i.e. if they are connected. Two links are *incident* if they start from or end at the same node. Incidence can also be defined for pairs of nodes and links in a similar way. A link $e$ is incident to a node $v$ and vice versa, if $e$ is connected to $v$.

The set $\{v' \in V : \exists e_{v_1, v_2} \in E\}$ of nodes which are adjacent to $v \in V$ are called *neighbours* of $v$. The graph $\Gamma_v = (V(\Gamma_v), E(\Gamma_v))$ containing all neighbours of $v$ and links between them is called the *neighbourhood* of $v$.

The graph in which all $N$ nodes are adjacent is called the complete graph $K_N$, which has $\left( \frac{N}{2} \right)$ edges.

Weights can also be derived as *weighted degrees* for nodes from their incident links:

$$W(v) := \sum_{v' \in V(\Gamma_v)} W(e_{v,v'}) \; \forall v \in V. \quad (3.1)$$

The *degree* $d(v)$ of a node $v$ is given by the number of links connected to a vertex. This gives rise to a simple, but important statistical measure of graph topologies, by interpreting the node degree as a random variable: the *node degree distribution* of a graph is described by the relative frequency

$$P(k) = \frac{|\{v \in V : d(v) = k\}|}{N} \quad (3.2)$$

of its node degrees. The average node degree $\bar{k}$ in an undirected, unweighted graph is

$$\bar{k} := \frac{2M}{N}, \quad (3.3)$$

since each edge is counted twice for the degree of the nodes it connects.

**Distances and Paths**  The *shortest path* between two nodes $v, v' \in V$ is the shortest possible sequence

$$(v, e_1, v_1, e_2, v_2, \cdots, e_k, v'), \; v_i \in V, e_i \in E, \; k \in \mathbb{N},$$

such that no node appears twice. If a path can be found between any two vertices $v, v'$, the graph is *connected*. 
3.1 Fundamentals

The *distance* between two nodes is the length of their shortest path, i.e. the number of links between them. In a weighted graph, the distance is the sum of the weights of all links of a shortest path between two nodes.

### 3.1.2 Matrix formulation

Many properties of graphs can be analysed with more ease by modelling their topology with linear algebraic expressions. The *adjacency matrix* of a graph $G$ stores the information on connections within the graph:

$$ A_G = (a_{jk})_{j,k} \in \mathbb{R}^{N \times N}, $$

$$ a_{jk} := \begin{cases} 1, & v_j \text{ adjacent to } v_k \\ 0, & \text{otherwise.} \end{cases} $$

Another graph representation matrix carrying important information is the *Laplacian* matrix $L_G$:

$$ L_G := D - A_G, $$

$$ D = (d_{jk})_{j,k} $$

$$ d_{jk} := \begin{cases} d(v_j), & j = k, \\ 0, & \text{otherwise.} \end{cases} $$

Sorting the eigenvalues of the Laplacian matrix in ascending order, the first eigenvalue $\lambda_1$ is always 0. The number of eigenvalues which are 0 is the number of connected components of the graph, i.e. if $\lambda_2 \neq 0$ the graph is connected [47]. Therefore, $\lambda_2$ is also called the *algebraic connectivity* of the graph $G$.

### 3.1.3 Voronoi diagrams

A *Voronoi diagram* is a partition of a Euclidean space, in this chapter always $\mathbb{R}^2$, based on a set of spatially distributed points. Given the set of points $P \subset \mathbb{R}^2$, the Voronoi region $V_P \subset \mathbb{R}^2$ for a $P \in P$ is defined as the set of all points $Q \in \mathbb{R}^2$ for whom

$$ \|P - Q\| \leq \|P' - Q\| \quad \forall P' \in P \setminus \{P\}, $$

\[42\]
i.e. all $Q \in \mathcal{V}_P$ are closer to $P$ than to any other point in $\mathcal{P}$, except for those at the border of $\mathcal{V}_P$ which have equal distance to another point.

The Voronoi region $\mathcal{V}_P$ of a $P \in \mathcal{P}$ is computed by intersecting all bisector lines between $P$ and all other $P' \in \mathcal{P}$, i.e. the lines which divide the space into half planes $H(P, P')$ and $H(P', P)$, where $H(P, P')$ is the set of all points which are closer to $P$ than to $P'$:

$$\mathcal{V}_P = \bigcap_{P \neq P'} H(P, P').$$

Figure 3.1 shows a Voronoi diagram with 5 points, as well as the so-called Delaunay triangulation which is the dual graph to the Voronoi diagram, treating regions as nodes which share an edge if they are neighbours in the geographical sense; an interesting property of the Delaunay triangulation is that it contains all Euclidean distance-based minimum spanning trees between the points in $\mathcal{P}$.

Due to their process of construction, Voronoi regions are always convex. The theoretical foundation of Voronoi diagrams can be found e.g. in [48, 49]. In this chapter, only the intersections with $[0, 1] \times [0, 1]$ are considered when Voronoi
3.1 Fundamentals

3.1.4 The power grid graph

Following the definition given in [10], the power grid is modelled as a graph, where the nodes are the substations or any other points where lines connect, and the edges are the connecting lines between them. This definition is related to the traditional Graph-Theory-based method for circuit analysis in which points of the same potential form nodes and components form edges, e.g. [50]. A special case arises for transformers which are modeled as a set of two connected nodes belonging to different voltage levels.

The line impedances correspond to weights in the adjacency matrix, forming the commonly used admittance matrix.

3.1.5 Categorising different graph types

The node degree distribution characterises the topology of different types of graphs, for each of which generation algorithms have been derived:

- *Trees*, which do not contain any circles;
- *Random graphs*, in which links are distributed across the nodes with no preference [51];
- *Scale-free graphs*, in which preferential attachment of links forms the topology, leading to the existence of nodes of great importance called *hubs* [12];
- *Small-world graphs*, which have a relatively small average path length and a high clustering coefficient, meaning that the neighbourhoods of all nodes are in average highly connected [52, 53].

Trees are a special category of graphs and are often used to abstract the edges constituting shortest paths within a graph through a *minimal spanning tree*. Since their average degree poses a limit for that of a connected graph, it is shortly discussed here: it can be shown quite easily that a tree has \(2N - 2\) edges [46], and therefore, according to (3.3), the average degree of a tree is smaller than 2 at

\[
\bar{k}_{tree} = 2 - \frac{2}{N}.
\]
While random and small-world graphs have similar node degrees following symmetrical Poisson-like distributions, the node degrees of scale-free graphs follow a power-law distribution, which indicates the existence of several nodes with small degree and a small number of hubs with disproportionately high node degrees.

However, the degree distribution alone was found to be insufficient for the characterisation of graph topologies, and authors contradict each other on the distribution type of power grids especially on the question whether the distribution is rather an exponential law or a power law [54, 44]. For example, it is not possible to assess the probability of a graph having a meshed structure solely based on the degree distribution.

An important measure is therefore the joint degree distribution (JDD), a two-dimensional distribution which captures the relative frequency of connections between nodes with degree $k$ and nodes with degree $k'$. The JDD is defined as a matrix

$$J = (j_{kk'})_{kk'} \in \mathbb{R}^{d_{max} \times d_{max}},$$

$$d_{max} := \max_{v \in V} d(v),$$

$$j_{kk'} := \frac{m_{kk'} \mu_{kk'}}{2M},$$

where $d_{max}$ is the largest degree in the graph, $m_{kk'}$ is the number of edges which connect nodes of degree $k$ and $k'$, respectively, and

$$\mu_{kk'} := \begin{cases} 2, & k = k', \\ 1, & \text{otherwise}. \end{cases}$$

Since a graph consisting solely of circles, an Euler graph, only contains nodes with even degrees [46], large entries at even rows and columns imply a higher probability of ringed structures in the corresponding graph.

In [55], Mahadevan et al. present a systematic way of narrowing down topological characteristics of graphs through different levels of properties named $dK$, $d \in \mathbb{N}_0$. For example, $0K$ refers to the selection of an average degree, $1K$ to the choice of a degree distribution, and $2K$ to the specification of a JDD. The statement in [55] is that specification of higher $dK$ narrows down the topology of a graph until finally a unique graph can be defined. In [44], it is stated
that the JDD of graphs with similar degree distributions may vary significantly. Therefore, by specifying not only the degree distribution but also the JDD, a structural similarity of generated grid statistics can be enabled. At the same time, some variety of structurally similar grids is desired, i.e. narrowing down further properties works against the purpose of offering large sets of unequal grids.

3.2 Grid generation algorithm

Three main steps are performed for the generation of a distribution grid model:

1. the generation of an MV grid;
2. the generation of LV grids attributed to each MV node;
3. assigning electrical properties to the generated grid.

Since each of these steps constitutes an own algorithm, they are treated in separate sections detailing the inputs and the steps taken to obtain the respective output. The final grid graph \( G = (V,E) \) is then obtained from the MV grid \( G_{MV} = (V_{MV},E_{MV}) \) and the overall LV grid \( G_{LV} = (V_{LV},E_{LV}) \) through

\[
V := V_{MV} \cup V_{LV}, \\
E := E_{MV} \cup E_{LV}.
\]  

(3.4)

3.2.1 Medium voltage grid generation

An MV grid \( G_{MV} = (V_{MV},E_{MV}) \) is generated in four steps:

**Step 1** create an initial connected graph with the nodes \( v \in V_{MV} \);

**Step 2** assign a target node degree \( k_v \) to each node \( v \) to fulfill the desired \( \bar{k} \);

**Step 3** for each node, successively connect to other nearby nodes to attain at least the target degree;

**Step 4** in case needed, remove edges.
Chapter 3 Statistical generation of distribution grid models

These steps are detailed in the following paragraphs. The main inputs to the MV algorithm are

- the desired number $N = |V_{MV}| \in \mathbb{N}$ of MV nodes (or substations);
- the average node degree $\bar{k} \in [2 - \frac{2}{N}, \infty]$ limited by the average degree of a tree with $N$ nodes;
- the choice of whether or not lines may cross each other through the Boolean parameter $\text{planarity} \in \{0, 1\}$;
- a measure for encouragement of radial or ringed structure through the Boolean parameter $\text{parity} \in \{0, 1\}$ and the real parameter $RU \in [0, 1]$.

While the first three inputs are intuitive and well-known properties in Graph Theory, the parameters $\text{parity}$ and $RU$ (for rural-urban) are newly created to reflect the degree distribution and JDD of the resulting grid and will be explained in detail in Steps 2 and 3.

**Step 1: forming an initial graph** The principle of the MV grid generation is similar to [15, 45, 44], including the geographical element through a relative distance and the random spatial distribution of nodes through a point process [56, 57]. In this chapter, a uniform point process is used on the unit square, meaning that each position is chosen with equal probability.

All nodes $v \in V_{MV}$ are therefore assigned uniformly distributed random coordinates $(x_v, y_v)$ in the Cartesian coordinate system in $[0, 1] \times [0, 1]$. Nodes will henceforth be interpreted as vectors in the Euclidean vector space $\mathbb{R}^2$ by defining

$$v := (x_v, y_v), \quad x_v, y_v \in [0, 1].$$

Likewise, edges will be interpreted as line segments in the geometric sense, the end points being the nodes they connect:

$$e_{v,v'} := [vv'] \subset [0, 1] \times [0, 1],$$

which are always within the unit square since it is convex.

