TECHNISCHE
UNIVERSITÄT
DARMSTADT

Adaptive Multi-Level Monte Carlo and Stochastic Collocation Methods for Hyperbolic Partial Differential Equations with Random Data on Networks

Vom Fachbereich Mathematik
der Technischen Universität Darmstadt
zur Erlangung des Grades eines
Doktors der Naturwissenschaften
(Dr. rer. nat.)
genehmigte

Dissertation

von

Elisa Strauch, M.Sc.
aus Mainz

Referent:	Prof. Dr. Jens Lang
Korreferent:	Prof. Dr. Rüdiger Schultz
Tag der Einreichung:	14. September 2022
Tag der mündlichen Prüfung:	15. November 2022

Darmstadt 2022

D17

Adaptive Multi-Level Monte Carlo and Stochastic Collocation Methods for Hyperbolic Partial Differential Equations with Random Data on Networks

Accepted doctoral thesis by Elisa Strauch, M.Sc.

Darmstadt, Technische Universität Darmstadt

Date of thesis defense: November 15, 2022
Tag der mündlichen Prüfung: 15. November 2022

Year of publication of the doctoral thesis on tuprints: 2023
Jahr der Veröffentlichung der Dissertation auf tuprints: 2023

Zusammenfassung

In dieser Arbeit entwickeln wir eine zuverlässige und ganzheitlich fehlergesteuerte Quantifizierung von Unsicherheiten für hyperbolische partielle Differentialgleichungen (PDEs) mit zufälligen Daten auf Netzwerken. Das Ziel ist es, adaptive Strategien im stochastischen und im physikalischen Raum mit einer Multilevel-Struktur so zu kombinieren, dass sowohl eine vorgegebene Genauigkeit der Simulation als auch eine Reduktion des Rechenaufwands erreicht wird.

Zunächst betrachten wir hyperbolische PDEs auf Netzwerken, die keine Unsicherheiten enthalten. Wir führen eine Hierarchie von Modellen mit abnehmender Genauigkeit ein, die durch Vereinfachungen von komplexen Modellgleichungen erreicht werden kann. Diese Hierarchie ermöglicht es detaillierte Modelle in Bereichen des Netzwerkes komplexer Dynamik, und vereinfachte Modelle in Bereichen geringer Dynamik anzuwenden. Als Nächstes erweitern wir das Netzwerkproblem um unsichere Anfangsdaten und unsichere Bedingungen, die an den Rand und an innenliegende Netzwerkkomponenten gestellt werden. Um das Verhalten des betrachteten Systems trotz der Unsicherheiten prognostizieren zu können, wollen wir relevante Ausgabegrößen und ihre statistischen Größen, wie Erwartungswert und Varianz, approximieren.

Für die Untersuchung des Einflusses der Unsicherheiten konzentrieren wir uns auf zwei Sampling-Methoden: die weit verbreitete Monte-Carlo-Methode (MC) und die stochastische Kollokation (SC), die eine vielversprechende Alternative darstellt. Folglich liegt das Hauptinteresse dieser Arbeit auf der stochastischen Kollokation. Diese Methoden ermöglichen numerische, für das deterministische Problem bestehende Löser wiederzuverweden, wodurch sich deren Implementierung vereinfacht. Wir entwickeln für beide Methoden einen adaptiven Singlelevel-Ansatz (SL), indem wir adaptive Strategien im stochastischen Raum mit adaptiven physikalischen Approximationen effizient kombinieren. Die physikalischen Approximationen werden mit einer sample-abhängigen Auflösung in Raum, Zeit und Modellhierarchie berechnet. Anschließend wird der Ansatz auf eine Multilevel-Struktur (ML) erweitert. Hierfür koppeln wir physikalische Approximationen mit unterschiedlichen Genauigkeiten so miteinander, dass die Rechenkosten minimiert werden. Mit Hilfe von a posteriori Fehlerindikatoren können wir die Diskretisierung der physikalischen und stochastischen Approximationen so kontrollieren, dass eine vorgegebene Genauigkeit der Simulation gewährleistet wird. Bei den SC-Methoden setzen wir die adaptive stochastische Strategie mittels adaptiver dünnbesetzter Gitter um, die im Gegensatz zu MC-Methoden glatte oder andere spezielle Strukturen im stochastischen Raum ausnutzen können. Außerdem analysieren wir die Konvergenz, den Rechenaufwand und die Komplexität unserer SL- und ML-Methoden.

Um die Zulässigkeit relevanter unsicherer Ausgabegrößen zu untersuchen, stellen wir ein samplebasiertes Verfahren auf. Das Verfahren approximiert die Wahrscheinlichkeit, dass die Ausgabegröße über den gesamten Zeithorizont Werte zwischen einer gegebenen unteren und oberen Schranke annimmt. Hierzu wird die im Allgemeinen unbekannte Wahrscheinlichkeitsdichtefunktion der Ausgabegröße benötigt. Deshalb analysieren wir zusätzlich den Kerndichteschätzer (KDE), der eine Approximation an die Wahrscheinlichkeitsdichtefunktion der Ausgabegröße liefert und in einem Nachbearbeitungsschritt von SC-Methoden sehr kosteneffizient berechnet werden kann.

Als anwendungsbezogenes Beispiel betrachten wir den Gastransport in Pipelines, der durch isotherme Euler-Gleichungen beschrieben werden kann. Wir präsentieren numerische Ergebnisse für zwei Gasnetzwerke mit unsicherer Gasnachfrage und demonstrieren die Zuverlässigkeit der Fehlerkontrolle unserer Methoden, wobei der Erwartungswert einer unsicheren Ausgabegröße approximiert wird. Die numerischen Beispiele zeigen, dass die MC-Methoden aufgrund der hohen Rechenkosten nicht konkurrenzfähig sind und auch, dass die Multilevel-SC-Methode bessere Ergebnisse als der Singlelevel-Ansatz liefert. Des Weiteren wenden wir die KDE-Methode auf den minimalen und den maximalen Druck an den Ausflussknoten des Netzwerkes an.

Abstract

In this thesis, we develop reliable and fully error-controlled uncertainty quantification methods for hyperbolic partial differential equations (PDEs) with random data on networks. The goal is to combine adaptive strategies in the stochastic and physical space with a multi-level structure in such a way that a prescribed accuracy of the simulation is achieved while the computational effort is reduced.

First, we consider hyperbolic PDEs on networks excluding any type of uncertainty. We introduce a model hierarchy with decreasing fidelity which can be obtained by simplifications of complex model equations. This hierarchy allows to apply more accurate models in regions of the network of complex dynamics and simplified models in regions of low dynamics. Next, we extend the network problem by uncertain initial data and uncertain conditions posed at the boundary and at inner network components. In order to predict the behavior of the considered system despite the uncertainties, we want to approximate relevant output quantities and their statistical properties, like the expected value and variance.

For the study of the influence of the uncertainties, we focus on two sampling-based approaches: the widely used Monte Carlo (MC) method and the stochastic collocation (SC) method which is a promising alternative and therefore of main interest in this work. These approaches allow to reuse existing numerical solvers of the deterministic problem such that the implementation is simplified. We develop an adaptive single-level (SL) approach for both methods where we efficiently combine adaptive strategies in the stochastic space with adaptive physical approximations. The physical approximations are computed with a sample-dependent resolution in space, time and model hierarchy. The extension to a multi-level (ML) structure is realized by coupling physical approximations with different accuracies such that the computational cost is minimized. Due to a posteriori error indicators, we can control the discretization of the physical and stochastic approximations in such a way that a user-prescribed accuracy of the simulation is ensured. For the SC methods, we realize the adaptive stochastic strategy by adaptive sparse grids which are able to exploit any smoothness or special structure in the stochastic space, in contrast to MC methods. In addition, we analyze the convergence, the computational cost and the complexity of our SL and ML methods.

In order to validate the feasibility of relevant uncertain output quantities, we propose and analyze a sample-based method which approximates the probability that the quantity takes values between a given lower and upper bound on the whole time horizon. To this end, the usually unknown probability density function (PDF) of the output quantity is required. Therefore, we introduce and analyze a kernel density estimator (KDE) which provides an approximation of the PDF of the output quantity and can be computed cost-efficiently in a post-processing step of SC methods.

As an application-relevant example, we consider the gas transport in pipeline networks which can be described by the isothermal Euler equations and their simplifications. We present numerical results for two gas network instances with uncertain gas demands and demonstrate the reliability of the error control of our methods approximating the expected value of a random

output quantity. The numerical examples show that the MC methods are not competitive due to high computational costs and that the multi-level SC approach outperforms the single-level SC method. Based on the SC methods, we successfully apply the KDE approach to the minimum and maximum pressures at the outflow nodes of the network.

Acknowledgments

First of all, I would like to thank my supervisor Prof. Dr. Jens Lang for his great and permanent support during the last five years. I really appreciate working with him in the research field of uncertainty quantification and the opportunity to realize my own research ideas.

Next, I would like to thank the co-referee Prof. Dr. Rüdiger Schultz and the members of the examination board Prof. Dr. Frank Aurzada, Prof. Dr. Marc Pfetsch and Prof. Dr. Ulrich Reif.

Special thanks goes to Pia Domschke for a lot of helpful advice and for sharing her huge expertise about the transient simulation of gas transport in pipeline networks with me. Moreover, I would like to thank Michi for the pleasant cooperation and the mathematical discussions.

Furthermore, thanks to my colleagues from the research group *Numerical Analysis and Scientific Computing* for the pleasant working atmosphere. In particular, Bogdan, Gabriel, Moritz and Nora for the funny conversations during the lunch break and the great Doppelkopf events. In addition, I would like to thank my proofreaders: Cathi, Christopher, Corinna, Michi, Nora, Philipp and Torben.