As in Barabasi’s algorithm for the generation of scale-free graphs, an initial graph is formed as a kernel for growing the rest of the grid [12]. There, a
complete graph is formed as a starting point to which nodes and connections are added. However, this approach does not serve the purpose of generating graphs with a certain size, and additionally complete subgraphs past the triangle graph $K_3$ can be considered unlikely in power grids, as they represent the case where the grid contains a region in which all substations are connected with each other.

In order to keep the desired size $N$ fixed, a minimum spanning tree $T_{min}$ of the spatially distributed nodes $v_i \in V_{MV}$ is chosen similarly to the grid generation algorithm in [44] as the initial kernel, adding the benefit of guaranteeing connectedness. For this purpose, Prim’s well-known algorithm for minimal spanning trees on weighted graphs is performed [58], assuming for the moment that all nodes are connected as the complete graph $K_N$, the edge $[v_i v_j]$ between any pair $\{v_i, v_j\}$ of nodes weighted with the Euclidean distance

$$|[v_i v_j]| = ||v_i - v_j|| \in \mathbb{R}.$$  

Since Prim’s algorithm is performed with the weighted adjacency matrix which is symmetrical, it requires

$$N^2 - \frac{1}{2}(N - 1)(N - 2) - \underbrace{N}_{\text{diagonal elements}} - \underbrace{N}_{\text{subdiagonal elements}}$$  

operations of determining distances between nodes before computing the minimum spanning tree. Now, all connections apart from those in the calculated tree are omitted, reducing the initial graph to $T_{min}$.

Since trees have the smallest average degrees of all graph types, when the input $\bar{k}$ is smaller than the average degree of $T_{min}$, adding more edges will create a larger gap between the output and the desired statistics, therefore in such a case the output grid $G_{MV}$ is the unchanged minimum spanning tree $T_{min}$.

**Step 2: assigning target node degrees** Throughout the process of generating MV grids, the topology is altered through statistical measures such as the node degree distribution or the JDD. Therefore, it is important to add edges to $T_{min}$ in a way that the desired statistics will be approximated. For this purpose, a discrete probability distribution is constructed from which target node degrees
can be sampled for each node $v \in V_{MV}$.

Let the function

$$p_{\text{assign}} : S \subset \mathbb{N} \rightarrow [0, 1),$$

$$i \mapsto \alpha_i, \ i \in S$$

be the probability mass function of this distribution, where the relative frequencies $\alpha_i$ of the degrees $i$ and the elements of $S$ are yet to be determined.

Since by the definition of expected values the desired average degree $\bar{k}$ is obtained from this distribution as

$$\bar{k} = \sum_{i \in S} i \alpha_i,$$

and by definition of a probability distribution it is known that

$$\sum_{i \in S} \alpha_i = 1,$$

there are some indications concerning the determination of the $\alpha_i$.

For the choice of the $\alpha_i$, there are two possibilities. In the simplest case, a degree distribution is available as a reference, e.g. through known statistics of a real or artificial grid type. Then a least squares fit of the desired degree distribution can be made for the $\alpha_i$ with the constraints of (3.8) and (3.9), and the problem can be considered solved.

In the more common case, no such details are known, and without a reference distribution, the main goal becomes the approximation of $\bar{k}$ by means of choosing the elements of $S$ sensibly and tuning the corresponding $\alpha_i$.

Some observations can be made on Equations (3.8) and (3.9):

- the two equations form an underdetermined system of equations for the $\alpha_i$ when $|S| > 2$, which can be considered the common case;

- the largest element of $S$ must be larger than $\bar{k}$, since $\bar{k}$ can otherwise not be determined through (3.8);

- the value of $\alpha_1$ is directly related to the likelihood of the grid to be radial or tree-like, since it corresponds to the number of nodes with degree 1.
The last point is especially important and leads to the definition of the rural-urban factor \((RU)\) which was mentioned as an input to the MV algorithm:

\[
RU := \alpha_1.
\]

When \(RU\) is small, there will be less nodes with degree 1, therefore representing a less radial grid typical for urban areas, and for large \(RU\) there will be several nodes which only connect to one other node, typical for rural areas \([59]\).

Since \(RU\) is given as an input parameter, two more \(\alpha_i\) can be determined uniquely through the two linear equations (3.8) and (3.9), for which the corresponding elements of \(S = \{1, x, y\}\) must be chosen. Following the observation that one element of \(S\), without loss of generality \(y\), needs to be larger than \(\bar{k}\) but not too large in order to stay close to a power or exponential law, an obvious choice is

\[
y = [\bar{k}] + 1.
\] (3.10)

The remaining element \(x\) is now chosen as \([\bar{k}]\), as the other integer in close proximity of \(\bar{k}\):

\[
S = \{1, [\bar{k}], [\bar{k}] + 1\}.
\] (3.11)

Now, \(\alpha_x := \alpha_{[\bar{k}]}\) and \(\alpha_y := \alpha_{[\bar{k}]+1}\) can be computed from the system of equations resulting from (3.8) and (3.9):

\[
\text{Equation I} \quad ([\bar{k}] - 1)\alpha_x + [\bar{k}]\alpha_y = \bar{k} - 1,
\]

\[
\text{Equation II} \quad \alpha_x + \alpha_y = 1 - RU.
\] (3.12)

This leads to the solution

\[
\alpha_x = \bar{k} - [\bar{k}] + RU \cdot ([\bar{k}] - 1),
\]

\[
\alpha_y = -\bar{k} + [\bar{k}] - [\bar{k}] \cdot RU + 1,
\]

concluding the definition of a probability mass function from which target degrees \(d_{\text{target}}(v)\) are generated for each \(v \in V_{MV}\).

It is clear that there is an upper limit smaller than 1 to the possible size of \(RU\) as only connected graphs are considered, and further node degrees will therefore naturally appear. The feasible \(RU\) values are discussed in the following.
Interpreting Equation I as a linear function of $\alpha_y$ as in
\[
\alpha_x(\alpha_y) = -\frac{\lceil \bar{k} \rceil}{\lceil \bar{k} \rceil - 1} \alpha_y + \frac{\bar{k} - 1}{\lceil \bar{k} \rceil - 1},
\]
(3.13)
it can be seen that the slope is smaller than $-1$ and that the zero lies at
\[
\alpha_y = \frac{\bar{k} - 1}{\lceil \bar{k} \rceil}.
\]
(3.14)
Likewise, Equation II in (3.12) describes a family of parallel lines with slope $-1$ which can be shifted through the parameter $RU$, interpreting $\alpha_x$ as a function of $\alpha_y$:
\[
\alpha_x(\alpha_y) = -\alpha_y + (1 - RU).
\]
(3.15)
Since the system of equations in (3.12) can only be solved if line (3.13) and line (3.15) intersect, finding the region in which such intersections can exist between 0 and 1 determines the boundaries for $RU$. Figure 3.2 shows the two limiting parallel lines, containing the grey region where solutions are possible.

![Figure 3.2: Boundaries for the input parameter $RU$](image)

Since the slope in (3.13) approximates 1 for larger average degrees, it is ob-
3.2 Grid generation algorithm

It is obvious that the region of feasible nonzero $RU$ values shrinks for large $\bar{k}$, as the line becomes closer and closer to being parallel to the lines defined in (3.15). However, according to [9, 10], the average degree in power grids usually lies between 2 and 3. Therefore, it is likely that there is some scope of variability for $RU$ as an input unless highly meshed grids with high average degrees are desired, where end nodes, i.e. nodes of degree 1, can hardly occur and there is no meaning in varying their probability.

**Step 3: connecting nodes** Having generated target node degrees, as long as

$$d(v) < d_{target}(v)$$

holds for some nodes $v \in V_{MV}$, connections are added to $T_{min}$ according to three criteria: Euclidean distance, parity and planarity.

The Boolean input *parity* refers to the enforcement of connections between nodes whose degrees are equal modulo 2, i.e. nodes with odd degrees connect to other nodes with odd degrees and nodes with even degrees connect to other nodes with even degrees. When $parity = 1$, this enforcement is activated. This is the input which drives the manipulation of the JDD, which as discussed in Section 3.1 was found to be crucial for a topological specification, e.g. of the likelihood of meshedness. In [44], Cloteaux shows that in the Polish grid the largest entry of the JDD corresponds to connections of matching parity, as opposed to the more radial American grids which feature higher JDD values corresponding to connections of opposite parity. This tendency of European grids could also be found in the 12 Dutch grids analysed by Pagani in [10], as shown in [60].

The Boolean input *planarity* is related to the possibility of lines crossing each other, derived from the concept in Graph Theory that a graph is planar if it can be drawn on a 2D plane without any edges intersecting. When the input is $planarity = 1$, the algorithm therefore needs to keep track of intersections when adding new connections. While planarity seems like a generally desirable property, this quantity is left as a parameter in order to enable topologies in which lines cross, which may occur e.g. due to underground cables.

The following process of adding connections is conducted one node at a time for each $v \in V_{MV}$ until $d(v) \geq d_{target}(v)$, assuming that *parity* and *planarity*
are true:

1. select a node \( v' \) from \( V' := V_{MV} \setminus (V(\Gamma_v) \cup \{v\}) \) where

\[
||v' - v|| = \min_{v^* \in V'} ||v^* - v||;
\]

2. if \( d(v) \equiv d(v') \mod 2 \), proceed to the next step, otherwise go back to the first step while deleting \( v' \) from \( V' \);

3. computing intersections of the edge \([vv']\) with all other existing edges in \( E_{MV} \setminus \{[vv']\} \), if

\[
[vv'] \cap [v_iv_j] = \emptyset \quad \forall [v_iv_j] \in E_{MV}
\]

proceed to the next step, otherwise go back to the first step while deleting \( v' \) from \( V' \);

4. add the connection \([vv'] = e_{v,v'}\) to \( E_{MV} \).

Note that when parity or planarity are false, Steps 2 or 3 can be skipped, respectively.

For nodes with a degree higher than 5, planarity is not enforced and Step 3 is omitted, motivated by the fact that all complete graphs larger than \( K_5 \) are nonplanar.

**Step 4: final adjustments** Since the graph generated in the last paragraphs only uses the input average degree \( \bar{k} \) as a reference but does not evaluate the average degree after each new connection, without adjustments a discrepancy between the desired \( \bar{k} \) and the actual average \( \bar{k}' \) is likely. Additionally, for each node \( v \in V_{MV} \) the test \( d(v) < d_{target}(v) \) is considered as a criterion for adding edges, however the opposite case \( d(v) > d_{target}(v) \) is currently not yet incorporated for removing edges.

As mentioned above, it is quite common for power grids to have small average degrees between 2 and 3, therefore no changes are implemented in the unlikely case where \( \bar{k} > \bar{k}' \) already holds. When \( \bar{k}' > \bar{k} \), i.e. there are still too many edges in the generated grid, edges are removed.
3.2 Grid generation algorithm

It is clear that edges incident to nodes with degree 1 cannot be deleted as the grid needs to stay connected. Therefore, only edges in

\[ E' := E_{MV} \setminus \{[v_i,v_j] \in E_{MV} : d(v_i) + d(v_j) \leq 3\} \]

are considered for removal in the following steps which are conducted until \( \bar{k} \approx \bar{k}' \):

1. select an \([v_i,v_j] \in E'\) such that either

\[ d(v_i) > 2 \quad \land \quad d(v_j) > 2, \]

or if this is not possible

\[ d(v_i) > 2 \quad \lor \quad d(v_j) > 2, \]

or if this is also not possible

\[ d(v_i) \neq 1 \quad \land \quad d(v_j) \neq 1; \]

2. if the algebraic connectivity \( \lambda_2 \) defined in Section 3.1 is nonzero for the graph \( G = (V_{MV}, E_{MV} \setminus \{[v_i,v_j]\}) \), proceed with the next step and otherwise go back to step 1 while deleting \([v_i,v_j]\) from \( E'\);

3. remove \([v_i,v_j]\) from \( E_{MV}\).