Last but not least, I would like to thank my family and friends for their support and loyalty. I am deeply grateful to my parents Agnes and Erhard and my sister Annika for their love and encouragement during my whole life. A huge thanks goes to my partner Torben for his patience and belief in me.

This work was supported by the German Research Foundation within the collaborative research center Transregio 154 *"Mathematical Modeling, Simulation and Optimization Using the Example of Gas Networks"*, more precisely in the subproject B01 *"Adaptive Dynamic Multiscale Approaches"* under the supervision of the principal investigators Prof. Dr. Jens Lang and Dr. Pia Domschke.

Contents

1 Introduction **1**

1.1 Uncertainty Quantification . . . 2
1.2 Kernel Density Estimation . . . 4
1.3 Contribution and Outline . . . 4

2 Deterministic Problem Setting **7**

2.1 Network Description . . . 7
2.2 Preliminaries and Notation . . . 8
2.3 Deterministic Hyperbolic Equations on Networks . . . 12
2.4 Numerical Methods for Simulation . . . 15
2.4.1 An Implicit Box Scheme . . . 16
2.4.2 Abstract Adaptive Strategy . . . 17
2.5 Gas Network Modeling . . . 18
2.5.1 Model Hierarchy for the Flow of Gas in Pipelines . . . 19
2.5.2 Boundary Conditions . . . 22
2.5.3 Coupling Conditions . . . 22
2.5.4 Adaptive Simulation in Space, Time and Model Hierarchy . . . 24

3 Stochastic Problem Setting **29**

3.1 Preliminaries . . . 29
3.1.1 Finite-Dimensional Noise . . . 30
3.2 Hyperbolic Equations with Random Data . . . 31
3.3 Parametrized Hyperbolic Equations with Random Data . . . 32

4 Adaptive Multi-Level Monte Carlo Method **35**

4.1 Single-Level Structure . . . 36
4.1.1 Complexity Analysis . . . 39
4.2 Multi-Level Structure . . . 41
4.2.1 Complexity Analysis . . . 44
4.3 Real-Valued Quantity of Interest . . . 51
4.3.1 Single-Level Structure . . . 51
4.3.2 Multi-Level Structure . . . 53
4.4 Implementation . . . 57
4.4.1 Real-Valued Quantity of Interest . . . 60

5 Adaptive Multi-Level Stochastic Collocation Method 63
5.1 Sparse Grid Interpolation . . . 64
5.1.1 Smolyak Sparse Grid Construction . . . 66
5.1.2 General Sparse Grid Construction . . . 68
5.1.3 Nested Nodes and Properties of the Interpolant . . . 71
5.1.4 Statistical Quantities . . . 73
5.1.5 The Choice of Collocation Points . . . 75
5.1.6 Adaptive Sparse Grid Algorithm . . . 77
5.2 Single-Level Structure . . . 85
5.2.1 Complexity Analysis . . . 87
5.3 Multi-Level Structure . . . 89
5.3.1 Complexity Analysis . . . 92
5.4 Real-Valued Quantity of Interest . . . 98
5.5 Implementation . . . 100
5.5.1 Real-Valued Quantity of Interest . . . 102

6 Kernel Density Estimator 105
6.1 Univariate Case . . . 106
6.2 Multivariate Case . . . 108
6.3 Convergence of the Kernel Density Estimator . . . 110
6.4 Approximation of Probabilities . . . 111
6.4.1 Computational Aspects using Gaussian Kernel . . . 112
6.5 Boundary Correction Method . . . 115
6.6 Post-Processing Step of SC Methods: Feasibility Check . . . 117

7 Application to Uncertain Gas Transport 119
7.1 A Small Gas Network with Compressor Stations and Valve (GasLib-11) . . . 120
7.1.1 Feasibility Check of the Pressure at the Exits . . . 127
7.2 A Large Gas Network with Compressor Stations (GasLib-40) . . . 132

8 Conclusion and Outlook 139
8.1 Conclusion . . . 139
8.2 Outlook . . . 140

List of Acronyms 143

Bibliography 145

Chapter 1

Introduction

The importance of natural gas transport has significantly increased in the last years due to the ongoing transition from fossil fuels to renewable energies, for example by shut downs of coal-fired plants. The role of gas has been changed from a pure energy source to an additional transport and storage option of energy generated from renewable resources like wind and solar power. In the future, natural gas will be replaced by climate-neutral gas, like biogas and low-carbon hydrogen [55]. Therefore, the comprehension of the dynamics in gas networks still remains important. Typically, the aim of operating a gas network is to achieve a reliable and efficient gas transport which means that the proper amount of gas has to be distributed to the consumers with the required quality and the operational costs as well as the contractual penalties have to be minimized. Due to the complexity of this operational task, modeling, numerical simulation and optimization of gas transport are necessary.