As in the edge addition, these steps are executed under the assumption that parity is true. If it is chosen to be false, the conditions for selecting the edge for removal in Step 1 do not need emphasise the preservation of nodes with degree 2, and the constraints can be relaxed to the last:

\[ d(v_i) \neq 1 \quad \land \quad d(v_j) \neq 1. \]

Step 2 checks whether the graph is still connected.

Finally, the remaining graph \( G_{MV} = (V_{MV}, E_{MV}) \) with the updated set of edges \( E_{MV} \) is the output of the algorithm for MV grid generation.
3.2.2 Low voltage grid

The generation of an LV grid \( G_{LV}^{(v)} = (V_{LV}^{(v)}, E_{LV}^{(v)}) \) is based on the creation of feeders attached to an MV node \( v \in V_{MV} \), treating it as a substation; this procedure is successively conducted for each MV node. Similarly to the MV grid generation, four steps are involved:

**Step 1** define geographical regions for an LV grid around each MV node;

**Step 2** for each MV node, determine the number \( F \in \mathbb{N} \) of feeders \( f_j \) and section the region into \( F \) subregions;

**Step 3** in each subregion, grow a feeder by adding nodes successively for the feeders that should not branch;

**Step 4** determine the branching pattern for feeders that should branch.

According to a study conducted by Kerber [61], there are three types of LV feeder topologies: simple feeders, branching feeders and meshed systems. However, meshed systems are considered very seldom and are therefore neglected in this section.

The main inputs to the algorithm for LV grid generation are

- the average number \( \mu_N \) of nodes per LV system;
- the standard deviation \( \sigma_N \) of this number;
- the average number \( \mu_F \) of nodes per single LV feeder;
- the standard deviation \( \sigma_F \) of this number.

**Step 1: defining geographical LV regions** Since it is assumed that LV nodes tend to be connected to the closest MV substation, the Voronoi diagram detailed in 3.1.3 is a suitable candidate for the definition of such regions. Therefore the first step is the partitioning of the unit square into Voronoi regions \( \mathcal{V}_v \) surrounding the MV nodes \( v \in V_{MV} \) as areas for the construction of LV systems, restricted to \( [0, 1] \times [0, 1] \) as discussed in 3.1.3. In the following paragraphs, \( \partial \mathcal{V}_v \) will refer to the boundary of \( \mathcal{V}_v \).
3.2 Grid generation algorithm

**Step 2: defining the number of feeders and subregions** Consider the Voronoi region $V_v, v \in V_{MV}$. The philosophy of placing single LV feeders in $V_v$ is to determine the total number $F \in \mathbb{N}$ of feeders connected to the corresponding MV node, and then to divide the Voronoi region based on angles into $F$ subregions which contain one feeder each.

In order to determine $F$, first the total number $|V_{LV}^{(v)}|$ of LV nodes in $V_v$ is computed by sampling a random number $r_N$ from the normal distribution $\mathcal{N}(\mu_N, \sigma_N)$ and then rounding to obtain an integer:

$$|V_{LV}^{(v)}| := \lfloor r_N + 0.5 \rfloor.$$  

Next, $|V_{LV}^{(v)}|$ is partitioned into single sizes $|f_j|$ of feeders $f_j$ by successively rounding a random number $r_{f_j}$ sampled from the normal distribution $\mathcal{N}(\mu_F, \sigma_F)$:

$$|f_j| := \lfloor r_{f_j} + 0.5 \rfloor.$$  

This step is repeated until

$$\sum_{j} |f_j| \geq |V_{LV}^{(v)}|.$$  

(3.16)

The number of times a feeder size was generated to obtain this relation is the total number $F$ of feeders. The last summand $|f_F|$ of (3.16) is now curtailed to achieve equality:

$$\sum_{j=1}^{F} |f_j| = |V_{LV}^{(v)}|.$$  

Now that $F$ is known, the task is to divide $V_v$ into $F$ subregions. The subregions are defined by drawing a circle around $v$ and dividing the $360^\circ$ angle of this circle into $F$ equal angles $\theta_F$ which define the regions. The respective angle sections are assigned to the $f_j$ counterclockwise.

However, depending on the shape of $V_v$, this simple division may result in very different sizes for the subregions. In order to discourage strongly diverging feeder lengths within the same LV network, some parts of $V_v$ are not considered for the forming of subregions depending on a threshold which is defined in the following.

56
(a) Case 1: circle lies completely within $\mathcal{V}_v$, shown with 5 subregions with $\theta_F = 72^\circ$

(b) Case 2: circle intersects $\partial \mathcal{V}_v$ twice

(c) Case 3: circle intersects $\partial \mathcal{V}_v$ 4 times

Figure 3.3: Constructing subregions for different shapes of $\mathcal{V}_v$
3.2 Grid generation algorithm

In a first step, a point \( t \in \mathcal{V}_v \) is found such that:

\[
||t - v|| = \max_{t' \in \mathcal{V}_v} ||t' - v||.
\]

Since all Voronoi regions are convex, \( t \) always is one of the corner points on the boundary \( \partial \mathcal{V}_v \). Now, a threshold of 25\% of \( ||t - v|| \) is defined:

\[
l_{\text{thresh}} := \frac{||t - v||}{4},
\]

which is used as the radius for the circle drawn around \( v \). Only the total angle \( \theta_{\text{total}} \leq 360^\circ \) of the arcs of the circle lying within \( \mathcal{V}_v \) is considered for forming subregions.

There are only three possibilities for this circle to intersect with the boundary \( \partial \mathcal{V}_v \), also shown in Figure 3.3:

- there is no intersection and the total angle \( \theta_{\text{total}} \) of the arcs of the circle lying within \( \mathcal{V}_v \) is \( 360^\circ \);
- there are 2 intersection points and the circle is divided into two parts by the boundary \( \partial \mathcal{V}_v \), so the total angle \( \theta_{\text{total}} \) of the arcs of the circle lying within \( \mathcal{V}_v \) is smaller than \( 360^\circ \);
- there are 4 intersection points and consequently two separate arcs of the circle lying within \( \mathcal{V}_v \) with angles \( \theta \) and \( \theta^* \).

In the first two cases, the total angle \( \theta_{\text{total}} \leq 360^\circ \) is considered for forming subregions with the angle span

\[
\theta_F = \frac{\theta_{\text{total}}}{F}.
\]

In the last case, \( \mathcal{V}_v \) is split into two regions, one containing the angle \( \theta \) and the other containing the angle \( \theta^* \). As it is unlikely that \( \theta \) and \( \theta^* \) are equal, the feeders are distributed to these two regions relatively to their angles:

\[
F' = \left\lfloor \frac{\theta}{F} \right\rfloor,
\]

\[
F^* = F - F',
\]

where \( F' \) is the number of feeders attributed to the region containing \( \theta \) and \( F^* \)
is the number of feeders attributed to the region containing $\theta^*$. Finally, both subregions are treated as separate regions with their respective total angle and new number of feeders, thus reducing the problem to the second case in the considerations on intersections above. This concludes the definition of subregions $R_j$ for each feeder $f_j \subset V_{LV}^{(v)}$.

Besides setting the sizes $|f_j|$ of the single feeders, it is also useful to determine how often each feeder should branch. Kerber shows that multiple branching is uncommon and the main feeder usually does not branch more than 3 times [61]. Therefore, only the numbers $\{0,1,2,3\}$ are considered for the branching of the main feeder, and each feeder $f_j$ is assigned a number $b(f_j)$ randomly selected from these, representing the number of times $f_j$ should branch:

$$b(f_j) = \begin{cases} u \sim \mathcal{U}\{0,1,2,3\}, & |f_j| \geq 10 \\ 0, & \text{otherwise} \end{cases} \quad (3.17)$$

Here, it is assumed that branching will only occur for feeders with more than 10 nodes, as explained later in Step 4.

**Step 3: creating feeders**

Having defined the geographical boundaries for each LV feeder connected to $v \in V_{MV}$, the actual feeders $f_j$ can be constructed by determining the positions of their nodes. The logic of constructing feeders is explained for one of the subregions created in Step 2, and is then subsequently conducted for all other subregions. This step concerns all feeders which do not branch, whereas all others are treated in the last step.

Let $f_j$ denote one of the $F$ feeders connected to $v$ for which $b(f_j) = 0$ holds, and let $R_j \subseteq V_v$ be the subregion created for $f_j$ in the last paragraph. Within $R_j$, consider $t_j$ to be the furthest corner point from $v$ on the boundary $\partial R_j$, similarly to $t$ in the last paragraph. The philosophy for feeder growth is to construct the position of the $i$-th node $v_i \in f_j$ in the direction of $t_j$, based on the position of the $(i-1)$-th node, $i = 1, \ldots, F$. In this numbering, $v_0$ is considered to be the MV node $v$. Given the position $(x_{v_{i-1}}, y_{v_{i-1}})$ of $v_{i-1}$, the task is therefore to determine $(x_{v_i}, y_{v_i})$.

First, the quantity

$$\rho_i := (|f_j| - i) \cdot \|t_j - v\|/|f_j| + u \cdot \|t_j - v\|/|f_j| \quad (3.18)$$
is defined, where the expression
\[ \frac{||t_j - v||}{|f_j|} \]
equally divides the longest distance from \( v \) into \( |f_j| \) parts. In (3.18), the first summand therefore represents all segments up to the \((i-1)\)-th segment counting from \( v \) towards \( t \), and the second summand adds a random fraction of the \( i \)-th segment through the \( U(0,1) \)-distributed random number \( u \). Essentially, \( \rho_i \) randomly selects a distance from \( t \) which lies somewhere within the \( i \)-th segment, counting from \( v \). Now, a radius
\[ r_i := ||t_j - v|| - \rho_i \]
can be defined. The next node \( v_i = (x_{v_i}, y_{v_i}) \) is randomly selected on the circle
\[ (x - x_{v_{i-1}})^2 + (y - y_{v_{i-1}})^2 = r_i^2 \] (3.19)
centered around \( v_{i-1} \) and added to \( V_{LV}^{(v)} \), with the restriction that it may only lie on the half of the circle closer to \( t_j \), and within \( R_j \). This half can be easily found by constructing the line through \( v_{i-1} \) which is perpendicular to \([v_{i-1}t_j]\) and splits the circle in the desired way. To conclude the feeder growth to \( v_i \), the connection \( e_{v_{i-1},v_i} = [v_{i-1}v_i] \) is added to \( E_{LV}^{(v)} \). Figure 3.4 shows the geometrical construction of the first feeder node \( v_1 \).

Repeating this procedure \( |f_j| \) times yields the graph for feeder \( f_j \), and executing the same logic for each feeder with \( b(f_j) = 0 \) concludes this step.

**Step 4: branching the feeders** One last topological phenomenon which may occur is the branching of feeders, which is conducted for all \( f_j \) with \( b(f_j) > 0 \). In [62], relative lengths of feeder branches are determined in relation to the total Euclidean length of the feeder, under the assumption that all nodes of a feeder are equidistant and that branching only occurs at nodes.

Though the distance between the nodes of a feeder as constructed in Step 3 may vary due to the random factor, the formulation in [62] can be modified to use node numbers instead of Euclidean length since these two are proportional in the equidistant setting. The branches are denoted as \( B_k, 0 \leq k \leq b(f_j) \), where \( B_0 \) is called the main feeder. The branching scheme is shown in Figure 3.5, where
Figure 3.4: Determining the position of the feeder node $v_1$ starting from the MV node $v$

each ratio corresponds to the number of nodes assigned to the single branch or branch section, showing why feeders with less than 10 nodes are not considered for branching in (3.17). When the ratios are not integers, they are rounded to the closest integer.

The procedure of branching is related to the feeder growth principle in Step 3, with the difference that multiple nodes are added in each iteration, extending the branches simultaneously until they have reached their respective node ratio. For each branch $B_k$, the position of the $i$-th node $v_{i,B_k}$ is constructed from $v_{i-1,B_k}$. Again, the process is explained for one feeder $f_j$ associated to the MV node $v$, and then applied to all other feeders which should branch. The terminology is the same as in Step 3.

Until the first branching which happens after the first ratio of nodes for the main feeder shown in Figure 3.5, the procedure is the same as in Step 3. Starting from the node $v_{i-1,B_0}$ where branching occurs first, nodes are added successively as detailed in the following.

In the $i$-th iteration, a circle of radius $r_i$ as in (3.19) around the centre $v_{i-1,B_0}$ is constructed as usual. Next, the intersection between the subregion $R_j$ and the circle is determined, yielding the arc of the circle closer to $t_j$ on which further
3.2 Grid generation algorithm

Figure 3.5: Branching scheme with ratios of nodes per branch [62]
nodes may be selected. This arc is divided into $b$ equal parts with angles $\theta$, if $b \leq b(f_j)$ is the number of branches which are still growing to achieve their ratio of nodes. First, the position of the node $v_{i,B_0}$ of the main branch is randomly selected from the arc closest to the boundary of $R_j$, in order not to cross other branches. The same step is then conducted for all active branches, selecting random positions for the branch nodes from the respective arcs, which are assigned to the branches according to the order shown in Figure 3.5. An overview of the branching procedure can be found in Figure 3.6. Nodes are added to all active branches in this way until all branches reach their assigned ratio of nodes.

![Figure 3.6: Simultaneous growth of two branches of a feeder](image)

Due to the rounding when calculating the ratios of nodes for each branch, it is possible that there are slightly more or less nodes in the feeders which branch than determined initially. However, since the ratios are always based on the original feeder length, and since there are several LV networks, it is not possible for the total number to stray far from the target, and it is likely that the effect is balanced between different feeders and LV networks.
Combining all LV networks In a final step, the overall LV network is obtained by assembling all \( V_{LV}^{(v_i)} \), \( E_{LV}^{(v_i)} \) generated for MV nodes \( v_i \in V_{MV} \):

\[
G_{LV} = (V_{LV}, E_{LV}) := \left( \bigcup_{i=1}^{N} V_{LV}^{(v_i)} , \bigcup_{i=1}^{N} E_{LV}^{(v_i)} \right), \tag{3.20}
\]

remembering that \( N \) is the number of MV nodes. It is obvious that the LV grid on its own is unconnected with \( N \) components.

### 3.2.3 Electrical properties

Having constructed the topological structure of the MV and LV grids, what remains is to assign impedances to the edges of the overall graph \( G = (V, E) \) defined by (3.4). As mentioned in Section 3.1, substation transformers need to be considered separately when constructing the admittance matrix.

**Assigning electrical weights to nodes and edges** Each MV node is equipped with an additional node which lies at the same location, and an edge connecting to the new node. The original MV node now connects to the MV grid, whereas the duplicated node connects to the LV grid instead of the original MV node. The edge with zero length between the MV node and the duplicated node is interpreted as the transformer. The set of duplicate nodes which are defined to belong to the LV level is denoted as \( V_{tr} \), the set of additional edges as \( E_{tr} \).

Given a distance reference \( d \) in kilometers for scaling the unit square, a per kilometer impedance value \( z_k \in \mathbb{C} \) for the desired type \( k \) of cable or line, as well as the transformer impedance \( z_{tr} \in \mathbb{C} \), a weighting function

\[
W_{Y, edge} : E \cup E_{tr} \rightarrow \mathbb{C}, \tag{3.21}
\]

\[
e_{v,v'} \mapsto \begin{cases} 
-1 & \text{if } e_{v,v'} \text{ is an edge of type } k, \\
\frac{-1}{||v - v'|| \cdot d \cdot z_k} & \text{if } e_{v,v'} \text{ is a transformer.} 
\end{cases} \tag{3.22}
\]

is easily constructed, characterising the electrical properties of the links. Here, the parameter \( k \) is kept as a possibility to assign different impedance values for different edges, e.g. for cables or lines depending on the distance and voltage
level.

Additionally, a node weighting function is defined as in Section 3.1, by sum-
ming for each node the weights of the incident edges. Here, the neighbourhood
\( \Gamma_v \) of \( v \) is extended to include adjacent nodes from the newly constructed \( V_{tr} \).

\[
W_{Y_{node}} : V \cup V_{tr} \rightarrow \mathbb{C},
\]

\[
v \mapsto \sum_{v' \in V(\Gamma_v)} W_{Y_{edge}}(e_{v,v'}). \quad (3.23)
\]

**Constructing the admittance matrix** Since simple modifications of the adja-
cency matrix with the two weighting functions yield the admittance matrix, the
main task is to set up the adjacency matrix of the grid graph in a meaningful
way [63].

Though any ordering of the nodes would be eligible for uniquely determining
the grid topology in an adjacency matrix, it is useful to order the nodes in a way
that the rows and columns of the adjacency matrix reflect the grid structure. In
particular, the order proposed here leads to an adjacency matrix in which the
LV regions can be located at first glance as blocks along the diagonal.

Let \( v_{MV} \) be an MV node with \( F \) LV feeders which start at the duplicate
transformer node \( v_{LV} \), where \( v_{f_j} \in f_j \) are nodes of the \( j \)-th feeder. The edge
\( e_{v_{MV},v_{LV}} \) then stands for the transformer and the LV grid nodes associated to
\( v_{MV} \); including \( v_{MV} \) for the sake of completeness, can be ordered as follows:

\[
v_{MV},
v_{LV},
v_{f_1}^{f_1}, \ldots, v_{f_1}^{f_1},
\vdots
v_{f_F}^{f_F}, \ldots, v_{f_F}^{f_F}.
\]

This ordering is conducted for each MV node with its associated LV network,
followed by the creation of a global ordering of nodes obtained by appending all
these lists. Sorting the rows and columns of the adjacency matrix accordingly,
this ordering yields blocks on the diagonal representing the LV grids. Moreover,
the transformer and feeders can be distinguished in the structure, since the \( 2 \times 2 \)
3.3 Validation

matrix at the upper left of each block characterises the transformer, and feeders can be found as smaller block diagonal matrices within the corresponding LV block.

Now, the adjacency matrix can be completed by adding entry 1 to all positions associated with links between the MV nodes.

Finally, the admittance matrix is obtained by performing two steps:

1. inserting the function values $W_{Y_{edge}}(v_i,v_j)$ to the corresponding positions $(i,j)$ of the adjacency matrix when $i \neq j$;

2. inserting the function values $W_{Y_{node}}(v_i,v_i)$ to the corresponding position $(i,i)$ on the diagonal of the adjacency matrix.

3.3 Validation

This section presents topologies generated by the algorithm developed in 3.2, comparing their MV grid statistics with those of real grids. LV grids are not part of this comparison, since the generation procedure follows schemes proposed by [61] and [62] with the adjustment of removing the equidistance of nodes, and does not provide much room for variation apart from the number of nodes and feeder sizes which are defined as a user input.

For the purpose of validating the generated MV grids, the average degree and number of nodes of real grids are used as inputs to the algorithm to generate artificial grids whose degree distributions and line lengths are compared to those of the real data sets.

Real data were available from the 12 Dutch MV distribution grids analysed by Pagani et al. in [10], as well as data from the ATLANTIDE project, featuring 7 real and synthetic Italian MV grids [64]. Tables 3.1 and 3.2 show the respective data sets for the Dutch and Italian grids. In the Italian data, a difference is made between real data, named “full”, and clustered artificial data. The data shows that the real Dutch grids tend to be less radial than the Italian grids, as can be seen in their higher average degrees; the Italian grids’ average degrees are close to the average degree of trees at less than 2 as shown in Section 3.1. In the Italian case, the algorithm terminates shortly after creating the minimum spanning tree in order to stay close to the low average degree input and therefore does not conduct any further structural changes.
Chapter 3 Statistical generation of distribution grid models

For this reason, all analyses focus on the comparison of the generated grids’ statistics with those of the Dutch grids.

<table>
<thead>
<tr>
<th>Grid</th>
<th>N</th>
<th>M</th>
<th>average degree</th>
<th>RU</th>
</tr>
</thead>
<tbody>
<tr>
<td>NL 1</td>
<td>457</td>
<td>510</td>
<td>2.2319</td>
<td>0.1794</td>
</tr>
<tr>
<td>NL 2</td>
<td>427</td>
<td>498</td>
<td>2.3326</td>
<td>0.1288</td>
</tr>
<tr>
<td>NL 3</td>
<td>226</td>
<td>255</td>
<td>2.2566</td>
<td>0.1770</td>
</tr>
<tr>
<td>NL 4</td>
<td>269</td>
<td>313</td>
<td>2.3271</td>
<td>0.1747</td>
</tr>
<tr>
<td>NL 5</td>
<td>218</td>
<td>234</td>
<td>2.1468</td>
<td>0.2110</td>
</tr>
<tr>
<td>NL 6</td>
<td>191</td>
<td>210</td>
<td>2.1990</td>
<td>0.1780</td>
</tr>
<tr>
<td>NL 7</td>
<td>928</td>
<td>1120</td>
<td>2.4138</td>
<td>0.1282</td>
</tr>
<tr>
<td>NL 8</td>
<td>352</td>
<td>392</td>
<td>2.2273</td>
<td>0.1648</td>
</tr>
<tr>
<td>NL 9</td>
<td>214</td>
<td>241</td>
<td>2.2523</td>
<td>0.1402</td>
</tr>
<tr>
<td>NL 10</td>
<td>199</td>
<td>209</td>
<td>2.1005</td>
<td>0.1457</td>
</tr>
<tr>
<td>NL 11</td>
<td>261</td>
<td>282</td>
<td>2.1609</td>
<td>0.1839</td>
</tr>
<tr>
<td>NL 12</td>
<td>471</td>
<td>518</td>
<td>2.1996</td>
<td>0.1550</td>
</tr>
</tbody>
</table>

Table 3.1: Data on Dutch grids from [10]

<table>
<thead>
<tr>
<th>Grid</th>
<th>N</th>
<th>M</th>
<th>average degree</th>
<th>RU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Industrial</td>
<td>99</td>
<td>102</td>
<td>2.0606</td>
<td>0.1818</td>
</tr>
<tr>
<td>Industrial full</td>
<td>335</td>
<td>338</td>
<td>2.0179</td>
<td>0.3343</td>
</tr>
<tr>
<td>Rural</td>
<td>102</td>
<td>101</td>
<td>1.9804</td>
<td>0.3235</td>
</tr>
<tr>
<td>Rural full</td>
<td>280</td>
<td>282</td>
<td>2.0143</td>
<td>0.3929</td>
</tr>
<tr>
<td>Suburban</td>
<td>132</td>
<td>131</td>
<td>1.9848</td>
<td>0.2197</td>
</tr>
<tr>
<td>Suburban full</td>
<td>345</td>
<td>351</td>
<td>2.0348</td>
<td>0.3681</td>
</tr>
<tr>
<td>Urban</td>
<td>96</td>
<td>96</td>
<td>2</td>
<td>0.1875</td>
</tr>
</tbody>
</table>

Table 3.2: Data on ATLANTIDE Italian grids from [64]

3.3.1 Degree distribution and JDD

Since the algorithm produces random grid topologies, it is obvious that two grids generated with exactly the same inputs will still be different. A meaningful evaluation therefore must be based on a large number of generated grids with the same inputs.