Renewable energies depend on the weather and have a seasonally fluctuating nature. Consequently, the electricity production varies which leads to fluctuations in the demand of the gas transportation. Furthermore, the gas consumption of consumers can also be uncertain due to unexpected weather changes or inaccurate weather forecasts like the temperature of the environment. The influence of these intra-day uncertain oscillations on the gas transport needs to be investigated in order to understand and predict the behavior of the system despite the uncertainties. This challenge is a key issue of uncertainty quantification and demands computing statistical quantities. To this end, the simulation has to be extended to these dynamic fluctuations. In order to assess the reliability of the gas transport, we are interested in how likely relevant output quantities meet the prescribed bounds over the whole time horizon. For example, we can validate if the delivered gas is within the pressure limits specified in a contract between gas company and consumer.

In the last decades, the modeling, simulation and optimization of gas transport in networks has been extensively investigated in many publications [9, 14, 61, 85, 91]. The gas dynamics can be described by the Euler equations which belongs to the class of hyperbolic partial differential equations (PDEs). Based on assumptions and simplifications, different models can be derived and arranged in a hierarchy with decreasing fidelity [26, 29, 75]. Typically, the computational effort of the simulation increases with the fidelity of the selected model. Since the gas dynamics can vary in space and time, Domschke et al. [29] developed an adaptive algorithm which uses simplified models in regions of the network with low dynamics and more detailed models in the case of complex dynamics so that the computational effort can be reduced. Furthermore, this cost-efficient algorithm adaptively adjusts the spatial and temporal discretization and it provides an automatic control of the accuracy of the simulation [25, 29]. Recently, the role of uncertain demands in gas networks has increased and is widely investigated in the context of simulation

and probabilistic constrained optimization [35, 36, 44, 45, 90]. In all these approaches, there is no automatic control of the accuracy of the simulation of uncertain gas transport.

The purpose of this thesis is to develop a reliable and fully error-controlled quantification of the transport of uncertainties through networks such that a prescribed accuracy of the simulation is automatically achieved while the computational effort is reduced as much as possible. To this end, we want to combine two separate concepts which are well-established in uncertainty quantification (UQ): (i) adaptive strategies in the stochastic and physical space and (ii) multi-level structures. Since we have access to the adaptive algorithm developed in [29] for gas transport in networks, we focus on sampling-based methods. These methods allow to reuse any numerical solver that already exists for the simulation without uncertainties. Moreover, we develop an efficient approach for the validation of the pressure of the delivered gas or other relevant quantities by applying a kernel density estimation. In this work, we generalize the setting to any physical phenomena on networks which can be described by hyperbolic PDEs, for example traffic flow, material flow or the aforementioned gas transport problems.

1.1 Uncertainty Quantification

Uncertainty quantification (UQ) plays an important role in application if the required input data or model parameters are subject to uncertainty because of a lack of knowledge, natural fluctuations or imprecise data (e.g. measurements). The main interest of this relatively young research field is to study how the uncertainties influence the so-called *quantity of interest* (QoI) which is typically the solution or a functional of the solution. An example for the QoI in gas networks are the compressor costs or the pressure at a certain node integrated over time. The uncertainties in the input data are modeled by random variables and transfer to the solution of the model such that the QoI is also uncertain. In order to predict the behavior of the system despite these uncertainties, we want to approximate the QoI and its statistical quantities, like the expected value and the variance. A general introduction to UQ can be found in [40, 73].

For the forward propagation in UQ, there are three main approaches: Stochastic Galerkin (SG), Monte Carlo (MC) and stochastic collocation (SC) methods. These approaches can be divided into two classes: intrusive and non-intrusive methods. For hyperbolic systems of conservation laws, the application of the three approaches is discussed in detail in [1]. The SG method is an intrusive approach and is based on generalized polynomial chaos expansion. In this approach, a highly expanded system of coupled equations has to be solved and existing numerical solvers for the deterministic problem cannot be reused. The other two approaches are non-intrusive: The physical (deterministic) dimension is fully separated from the random dimension. Therefore, we only have to solve decoupled deterministic problems which correspond to specific points in the stochastic space and do not include any uncertainties. The QoI of these decoupled problems, the so-called *samples*, are independent of each other and are needed to approximate the desired statistical quantities. Accordingly, these approaches are also called sampling-based methods. If a solver for the deterministic problem is already available, then these methods are highly preferred since we can reuse the original deterministic solver without changes. This is a great advantage and strongly simplifies the implementation of non-intrusive methods. Additionally, we can solve the problems in parallel because the problems are decoupled.

Monte Carlo (MC) methods are very widely used because they are robust even for a lack of regularity of the solution. In addition, they are easy to implement and the convergence rate is independent of the stochastic dimension. Therefore, they are preferred especially for problems with very high-dimensional stochastic spaces. However, the convergence rate of 0.5 is rather small

and usually a huge number of problems has to be solved to achieve an adequate accuracy. Since computing numerical solutions is often computationally expensive, we can get huge computation times. SC methods are based on a polynomial interpolation in the stochastic space using samples, i.e. the QoI of deterministic problems. Therefore, we obtain an interpolant of the uncertain QoI, which provides a lot of statistical information in a post-processing step. Compared to MC, SC methods do not choose the needed points randomly, but according to a fixed structure. For moderate stochastic dimensions and a smooth dependency on the uncertain parameters, sparse grid methods allow us to solve significantly fewer problems than in the case of MC. This is an important advantage because a numerical solution is often computationally expensive.