For each of the 12 Dutch grids, 1000 artificial grids are produced with the algorithm developed in Section 3.2.1, using the number of nodes and average
degree given in Table 3.1. The degree distributions were available and could be used as a reference for generating target node degrees in Step 2 of the MV generation algorithm in Section 3.2.1. The overall average JDD of the 1000 sample grids is then compared with the JDD of the original grid. Due to the shape of the real grids’ JDDs which feature a prominent 2,2-component, the Boolean parameter parity was set to 1. Figure 3.7 shows the comparison of the original JDD with generated grids exemplarily for the NL1 grid. It can be seen in Figure 3.7c that a single grid generated with the NL1 grid statistics as input achieves a similar JDD in terms of topological features such as the strong second diagonal component, a tendency which could be shown in Figure 3.7b to be true for a large ensemble of 1000 generated grids as well. The fact that the highest node degrees differ by 2 in average must be considered in relation to the probability of existence of such nodes, which can be seen to be close to zero, i.e. it is likely that the difference lies in the existence of only one node each with such behaviour.

In Figure 3.7d, it can be seen that the average JDD of 1000 generated grids hardly differs from that of the real data set with the largest difference in the range of 10% only found in one component, thus proving the success of the desired tuning of the JDD through Step 3 of the MV generation algorithm. Similar results are found for all other Dutch grids as shown in Table 3.3, where $\Delta_{\text{max}}^{\text{ens}}$ is the maximum difference between components of the average ensemble JDD and the corresponding real grid’s JDD, $\mu_{\Delta_{\text{max}}}$ is the mean difference between all components of the average ensemble JDD and the corresponding real grid’s JDD, and $\sigma_{\Delta_{\text{max}}}$ the standard deviation thereof. The maximum differences are small, lying in the range of 4–13% with one outlier at 16% in the NL5 grid, showing the success in influencing the JDD.

A good fit of degree distributions could be achieved as well. In general, the degree distributions of the Dutch grids show exponential behaviour for smaller degrees, and close to constant behaviour past a threshold node degree. Figure 3.8 presents the degree distribution of generated grids compared to the original NL1 grid with the reference distribution fits of the original degree distribution as dashed lines; it is to be noted that the degree distribution of the generated grids incorporates the data from all 1000 generated grids. Therefore, the difference occurring for $k > 6$ is not to be interpreted as a real topological difference, but rather can be explained by the fact that the minimal possible
Figure 3.7: Comparison of the JDDs of the NL1 grid and generated grids
3.3 Validation

probability $\frac{1}{N}$ is approximately 0.0021 for the single NL1 grid with $N = 457$ nodes, and one thousandth of this number for the data set of 1000 generated grids, which has $N = 457000$ nodes. What is interesting to observe is the fact that the position of the data point corresponding to the highest node degree sits right above the $\frac{1}{N}$ line in both data sets. The actual topological behaviour consequently can be seen as similar, since the degree distribution indicates both for the single Dutch grid as well as the ensemble generated grid data that there are only very few nodes with the highest degree of 9 and 11, respectively.

Similar results are demonstrated for all 11 other comparisons between ensembles of 1000 generated grids and their real counterparts. These comparisons are shown in Appendix A. Exemplary topologies, generated without a given degree distribution, can be seen in Section 3.4, in Figures 3.12 and 3.13.

![Figure 3.8: Comparison of the degree distribution of the NL1 grid and 1000 generated grids](image)

Additionally, a comparison of the topological statistics is performed for a single generated grid and the corresponding real grid data in order to give a feeling for the behaviour of a random sample, to be seen in Figure 3.9. The NL1 grid is used again for consistency. Since the output degree distribution is calculated from the actual connections added to the minimal spanning tree
Table 3.3: Difference between the JDDs of Dutch MV grids and 1000 generated grids with corresponding statistics

<table>
<thead>
<tr>
<th></th>
<th>$\Delta_{max}^{ens}$</th>
<th>$\mu_{max}$</th>
<th>$\sigma_{max}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>NL 1</td>
<td>0.1262</td>
<td>0.1263</td>
<td>0.0299</td>
</tr>
<tr>
<td>NL 2</td>
<td>0.0599</td>
<td>0.0521</td>
<td>0.0377</td>
</tr>
<tr>
<td>NL 3</td>
<td>0.1176</td>
<td>0.1191</td>
<td>0.0202</td>
</tr>
<tr>
<td>NL 4</td>
<td>0.0739</td>
<td>0.0796</td>
<td>0.0160</td>
</tr>
<tr>
<td>NL 5</td>
<td>0.1626</td>
<td>0.1633</td>
<td>0.0428</td>
</tr>
<tr>
<td>NL 6</td>
<td>0.0733</td>
<td>0.0967</td>
<td>0.0243</td>
</tr>
<tr>
<td>NL 7</td>
<td>0.0682</td>
<td>0.0734</td>
<td>0.0109</td>
</tr>
<tr>
<td>NL 8</td>
<td>0.1180</td>
<td>0.1180</td>
<td>0.0172</td>
</tr>
<tr>
<td>NL 9</td>
<td>0.0747</td>
<td>0.0753</td>
<td>0.0280</td>
</tr>
<tr>
<td>NL 10</td>
<td>0.0564</td>
<td>0.0666</td>
<td>0.0267</td>
</tr>
<tr>
<td>NL 11</td>
<td>0.0862</td>
<td>0.0920</td>
<td>0.0172</td>
</tr>
<tr>
<td>NL 12</td>
<td>0.0459</td>
<td>0.0610</td>
<td>0.0174</td>
</tr>
</tbody>
</table>

according to the logic of the algorithm, its great similarity to the original degree distribution reflects the performance of the algorithm.

3.3.2 Line lengths

Though line lengths are not manipulated other than through the logic of connecting nodes which are geographically close and therefore are not associated with the success or failure of the algorithm, they are important parameters in considering the behaviour of power grids. Therefore, they are also compared in a similar way as for the degree distributions in Section 3.3.1, using the 1000 generated grids corresponding to each of the Dutch grids. The only input related to the line length is the distance reference input for electrical properties, as shown in Section 3.2.3.

Figure 3.10 shows how the average line length calculated from 1000 generated grids with the NL1 grid’s input exhibits near exponential behaviour past the maximum at approximately 2.5. Figure 3.10a shows a similar phenomenon as in the comparison of degree distributions, where the original data behaves exponentially until a certain threshold line length after which it is close to being constant. The occurrence of lower values past a line length of 10 km in the generated grid data can be explained with the same logic as above, i.e. the
### 3.3 Validation

![Comparison of the degree distribution of the NL1 grid and one generated grid](image)

Figure 3.9: Comparison of the degree distribution of the NL1 grid and one generated grid

... the minimal possible probability $\frac{1}{M}$ corresponding to a single edge is a thousandth of the original for 1000 generated grids.

In Figure 3.10b, $s_{ref}$ is the length of maximal probability in the original data set which also indicates the beginning of the exponential behaviour of the original distribution; $s_\mu$ is the artificial counterpart, calculated as the average length of maximal probability for 1000 generated grids with the standard deviation $s_\sigma$. In Figure 3.10c, $\beta_\mu$ is the average exponential parameter from the 1000 fits, with standard deviation $\beta_\sigma$. While it seems from the discrepancy of $s_{ref}$ and $s_\mu$ that without any regulations on line lengths it is difficult to achieve a similarly probable small distance as in the real grid data, the reference fitline for the exponential part of the original data in Figure 3.10c lies within the standard deviation of all single fits performed for the 1000 generated grids.

It is noteworthy that the original data is scattered around the exponential average line length distribution of the generated grids, despite the fact that line lengths were not explicitly optimised when creating connections. The same method of comparison was chosen for all other Dutch grids, and the line lengths of the artificial grids corresponding to all other Dutch grids showed similar...
Chapter 3 Statistical generation of distribution grid models

exponential behaviour, but also the same discrepancies of \( s_{\text{ref}} \) and \( s_{\mu} \) as can be expected since the generation algorithm was the same in all cases. Table 3.4 presents the partly large differences between the exponential distribution fits of the 1000 artificial samples and their real counterparts.

![Line length distribution of the NL1 grid compared to 1000 generated grids](image1)

(a) Line length distribution of the NL1 grid compared to 1000 generated grids

![Peak of the line length distribution of the NL1 grid compared to 1000 generated grids](image2)

(b) Peak of the line length distribution of the NL1 grid compared to 1000 generated grids

![Exponential fit of the line length distribution of the NL1 grid compared to 1000 generated grids](image3)

(c) Exponential fit of the line length distribution of the NL1 grid compared to 1000 generated grids

Figure 3.10: Comparison of the line lengths of the NL1 grid and generated grids

However, far better results can be seen when comparing the line lengths of single generated grids with their real counterparts. This is shown in a further analysis, where single random grids are generated with the statistical settings of each of the Dutch grids and then compared as an ensemble, with a distance reference normalised to the unit square to allow for a meaningful comparison.
### 3.3 Validation

Table 3.4: Comparison of line length distribution fits

<table>
<thead>
<tr>
<th>Distribution peak location</th>
<th>Exponential Fit Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \mu_s ) (km)</td>
<td>( \beta_{\text{ens}} )</td>
</tr>
<tr>
<td>( \sigma_s ) (km)</td>
<td>( s_{\text{ref}} ) (km)</td>
</tr>
<tr>
<td>NL 1</td>
<td>2.4870</td>
</tr>
<tr>
<td>NL 2</td>
<td>1.1873</td>
</tr>
<tr>
<td>NL 3</td>
<td>1.0417</td>
</tr>
<tr>
<td>NL 4</td>
<td>2.0259</td>
</tr>
<tr>
<td>NL 5</td>
<td>3.6264</td>
</tr>
<tr>
<td>NL 6</td>
<td>1.4937</td>
</tr>
<tr>
<td>NL 7</td>
<td>4.3481</td>
</tr>
<tr>
<td>NL 8</td>
<td>0.7346</td>
</tr>
<tr>
<td>NL 9</td>
<td>0.7911</td>
</tr>
<tr>
<td>NL 10</td>
<td>1.2112</td>
</tr>
<tr>
<td>NL 11</td>
<td>0.9743</td>
</tr>
<tr>
<td>NL 12</td>
<td>1.5115</td>
</tr>
</tbody>
</table>

This approach is justified by the structural similarity of the 12 Dutch grids, as seen in the behaviour of their degree distributions shown in Appendix A. With the difference of being slightly skewed towards higher values, it can be seen in Figure 3.11 that the generated connections have reasonably distributed line lengths, with similar exponential distribution shapes as until a relative length of about 0.5. Past this length, the generated grids continue to exhibit very similar behaviour to the real grid data, with probabilities mostly staying within the range of \( \frac{1}{M} \) to \( \frac{10}{M} \), i.e. these lengths are attributed to very few lines as \( \frac{1}{M} \) represents the probability assigned to a single connection.

Summarising the line length analysis, the inability of the algorithm to create links with small line lengths can currently be explained mainly by the choice of the spatial point process, since the links are already encouraged to be added between nodes of shortest distance. Nevertheless, in average the line lengths of real grids lie scattered around those of the generated grids, showing that the generated grids’ line lengths do not stray too far from reality.
Figure 3.11: Comparison of 12 Dutch grids’ normalised line lengths with 12 corresponding generated grids’ normalised line lengths
3.4 Graphical user interface for topology generation

To ensure usability, a GUI was implemented in MATLAB which enables the generation of distribution grid models in the three steps presented in Section 3.2. First, MV networks are created with desired average node degree, varying the degree of meshedness through parity and the RU factor defined in Section 3.2.1. LV feeders are then attached to each MV node following Section 3.2.2. The electrical properties and scale of distance of the lines can then be specified in a last step, with a default setting of values typical for the German grid [65]. The output of the GUI is a file containing

- geographical information on nodes as coordinates;
- the electrical parameters which were entered;
- information on branching of feeders;
- the adjacency matrices of the LV networks;
- the sparse admittance matrix of the overall grid.

Generated grids can be stored in a format which can be saved for further modification, such as adding LV networks of several different characteristics to an existing MV grid, or changing the electrical parameters. Figures 3.12 and 3.13 shows the general structure of the GUI in its MV grid generation mode which can be switched to LV mode once an MV grid has been created. The GUI’s functionalities are explained in further detail in [60, 66].
Chapter 3 Statistical generation of distribution grid models

Figure 3.12: Distribution grid generator GUI with a generated MV network

Figure 3.13: Distribution grid generator GUI with a generated LV network
3.5 Conclusion

In this chapter, an algorithm for the statistical generation of distribution networks was presented. Since the algorithm is based on a spatial distribution of nodes, it can easily be related to arbitrary distance references and electrical parameters depending on distance, such as line impedances. The algorithm is split into two parts: the creation of a MV grid and the generation and attachment of LV grids.

For the MV algorithm, the JDD is incorporated into the algorithm instead of only relying on the commonly used degree distribution, to have more leverage on topological properties such as the likelihood of ringed versus radial structure. Following an extensive study by Kerber [61] on German LV networks, the LV networks are chosen to be radial feeders with no more than 3 levels of branching.

In order to validate the developed algorithm, generated grids were compared to real MV distribution grids. It could be shown that the degree distributions, JDDs and line lengths of the generated grids could reproduce those of the Dutch grids, but did not match the Italian grids well, which is explained with the topological properties of the Italian grids which are tree-like, leaving little scope of approximation through link addition for the minimal-spanning-tree-based MV generation algorithm. This is the main limitation of the algorithm, meaning that ringed topologies can be produced with matching statistics, while a good match is increasingly difficult as the desired grids become more radial.

There are three main topics of future research which can be derived from this work.

Firstly, through the use of the generation algorithm, the testing of any grid-related methods such as control strategies can be conducted with large samples of grids, solving the problem of data unavailability when specific topologies are not of note. This is relevant for achieving robustness of such methods, since on one hand results can be shown to be independent of specific grid configurations by testing several grids with similar statistics, and on the other hand differences of performance in grids with different types of topologies can be readily analysed.

Secondly, the current algorithm can be improved for the better geographical representation of current distribution grids, assuming that changes in the grid structure will not happen in a short time span. In particular, the uniform point process which was used could be exchanged for a point process which is
more likely to form clusters, motivated by the fact that especially in Europe the distribution grid often was constructed for pre-existing settlements and cities. A better approximation of real MV line length distributions, which were not targeted in the presented algorithm, could also possibly be achieved through a more custom point process. At the same time, the simplicity of the currently chosen point process with no need for custom adjustments can be considered an advantage as well, since the generated grids are intended as easily accessible models for testing.

Thirdly and finally, the LV network generation could be complemented by the formulation of another algorithm which enables studies on LV grid topologies which are currently atypical, e.g. with a highly meshed structure.
Chapter 4

Generation of single load profiles for simulation and analysis

The progressing development of concepts aiming at small sets of electrical loads in the grid, ranging from smart city quarters to microgrids or home energy systems, calls for new approaches to describe electrical demand when modelling or simulating these systems. This is due to the fact that the commonly used SLPs are only meaningful when studying statistically significant numbers of loads, since they only represent an average behaviour of thousands of loads [67]. For smaller sets of loads, load profiles with a higher resolution and with more fluctuation are needed in order to study a more realistic behaviour. The use of SLPs or the scaling down of clustered loads are typical representatives of top-down approaches to load modelling.

The bottom-up approach on the other hand uses the behaviour of the single electrical consumers constituting the overall demand [68, 69, 70, 71]: either through the measurement of real appliance data e.g. through smart meters, or through the modelling of the time varying behaviour. The drawback is the effort spent, and the difficulty of assuring randomness in a realistic way. The deployment of a smart meter just for testing purposes needs both a financial investment as well as a sufficient amount of time to collect measurement data. While the mathematical modelling of each electrical appliance does not require the purchase of any physical devices, it is cumbersome and does not bring any proportionate benefit to the overall analysis of energy systems at household or
building level, let alone in city quarters.

The algorithm introduced in this chapter can be considered a bottom-up approach, however it is based on randomly generated consumption events attributed to certain appliance categories instead of relying on measurement data of single appliances. A similar approach for household load profiles can be found in [20], in the sense that appliances are categorised according to their type of use and their type of occurrence; several additional inputs were introduced such as the setting of typical working hours and bedtime, yielding realistic results. At the same time, the number of necessary inputs increases due to adjustments of single user behaviour, and though the generated profiles can be generally considered realistic, they do not capture improbable but possible behaviour such as activity at unusual times, unless explicitly adjusted.

The two novelties of the algorithm presented in this chapter are

- the automated definition of user behaviour through the use of the SLPs as a reference for user activity throughout the day, thereby ensuring that for a sufficient number of generated profiles the resulting demand curves follow a realistic average behaviour;

- the generality of the framework, enabling the generation of load profiles for arbitrary consumers, thus offering the possibility to consider small sets of mixed-use consumers which may co-exist in neighbourhoods as seen e.g. in the EU project COOPERATE\(^{(i)}\).

Households were chosen as a specific example for implementation.

### 4.1 Preliminaries

This section introduces some definitions and annotations as well as specifications on the availability of data which are needed in the subsequent chapters.

#### 4.1.1 Definitions

Each day is considered to be divided into equidistant time intervals \(I_1, \ldots, I_N\), \(N \in \mathbb{N}\), which match the time resolution \(t_{SLP}\) of the SLP which serves as a

\(^{(i)}\)\url{http://cooperate-fp7.eu/}
reference for user activity.

\begin{align*}
    t_{SLP} \cdot N &= 24h, \quad (4.1) \\
    I_j &= [(j-1) \cdot t_{SLP}, j \cdot t_{SLP}), \quad j = 1, \ldots, N. \quad (4.2)
\end{align*}

A load event or simply event \( X \) refers to a single operation of an electrical device, and is characterised by its duration \( l_X \) in terms of number of consecutive active intervals as well as its active power consumption \( P_X \) at time \( t \) after starting interval \( I_{start} \):

\[ P_X(t), \quad t \in \bigcup_{i=0}^{t_X-1} I_{start+i}. \]

All events can be considered to be either

- user-driven load events, denoted with the letter \( X \), which occur when users switch devices on or off;

- or load events independent of users, denoted with a subscript as \( X_{ind} \) which occur automatically at predefined times without involvement of the users.

A load category is a group of electrical loads which serve the same purpose, independently of the specific device or brand. For an arbitrary consumer type, the set \( \{C_1, \ldots, C_M\} \), \( M \in \mathbb{N} \) denotes the set of categories which later contain the respectively generated load events. The total energy consumption per day for a specific consumer is referred to as \( E_{total} \), the statistical average total energy consumption per day as \( \bar{E}_{total} \). Likewise, the energy consumption per day for a specific consumer restricted to category \( C_i \) is denoted as \( E_{C_i} \), whereas the average energy consumption per day per consumer for category \( C_i \) is written as \( \bar{E}_{C_i} \).

### 4.1.2 Data

SLPs for individual homes with an annual consumption of 1000 kWh are shown in Figure 4.1. Apart from the SLPs which are freely available for several types of consumers in Germany, individual data on appliances typical for the considered consumer type are used in the algorithm.
4.1 Preliminaries

![Standard load profiles for German households in different seasons](image)

Figure 4.1: Standard load profiles for German households in different seasons

**Statistics on appliances** For the generation of a meaningful number of load events from certain categories, statistics on their relative share

\[
\frac{\bar{E}_{C_i}}{\bar{E}_{total}}, \ i = 1, \ldots, M
\]

of energy consumption compared to the total consumption are needed as input to the algorithm. When no such data is available, \( M \) is set as 1. For households, the Energie Agentur NRW\(^{(ii)}\) conducted a statistical analysis of \( M = 10 \) consumption categories in the German state of North-Rhine-Westfalia. Some results of this study related to load categories in households are presented in Table 4.1.

**Consumption of single appliances** When available, measured profiles can form events corresponding to certain categories. Especially for devices which have very characteristic demand curves, e.g. with certain cycles, the availability of measurement data results in a more realistic load profile in which it is also possible to highlight the time of occurrence of these specific events.

In the specific example of households, the active power consumption of a washing machine and a dishwasher were measured and are presented in Figure 4.2. Both devices exhibit distinctive behaviour due to heating cycles and pumping.

\(^{(ii)}\) [www.energieagentur.nrw.de](http://www.energieagentur.nrw.de)
Chapter 4 Generation of single load profiles for simulation and analysis

<table>
<thead>
<tr>
<th>Category</th>
<th>2011</th>
<th>2006</th>
</tr>
</thead>
<tbody>
<tr>
<td>Refrigerating</td>
<td>14.9%</td>
<td>15.8%</td>
</tr>
<tr>
<td>IT &amp; communication</td>
<td>12.9%</td>
<td>12.2%</td>
</tr>
<tr>
<td>TV &amp; audio</td>
<td>11.9%</td>
<td>11.1%</td>
</tr>
<tr>
<td>Hot water</td>
<td>11.9%</td>
<td>11.5%</td>
</tr>
<tr>
<td>Lighting</td>
<td>9.7%</td>
<td>11.1%</td>
</tr>
<tr>
<td>Cooking</td>
<td>9.0%</td>
<td>8.4%</td>
</tr>
<tr>
<td>Dryer</td>
<td>6.6%</td>
<td>10.1%</td>
</tr>
<tr>
<td>Dishwasher</td>
<td>5.1%</td>
<td>5.4%</td>
</tr>
<tr>
<td>Washing machine</td>
<td>4.9%</td>
<td>5.1%</td>
</tr>
<tr>
<td>Miscellaneous</td>
<td>13.2%</td>
<td>9.3%</td>
</tr>
</tbody>
</table>

Table 4.1: Energy consumption of different categories of household appliances in 2011 compared to 2006

Figure 4.2: Measurement of active power consumption of household devices

(a) Dishwasher

(b) Washing machine
4.2 Development of the generation algorithm

The generation of a load profile consists of three main steps:

- determining the user-independent events;
- generating random user-driven load events from the category statistics;
- distributing them according to the statistics of the underlying SLP.