In order to enhance the computational efficiency of MC methods, multi-level strategies were first introduced as a variance reduction technique in an abstract setting by Heinrich [53]. This approach extends the MC setting where only one problem with a fixed resolution is considered to a hierarchy of subproblems with increasing resolution and couples the subproblems via a telescoping sum. Moreover, Giles [42] extended the multi-level Monte Carlo method (MLMC) to stochastic differential equations in financial engineering and proposed a numerical algorithm to choose the number of samples adaptively. Barth et al. [10] and Cliffe et al. [18] developed MLMC methods for elliptic PDEs with uncertain data using a hierarchy of uniform mesh refinements for the spatial approximation. Considering hyperbolic PDEs, the MLMC method was combined with finite volume discretizations for the physical space using uniform refinements in [76–78]. Several years later, Teckentrup et al. [106] extended the SC approach by a multi-level structure.

The efficiency of MLMC methods applied to uncertain PDEs can be further improved by using adaptive spatial mesh refinements obtained via error estimators. In [32], a hierarchy of spatial meshes was generated by adaptively refining an initial spatial mesh based on expected values of local sample-dependent error estimators. For the computation of failure probabilities, Elfverson et al. [33] performed a selective refinement for every sample in an MLMC method. Moreover, Detommaso et al. [22] proposed to consider the level parameter of the multi-level approach as a continuous variable and to construct different level hierarchies which are determined independently for each sample. Kornhuber and Youett [64] replaced the hierarchy of uniform spatial meshes by a sequence of spatial tolerances which specify the desired accuracy of the computed samples.

In the context of high-dimensional quadrature, Gerstner and Griebel [38] introduced adaptive sparse grid methods which adaptively select the sample points in the stochastic space and are able to exploit any smoothness or special structure with respect to the stochastic parameters. These methods were used with SC methods for PDEs with uncertain data in [49, 81, 95], but without a multi-level structure. Recently, Lang et al. [67] combined adaptive sparse grids in the stochastic space and adaptive sample-dependent mesh refinements for the spatial approximation with a multi-level structure. Until then, spatial approximations which are locally refined for each sample, were only considered in MLMC methods [22, 32, 33, 64]. Thereafter, Feischl and Scaglioni [34] proved the convergence of an SC method which is based on the adaptive sparse grid algorithm of [49] and extended, similar to [67], by adaptive sample-dependent spatial refinements.

In this thesis, we focus on SC and MC methods since we want to reuse existing efficient deterministic solvers. We extend the fully adaptive multi-level stochastic collocation (MLSC) approach for elliptic PDEs [67] to hyperbolic PDEs with random data on networks. Adaptive sparse grids in the stochastic space are efficiently combined with physical approximations computed individually for each stochastic sample point in a space-time-model adaptive manner. Moreover, the multi-level structure is realized in such a way that the computational cost is minimized and the

usage of error estimators enables us to achieve a user-prescribed accuracy of the approximated QoI. For the purpose of comparison with the SC method, we analyze a fully adaptive MLMC approach where again the same concepts, adaptivity and multi-level, are combined. A similar approach was presented for random elliptic problems in [64], but to the author's knowledge not so far for hyperbolic PDEs with random data on networks.

1.2 Kernel Density Estimation

The validation of real-valued QoIs, like the minimum and maximum pressures at the delivery points over time, implies the computation of the probability that the QoI lies in the prescribed range. For this computation, the probability density function (PDF) of the QoI needs to be integrated. However, the PDF as well as the probability distribution of the QoI are usually unknown. Applying a kernel density estimator (KDE) provides an approximation of the required PDF which can be performed in a post-processing step of MC or SC methods since the KDE is based on samples of the QoI. In the case of MC methods, the samples which are already computed can be reused of course. But if more samples are needed to obtain an accurate approximation, then new samples has to be computed which can be computationally expensive, especially for complex problems. In contrast to the MC approach, SC methods provide an approximation of the QoI which only needs to be evaluated in order to get samples for the KDE approach. This evaluation is extremely cheap and has the advantage that no computational effort is invested in solving further deterministic problems.

The KDE was originally introduced in [86, 92] and became a widely used data-smoothing method (e.g. [46]). In the context of stationary gas networks and transient flow networks, the approach was applied to compute probabilities in optimization problems with probabilistic constraints [98]. In this thesis, we analyze the KDE for univariate and multivariate QoIs. Furthermore, we investigate different techniques how to compute probabilities by integrating the KDE in MATLAB®.

1.3 Contribution and Outline

In this thesis, we develop a reliable and fully error-controlled quantification of the transport of uncertainties through a network problem. We extend the fully adaptive MLSC method introduced recently for elliptic PDEs by Lang et al. [67] to hyperbolic PDEs with uncertain initial data, uncertain boundary conditions and to some extent uncertain coupling conditions on networks. Moreover, we provide an extensive analysis of the convergence and the computational complexity of this method. For comparison, we propose a fully adaptive MLMC method, similar to the elliptic setting in [64], and analyze its convergence and complexity, too. In a post-processing step of the considered methods, we propose the validation of uncertain real-valued QoIs on the basis of a KDE.

In **Chapter 2**, we describe the setting of hyperbolic PDEs on networks without any uncertainties, the so-called deterministic (network) problem. First, we consider the definition of networks based on directed connected graphs and extend common function spaces to the network case. We introduce a general hyperbolic PDE and its entropy solution on networks. If the considered physical process can be described by different hyperbolic PDEs, then we arrange the equations in a model hierarchy with decreasing fidelity. Consequently, we can use the simplified models in regions of the network with low dynamics and more accurate models in the case of complex dynamics so that the computational cost can be reduced. Furthermore, we allow to supplement the model hierarchy by a suitable quasi-stationary equation as the simplest model and adapt the

definition of the solution appropriately. Afterwards, we present an implicit box scheme [63] which is used in our computations and the general concept of adaptive algorithm for simulations. We assume to have such an adaptive deterministic black box solver at hand since we only consider UQ methods which reuse the deterministic solver, see Chapters 4 and 5. Finally, we concentrate on gas networks including compressor stations and valves. We describe the gas transport on a pipe network by isothermal Euler equations in one spatial dimension. Then, we derive a model hierarchy based on [24, 29] and present an adaptive simulation algorithm in space, time and model hierarchy developed for gas transport by using adjoint-based error estimators [24, 29].

In **Chapter 3**, we extend the deterministic network problem described in Chapter 2 by uncertain input data and uncertain conditions posed at the boundary and at inner network components. In order to formulate the stochastic problem, we introduce the needed solution spaces and the definition of finite-dimensional noise which is assumed for SC methods. The uncertain data are modeled by random fields with given distribution. Since the uncertainties propagate to the solution of the problem, the solution is a random field as well. The derived stochastic formulation has an infinite-dimensional probability space as stochastic space and constitutes the basis for the MC approach. Assuming that the uncertain data can be described by a finite number N of real-valued random variables ξ_i (finite-dimensional noise assumption), we can transfer the stochastic formulation to the finite-dimensional image space of the random vector $\xi = (\xi_1, \ldots, \xi_N)$. This parametrized problem is purely deterministic and is considered in SC methods. Furthermore, we extend the definition of the deterministic network solution given in Chapter 2 to the two new problems including uncertainties.

In **Chapter 4**, we develop a fully adaptive MLMC method which approximates the expected value of the QoI. We consider the two common QoIs: the solution of the stochastic network problem itself and a functional of the solution. First, we analyze the single-level (SL) approach where we combine an adaptive choice of the number of samples with adaptive computations of each sample computed for a given physical tolerance. Then, we extend it by a multi-level structure such that the computational cost is reduced. Due to these adaptive strategies, both methods are fully controlled by a prescribed total accuracy of the expected value of the QoI. For both methods, we analyze the error, the computational cost and the complexity. Finally, the implementation of the methods is explained.

Chapter 5 is concerned with the development and analysis of the fully adaptive MLSC method, an effective and promising alternative to MC methods. As in Chapter 4, we apply the method to the full solution of the parametrized network problem as well as to a functional of the solution. SC methods are based on an interpolation in the stochastic space and thus they provide an approximation of the QoI. In contrast, MC methods handle only statistical quantities. Since we use global polynomials for the interpolation, we assume a smooth dependence of the QoI on the stochastic parameters. First, we explain the construction and main properties of sparse grid interpolation operators for multi-dimensional spaces. Then, we describe how to compute statistical quantities using the approximated QoI and present the adaptive algorithm described in [81]. We combine the adaptive sparse grid algorithm with the adaptive physical solver which is assumed to be available for the deterministic network problem such that we get a fully error-controlled method. Then, we extend this method by a multi-level structure and analyze its convergence and complexity. Finally, we present the whole algorithm for the single-level and multi-level approach as pseudocode. In the context of uncertain gas transport, the results concerning functionals of the solution are already published by the author in collaboration with Lang and Domschke in [68].

In **Chapter 6**, we investigate an approach based on KDEs to check the feasibility of a real- or function-valued QoI. Using the approximated QoI given by SC methods, this post-processing step is computationally very cheap. First, we introduce the KDE in order to approximate the usually unknown PDF of a random variable which takes values in $\mathbb{R}^d$. Then, we analyze the convergence of this estimator and describe how to approximate the probability that the random variable meets some prescribed bounds. Furthermore, we show the convergence of this approximated probability and study computational aspects using a common choice of the KDE. In the case of bounded random variables, the KDE has a larger support than the exact PDF. For this case, we consider a boundary correction method which provides an approximated PDF with the same support as the exact PDF. Some intermediate results of Chapter 6 are already published by the author in collaboration with Schuster, Gugat and Lang in [98].