These steps apply to both the user-driven and user-independent events which will be treated separately in the following sections.

When there is only one category, i.e. $M = 1$ due to lack of data, all events are created according to the logic of user-driven event generation in order to enable the load profile generation despite the lack of information, though the resulting profiles may not be physically as meaningful.

4.2.1 Generation of consumption events

With the statistical data on the share of different categories, it is possible to allocate corresponding shares of the chosen daily energy consumption $E_{\text{total}}$ which in the following will be interpreted as an upper limit. For all load categories, the logic is to determine a random number of events with the constraint that their respective energy consumptions need to add up to a predetermined limit.

**User-independent events** In this section two types of user-independent events are distinguished: fairly constant standby load events and specific devices, which have certain times or cycles of operation. It is assumed without loss of generality that category $C_1$ stands for miscellaneous standby consumption which is assumed to be constant and $C_2, \ldots, C_L$, $L \leq M \in \mathbb{N}$ are the categories comprising other user-independent specific devices if available.

The standby event $X_{\text{standby}} \in C_1$ can then be generated simply by distributing the daily consumption $\bar{E}_{C_1}$ onto the $N$ time steps and dividing by their width $t_{\text{SLP}}$:

$$P_{X_{\text{standby}}}(t) = \frac{\bar{E}_{C_1}}{t_{\text{SLP}} \cdot N}, \quad t \in [0, t_{\text{SLP}} \cdot N],$$

$$l_{X_{\text{standby}}} = \frac{\bar{E}_{C_1}}{N}. \quad (4.3)$$

$$l_{X_{\text{standby}}} = \frac{\bar{E}_{C_1}}{N}. \quad (4.4)$$
In the case of a specific device category $C_i$, $2 \leq i \leq L$, the exact behaviour of any single event $X_{ind} \in C_i$ is known between its start time $t_0$ and duration $t_{SLP} \cdot l_{X_{ind}}$ from the measurements. What remains to be determined is the number $|C_i|$ of these events per day, which is also calculated from the overall consumption $\bar{E}_{C_i}$:

$$E_{X_{ind}} = \int_{t_0}^{t_{SLP} \cdot l_{X_{ind}}} P_{X_{ind}}(\tau) d\tau,$$  

$$|C_i| = \left\lfloor \frac{\bar{E}_{C_i}}{E_{X_{ind}}} + 0.5 \right\rfloor.$$  

(4.5)

(4.6)

Since the number of events is rounded to the next closer integer, this formulation means that the specific daily consumption $E_{C_i}$ in category $C_i$ may be slightly higher or lower than the average $\bar{E}_{C_i}$. This step is repeated for each category of user-independent devices until all user-independent events have been created.

**Measured user-driven events** There are two types of user-driven events: those with and those without measured profiles. For all loads without measurements or without characteristic behaviour it is assumed that their consumption is constant for the sake of generality, since their profiles depend highly on the actual users’ behaviour.

In a first step, the number of events is determined for each category for which measurements are available. Without loss of generality, let these categories be $C_{L+1}, \ldots, C_{L+K}$, $L + K \leq M$. The logic of event generation is similar to the procedure described for user-independent measured devices, with the difference that here, human behaviour is involved and therefore calls for a randomisation. For a device in $C_i$, $L + 1 \leq i \leq L + K$, let $P_X(t)$ describe its power consumption throughout its duration $l_X$ in terms of number of intervals from starting time $t_0$. The energy consumption of the device is

$$E_X = \int_{t_0}^{l_X \cdot t_{SLP}} P_X(\tau) d\tau.$$  

A random number $u \sim U(0,1)$ is drawn to account for the possibility that the user may or may not use the device. Now, when

$$r := \frac{\bar{E}_{C_i}}{E_X} > u$$  

87
is true, an event $X$ with the characteristics $P_X(t)$ and $l_X$ is created. This step is repeated $\lfloor r \rfloor$ times with newly drawn random numbers, each time reducing $\bar{E}_{C_i}$ by $E_X$ until the last repetition when the ratio is smaller than 1. As a result, $n_{C_i}$ events are generated, equal to the number of times the test was positive. Conducting this procedure for all $C_{L+1}, \ldots, C_{L+K}$ determines all user-driven measured events.

**Random user-driven events** Finally, what remains is to generate random user events from the remaining categories $C_{L+K+1}, \ldots, C_M$. In order to generate an arbitrary event $X \in C_i, i \in \{L + K + 1, \ldots, M\}$, the following steps are taken:

1. Randomly select a duration $l_X$ from the uniform distribution on $\{1, \ldots, N\}$ and a positive power consumption value $P_X$, within a certain limit if specified.

2. Calculate the energy for the operation cycle:
   \[ E_X = P_X \cdot l_X \cdot t_{SLP}. \]

3. If the residual daily energy consumption
   \[ \bar{E}_{C_i} - E_X \]
   is larger than 0, generate $X$ with the aforementioned duration and power characteristics. If not, the power is selected in a way that
   \[ \bar{E}_{C_i} = E_X. \]

4. Repeat the first three steps to generate further events from category $C_i$ until the residual energy consumption is 0.

This procedure is executed for every category of user-driven loads, until the list of random user-driven events is complete.

### 4.2.2 Distribution of consumption events

Since the SLP is used as a reference for the probability of electrical consumption activity, it must first be standardised to serve as a PDF. Let the SLP be
defined by the function $P_{SLP}(t)$ with $t \in [0, t_{SLP} \cdot N]$ as defined in Section 4.1. Then the average total energy consumption $\bar{E}_{total}$ is

$$\bar{E}_{total} = \int_{0}^{t_{SLP} \cdot N} P_{SLP}(\tau) d\tau,$$

and the probability of activity for any interval $I \subseteq [0, t_{SLP} \cdot N]$ can be defined as

$$p(I) := \frac{1}{\bar{E}_{total}} \cdot \int_{I} P_{SLP}(\tau) d\tau. \quad (4.7)$$

This definition also ensures that all events occur within 24 hours, or $N$ corresponding time steps, since

$$p([0, t_{SLP} \cdot N]) = 1.$$

**Placement of user-independent events** Since the user-independent events are not placed randomly but according to a schedule, their placement especially serves the calculation of a residual probability distribution for the user-driven event activity. For each event $X$, a starting interval $I_{start}$ is chosen according to the knowledge on its category’s behaviour. Then the interval in which $X$ is active is equal to

$$I_{X} := \bigcup_{i=start}^{start+l_{X}-1} I_{i},$$

and its power consumption throughout the day can be interpreted to be $P_{X}(t)$ for $t \in I_{X}$ and 0 otherwise.

For the purpose of updating the PDF defined in (4.7), a function $P_{ind}(t)$ is defined which describes the total power consumption coming from the user-independent events, with the above mentioned interpretation:

$$P_{ind}(t) := \sum_{i=1}^{L} \sum_{X \in C_i} P_{X}(t). \quad (4.8)$$

The total energy consumption from user-independent events is equivalently defined as $E_{ind}$:

$$E_{ind} := \sum_{i=1}^{L} E_{C_i} = \int_{t=0}^{t_{SLP} \cdot N} P_{ind}(\tau) d\tau. \quad (4.9)$$
4.2 Development of the generation algorithm

Let $P_\Delta$ be the difference of $P_{SLP}$ and $P_{ind}$ cut off at 0:

$$P\Delta(t) := \begin{cases} P_{SLP}(t) - P_{ind}(t) & \text{if } P_{SLP}(t) - P_{ind}(t) \geq 0 \\ 0 & \text{otherwise.} \end{cases}$$

From (4.8) and (4.9) it is possible to calculate the residual of the PDF which was defined in (4.7):

$$p_1(I) := \frac{1}{E_{total} - E_{ind}} \int_I P\Delta(\tau) d\tau. \quad (4.10)$$

It is to be noted that the modification to the difference of the two power functions ensures that the residual PDF remains positive. Additionally, it can be considered to be untypical that the difference is smaller than zero since on one hand the number of user-independent categories can be considered small for average consumer types, and on the other hand if such behaviour were typical, it would be captured in the SLP which averages tens of thousands of consumption profiles. The definitions provided throughout this chapter provide the most general case.

**Placement of user-driven events** Having developed a measure of probability for user activity and having placed all user-independent events, all that remains for each user-driven event is to select a starting time or interval, respectively. For this purpose, the principle of the inverse transform sampling which is based on the probability integral transform [33] is implemented in a simple way. Given the probability for activity in each interval $p_1(I_i), i = 1, \ldots, N$, the interval $[0, 1]$ can be partitioned into $N$ subsets, the lengths of which are equal to the probability of the corresponding intervals:

$$[0, 1] = \bigcup_{i=1}^{N} S_i,$$
where the subsets $S_i$ are defined as

\[ S_i = [s_{i1}, s_{i2}], \quad i = 1, \ldots, N, \]

\[ s_{i1} = \sum_{j=0}^{i-1} p_1(I_j), \]

\[ s_{i2} = s_{i1} + p_1(I_i). \]

For the sake of correctness, $I_0$ is defined as $\emptyset$, and the last interval $S_N$ should include its upper limit 1. A random time slot following the PDF defined in (4.10) can now easily be selected by generating a random number $u \sim U(0,1)$ and relating it to $I_j$ if $u \in S_j$.

The SLP only serves as a reference for the degree of user activity, not of the type or power of the events. This results in profiles which do not limit the variety of possible consumer behaviour, while ensuring through category statistics and a probability proportional to the SLP that the average of a large sample of generated profiles approximately converges to the SLP curve. Since the profile activity is sampled from the SLP, this is to be expected following the law of large numbers.

### 4.3 Performance of the algorithm

In order to demonstrate the characteristics of generated profiles, the specific application on single households is chosen exemplarily, since statistics and data are publicly available for this consumer type.

The statistics on the $M = 10$ categories shown in Table 4.1 are used to calculate the events. In terms of user-independent events, the “Miscellaneous” category provides the basis for the standby consumption event. In households, standby consumption is caused by plugged-in charging devices, transformers, routers, etc.; refrigerators are an example for specific user-independent devices. The refrigerator is the only other user-independent event which is considered, using a measured refrigerating cycle which is repeated throughout the day. For the user-driven events, the “Washing” and “Dishwasher” categories consist of measured profiles which are generated and distributed as described in Section 4.2.2, and all other events are randomly generated.

A reference household with an annual consumption of 1000 kWh is considered
4.3 Performance of the algorithm

Figure 4.3: Generated household load profiles for households with an annual consumption of 1000 kWh
in order to benchmark the statistical behaviour against the SLP, for which a summer weekend day is chosen. Figure 4.3 shows the behaviour of two load profiles generated from the winter work day SLP. Both profiles feature single high peaks which tend to lie around the midday and afternoon or evening peaks of the SLP with the exception of few.