In **Chapter 7**, we present numerical experiments of uncertain gas transport. We apply the fully adaptive SL and MLSC methods to gas networks and compare them with the fully adaptive SL and MLMC methods. We investigate two networks from a public gas library in order to show the reliability of the error control and especially the efficiency of the fully adaptive MLSC method. Furthermore, we apply the KDE approach in a post-processing step of the SC methods in order to check the feasibility of the pressure at the exits.

Finally, we summarize our investigations and show interesting open questions for future research in **Chapter 8**.

Chapter 2

Deterministic Problem Setting

In this chapter, we describe a network of hyperbolic partial differential equations (PDEs) with suitable initial values, boundary and coupling conditions. In order to reduce the computational cost of the simulation, we use hyperbolic equations with different fidelity assuming that all models describe the same physical process. Afterwards, we present an implicit box scheme and the general concept of adaptivity in the simulation of hyperbolic PDEs. Finally, we concentrate on gas networks including their components and the modeling of gas transport in pipelines. For this case, we present a specific adaptive algorithm based on a posteriori error estimators.

2.1 Network Description

Let $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ be a directed, connected and finite graph with the set of vertices $\mathscr{V} = \{v_1, \dots, v_R\}$, also called nodes, and edges $\mathscr{E} = \{e_1, \dots, e_P\} \subset \mathscr{V} \times \mathscr{V}$. Each edge $e_j = (v, w) \in \mathscr{E}$ is directed from node v to node w with $v \neq w$ and corresponds to the interval $\Omega_j := (a_j, b_j)$ with $-\infty < a_j < b_j < \infty$ and finite length $L_j := b_j - a_j > 0$. For each vertex $v \in \mathscr{V}$, we define the index set $\delta^+(v)$ of outgoing edges and the index set $\delta^-(v)$ of ingoing edges by

$$\delta^+(v) := \{j : e_j = (v, w) \in \mathscr{E}, w \in \mathscr{V}\},$$
$$\delta^-(v) := \{j : e_j = (w, v) \in \mathscr{E}, w \in \mathscr{V}\}.$$

Hence, the index set of all edges which belong to vertex $v \in \mathscr{V}$ is given by $\delta(v) := \delta^+(v) \cup \delta^-(v)$. The incidence matrix $\boldsymbol{A} \in \mathbb{R}^{R \times P}$ of the graph $\mathscr{G}$ is given by

$$\boldsymbol{A}_{ij} = \begin{cases} 1 & \text{if } j \in \delta^+(v_i), \\ -1 & \text{if } j \in \delta^-(v_i), \\ 0 & \text{otherwise} \end{cases}$$

for $i = 1, \dots, R$ and $j = 1, \dots, P$. The component $\boldsymbol{A}_{ij}$ indicates whether the edge e_j is connected to the node v_i (differing in ingoing or outgoing edge) or not. The set $\mathscr{V}$ of nodes can be divided into two disjoint sets: boundary and inner/coupling nodes. Boundary nodes are connected to the graph by only one single edge and are defined by

$$\mathscr{V}_\partial := \{v_i \in \mathscr{V} : |\delta(v_i)| = 1\}$$

with the index set

$$\mathcal{I}_\partial := \{i : v_i \in \mathscr{V}_\partial\}.$$

Coupling nodes $\mathscr{V}_0 := \mathscr{V} \setminus \mathscr{V}_\partial$ connect at least two edges and can represent not only junctions but also different network components, for example valves and compressors in gas networks. The set

$$\mathcal{I}_0 := \{i : v_i \in \mathscr{V}_0\} = \{i : v_i \in \mathscr{V},\ |\delta(v_i)| \geq 2\}$$

contains the indices of all coupling nodes. In order to illustrate the introduced notation, we provide an example of a small network in Figure 2.1. For a well-arranged design of the network, we allow to add edges with zero length. Note that every one-dimensional interval $(a, b) \in \mathbb{R}$ can be represented as a graph having one edge and two nodes.

In order to describe the behavior of each component in the network, we use algebraic equations as well as hyperbolic partial differential equations. Usually, we store only the state of the edges instead of the whole network since the state of a node can be defined implicitly by the states of the edges connected to this node [62].