At the same time, the overall behaviour is characterised by several abrupt peaks and drops, more similar to the real profiles of single appliances which were presented in Figure 4.2, and to measured household profiles presented in [20]. Another phenomenon which can be considered realistic is that in many cases, there are further peaks in the proximity of a peak, and few which stand alone. This is due to the fact that events are not created as single time step events but have a random duration, and since they are placed around the peaks of the SLP at a higher probability, there is a high statistical chance that several adjacent peaks will form due to summation of multi-time step events which start around a similar time. Since the load profiles are generated for the same type of household, the minimum power consumption is equal at 11.84 W in both profiles, since additional events were not allocated to all time slots. In the night hours some small peaks with additional 80 W can be seen which can be attributed to the periodically occurring refrigerating event.

In order to examine the performance of the algorithm, a sample of 5000 load profiles was aggregated and averaged. In Figure 4.4, it can be seen that the aggregation of a large number of generated profiles indeed does follow the statistics given by the underlying SLP. Concerning the percentage difference, the consumption of the generated profiles is in average the same as that of the SLP, shown in Figure 4.5. The most notable relative difference occurs in the first few time slots of the day, though small in absolute values; apart from these outliers the variation around the average is not too large at about ±10%.

Comparisons of samples from other SLPs are also shown in the Appendix B and exhibit a similar performance, in general showing slight overestimations in the generated profiles, but with an average difference in consumption of less than 5%. It can therefore be concluded that the proposed algorithm generates statistically correct single load profiles, as suggested by the theoretical foundation.
4.3 Performance of the algorithm

Figure 4.4: Comparison of the average of 5000 single load profiles with the summer weekend day SLP

Figure 4.5: Difference between the average of 5000 single load profiles and the summer weekend day SLP
4.4 Graphical user interface for load profile generation

In order to make the proposed algorithm easily usable, it was implemented in a MATLAB GUI which specialises on household load profiles. Required inputs are the selection of a SLP, the desired total daily energy consumption and the time step $t_{SLP}$. Additionally, it is possible to specify the number of generated profiles, and to select or deselect load categories, updating the percentages of the remaining categories. The output is a single load profile as a time series with time step $t_{SLP}$, or in the case where several load profiles are desired, a set of such profiles numbered from 0. An exporting function to text files is available for further use of the load profiles. A functionality plotting the resulting average profile is included, thus giving a feel for the statistical approximation of the SLP curve for larger numbers of profiles. The GUI is described in more detail in [72] and is shown in Figure 4.6.
4.5 Conclusion

This chapter presented an algorithm for the bottom-up generation of single load profiles based on statistics provided by SLPs. The resulting time series can be used for modelling and simulation, or as estimations for small sets of consumers when measurements are not available. In contrast to other approaches which distinctly belong to the category of bottom-up methods, the proposed algorithm features elements of top-down approaches, taking into account that a large number of generated profiles must be comparable to real statistics.

While households were used as an example, the algorithm is stated in a way that it is suitable for general purposes: by exchanging the SLP for different consumer types or seasons and by adjusting the data on appliances, if available, load profiles can be generated for any type of consumer. In terms of immediate applicability, a GUI was designed for the generation of single household profiles in different seasons.

The statistical validity of the resulting profiles was ensured through the theoretical fundament of inverse transform sampling, coupled with the statistics of demand per load category. Additionally, the performance of the algorithm was demonstrated to be acceptable in a comparison between large samples of generated household profiles and the SLP they were based on.

It will be of great importance, especially in interdisciplinary modelling and simulation scenarios such as the electrical and thermal systems in buildings, to conduct further research on coupled multi-physics load profiles for single arbitrary consumers, possibly in conjunction with user presence models [73], apart from aggregated multi-physics profiles as presented for households in [74].
Chapter 5

Conclusion

In this dissertation, three stochastics-based methods were developed, targeting the needs for

- a fast approach for uncertainty propagation;
- distribution grid models with various topologies;
- electrical load profiles for single consumers featuring typical unsmooth behaviour.

Chapter 2 presented a mathematical proof of quadrature-based nonintrusive Polynomial Chaos, combining the generalised Polynomial Chaos theory with numerical integration methods in order to enable black box applications of Polynomial Chaos. It was shown in several examples that the results of this method were comparable to those of Monte Carlo simulations, which at a sufficiently high number can be treated as realistic. The main benefit of this approach lies in the fact that numerous Monte Carlo simulations can be replaced with very few simulation runs whose results are post-processed in a single step. Limitations arise from the fact that the Polynomial Chaos representation of system quantities involves nested integrals, which become computationally more expensive than Monte Carlo simulations at a rather small number of parameters due to the curse of dimensionality. However, for systems with few random inputs, this method provides a fast way of assessing the overall range of stochastic behaviour of system variables as opposed to simple analyses of mean values. In the case
of 2 parameters, results comparable to thousands of Monte Carlo simulations may be obtained at as little as 36 simulation runs.

As a second method, the distribution grid generation algorithm introduced in Chapter 3 was developed based on principles from Graph Theory and Complex Network Analysis. Past studies on power grid statistics were used as a reference to build procedures for generating medium voltage and low voltage grids, respectively. Medium voltage grid topologies were not only manipulated through the widely used degree distribution, which is shown in literature to be insufficient in the characterisation of e.g. meshedness, but also through the joint degree distribution which fills this gap. Low voltage grid topologies were generated based on extensive grid studies found in literature, but currently do not allow for a large degree of variation. The generated grids were shown to have similar topological behaviour to real grids, comparing their statistics with those of real Dutch medium voltage grids. It is stressed however that the goal of this study is not the realistic planning of grids but the possibility to provide large ensembles of test grid models for the sake of making grid-related algorithms less dependent on specific topologies.

In Chapter 4, a third method was proposed for the generation of statistically correct electrical load profiles of single consumers. The concept was based on the derivation of a probability distribution from standard load profiles, which provided a reference for the degree of user activity. Single load events were generated according to their load categories’ consumption statistics and were then placed throughout a day according to this probability distribution, yielding time series of power demand. The behaviour of these single load profiles was shown to be more abrupt with several sudden peaks, thus being similar to real behaviour of single loads and posing higher challenges to control algorithms or simulation. At the same time, it was also shown that a large ensemble of these artificial load profiles approximated the standard load profile they were based on. A generalistic mathematical framework was developed which is applicable for any type of consumer, ranging from households to industries, as long as statistical data such as standard load profiles are available.

In all three cases, a major factor of consideration was the immediate applicability of the respective methods which partly are mathematically complex. Therefore, graphical user interfaces were created for each method, facilitating their application for further use in arbitrary simulation-based activities.
To conclude this dissertation, each of the three approaches were developed for the enhancement of testing through simulation, in dimensions which are currently often not considered: nonintrusive Polynomial Chaos for the fast analysis of the overall range of probability in systems with uncertainty, random distribution grid models for the robust testing of grid-related algorithms, and finally realistic load profiles for the evaluation of automation concepts such as energy management or control in the presence of the abrupt behaviour of single consumers. As such, the proposed methods unfold their usefulness when deployed in the validation of such applications, and it will be of great interest to evaluate currently used methods or to design new methods with more robustness.
Appendix A

Comparison of degree distributions of real Dutch grids and 1000 generated grids

Figure A.1: Comparison of the degree distribution of the NL2 grid and 1000 generated grids
Figure A.2: Comparison of the degree distribution of the NL3 grid and 1000 generated grids

Figure A.3: Comparison of the degree distribution of the NL4 grid and 1000 generated grids
Appendix A Comparison of degree distributions of real Dutch grids and 1000 generated grids

Figure A.4: Comparison of the degree distribution of the NL5 grid and 1000 generated grids

Figure A.5: Comparison of the degree distribution of the NL6 grid and 1000 generated grids
Figure A.6: Comparison of the degree distribution of the NL7 grid and 1000 generated grids

Figure A.7: Comparison of the degree distribution of the NL8 grid and 1000 generated grids
Appendix A Comparison of degree distributions of real Dutch grids and 1000 generated grids

Figure A.8: Comparison of the degree distribution of the NL9 grid and 1000 generated grids

Figure A.9: Comparison of the degree distribution of the NL10 grid and 1000 generated grids
Figure A.10: Comparison of the degree distribution of the NL11 grid and 1000 generated grids

Figure A.11: Comparison of the degree distribution of the NL12 grid and 1000 generated grids
Appendix B

Comparison of SLPs with generated profiles

Figure B.1: Comparison of the average of 5000 single load profiles with the intermediary season weekend day SLP
Figure B.2: Difference between the average of 5000 single load profiles and the intermediary season weekend day SLP

Figure B.3: Comparison of the average of 5000 single load profiles with the winter weekend day SLP
Figure B.4: Difference between the average of 5000 single load profiles and the winter weekend day SLP
Bibliography


E.ON ERC Band 18
Flieger, B.
Innenraummodellierung einer Fahrzeugkabine in der Programmiersprache Modelica
1. Auflage 2014
ISBN 978-3-942789-17-2

E.ON ERC Band 19
Liu, J.
Measurement System and Technique for Future Active Distribution Grids
1. Auflage 2014
ISBN 978-3-942789-18-9

E.ON ERC Band 20
Kandzia, C.
Experimentelle Untersuchung der Strömungsstrukturen in einer Mischlüftung
1. Auflage 2014
ISBN 978-3-942789-19-6

E.ON ERC Band 21
Thomas, S.
A Medium-Voltage Multi-Level DC/DC Converter with High Voltage Transformation Ratio
1. Auflage 2014
ISBN 978-3-942789-20-2

E.ON ERC Band 22
Tang, J.
Probabilistic Analysis and Stability Assessment for Power Systems with Integration of Wind Generation and Synchronization Measurement
1. Auflage 2014
ISBN 978-3-942789-21-9

E.ON ERC Band 23
Sorda, G.
The Diffusion of Selected Renewable Energy Technologies: Modeling, Economic Impacts, and Policy Implications
1. Auflage 2014
ISBN 978-3-942789-22-6

E.ON ERC Band 24
Rosen, C.
Design Considerations and Functional Analysis of Local Reserve Energy Markets for Distributed Generation
1. Auflage 2014
ISBN 978-3-942789-23-3

E.ON ERC Band 25
Ni, F.
Applications of Arbitrary Polynomial Chaos in Electrical Systems
1. Auflage 2015
ISBN 978-3-942789-24-0

E.ON ERC Band 26
Michelsen, C. C.
The Energiewende in the German Residential Sector: Empirical Essays on Homeowners’ Choices of Space Heating Techniques
1. Auflage 2015
ISBN 978-3-942789-25-7

E.ON ERC Band 27
Rolfs, W. - FCN
Decision-Making under Multi-Dimensional Price Uncertainty for Long-Lived Energy Investments
1. Auflage 2015
ISBN 978-3-942789-26-4

E.ON ERC Band 28
Wang, J. - ACS
Design of Novel Control Algorithms of Power Converters for Distributed Generation
1. Auflage 2015
ISBN 978-3-942789-27-1

E.ON ERC Band 29
Helmedag, A. - ACS
System-Level Multi-Physics Power Hardware in the Loop Testing for Wind Energy Converters
1. Auflage 2015
ISBN 978-3-942789-28-8