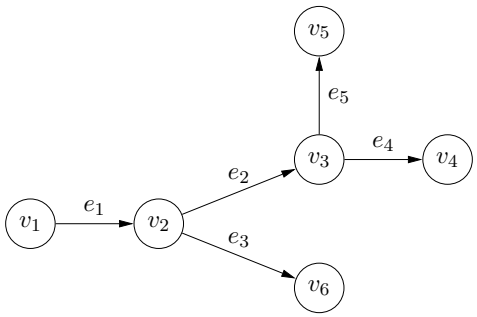


Figure 2.1: Example of a network with the set of nodes $\mathscr{V} = \{v_1, v_2, v_3, v_4, v_5, v_6\}$ and edges $\mathscr{E} = \{e_1, e_2, e_3, e_4, e_5\}$. The set $\mathcal{I}_\partial = \{1, 4, 5, 6\}$ contains the indices of the four boundary nodes v_1, v_4, v_5 and v_6. The two remaining nodes v_2, v_3 are coupling nodes and thus we have $\mathcal{I}_0 = \{2, 3\}$. Considering node v_3, there are two outgoing edges e_4, e_5 and one ingoing edge e_2 which leads to the index sets $\delta^+(v_3) = \{4, 5\}$ and $\delta^-(v_3) = \{2\}$.

2.2 Preliminaries and Notation

In this section, we introduce basic function spaces which we extend to networks. Let $G \subset \mathbb{R}^n$ be a domain. We denote the space of all functions $v : G \to \mathbb{R}$ which are continuous on G by $C(G)$. If the domain G is bounded, the function space $C(\bar{G})$ with the norm

$$\|v\|_{C(\bar{G})} = \max_{x \in \bar{G}} |v(x)|$$

is a Banach space, see for example [113]. Let $(\mathcal{W}, \|\cdot\|_{\mathcal{W}})$ be a Banach space. The space $C([0, T], \mathcal{W})$ for $T > 0$ contains all continuous functions $v : [0, T] \to \mathcal{W}$ endowed with the norm

$$\|v\|_{C([0,T],\mathcal{W})} = \max_{t \in [0,T]} \|v(t)\|_{\mathcal{W}}.$$

The space $C([0, T], \mathcal{W})$ is a Banach space [73, Chapter 1]. Moreover, the set of continuously differentiable functions $v : G \to \mathbb{R}^d$ is described by the space $C^1(G, \mathbb{R}^d)$.

The Lebesgue space $L^p(G)$ with $p \in [1, \infty)$ is defined as

$$L^p(G) = \{v : G \to \mathbb{R} \text{ measurable}, \ \|v\|_{L^p(G)} < \infty\}$$

with the norm

$$\|v\|_{L^p(G)} = \left(\int_G |v(x)|^p \, \mathrm{d}x \right)^{\frac{1}{p}},$$

see [4]. The space $L^\infty(G)$ contains all (equivalence classes of) essentially bounded, measurable functions $v : G \to \mathbb{R}$ and is equipped with the norm

$$\|v\|_{L^\infty(G)} = \operatorname*{ess\,sup}_{x \in G} |v(x)|.$$

The Lebesgue spaces have the following properties [2, Chapter 2].

Proposition 2.1 (Properties of $L^p(G)$)
Let $G \subset \mathbb{R}^n$ be a domain.

(i) $L^p(G)$ is a Banach space for $1 \leq p \leq \infty$.

(ii) $L^2(G)$ is a Hilbert space with the scalar product $(v, w)_{L^2(G)} := \int_G v(x) w(x) \, \mathrm{d}x$.

(iii) If G is a bounded domain, then it holds $L^q(G) \subset L^p(G)$ for $1 \leq p \leq q \leq \infty$.

Extending the Lebesgue spaces to Banach space-valued functions, we introduce the Lebesgue-Bochner space $L^p(G, \mathcal{W})$ which consists of all measurable functions $v : G \to \mathcal{W}$ with finite norm given by

$$\begin{aligned} \|v\|_{L^p(G,\mathcal{W})} &:= \left(\int_G \|v(x)\|_{\mathcal{W}}^p \, \mathrm{d}x \right)^{\frac{1}{p}} && \text{for } p \in [1, \infty), \\ \|v\|_{L^\infty(G,\mathcal{W})} &:= \operatorname*{ess\,sup}_{x \in G} \|v(x)\|_{\mathcal{W}} && \text{for } p = \infty. \end{aligned}$$

For $1 \leq p \leq \infty$, the Lebesgue-Bochner spaces are Banach spaces [73, Chapter 1].

Let $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{N}^n$ be a multi-index and $|\alpha| := \sum_{i=1}^n \alpha_i$ be the length of α. A function $w \in L^2(G)$ is called a weak derivative of $v \in L^2(G)$ of order $|\alpha|$ if

$$\int_G v(x) \, \partial^\alpha \phi(x) \, \mathrm{d}x = (-1)^{|\alpha|} \int_G w(x) \, v(x) \, \mathrm{d}x \qquad \forall \phi \in C_c^\infty(G).$$

We denote the weak derivative w by $\partial^\alpha v$. The Sobolev space $H^1(G)$ consists of all functions $v \in L^2(G)$ with weak derivatives $\partial^\alpha v \in L^2(G)$ for all $\alpha \in \mathbb{N}^n$ with $|\alpha| \leq 1$ [4]. The space $H^1(G)$ is a Hilbert space with scalar product

$$(v, w)_{H^1(G)} = \sum_{|\alpha| \leq 1} (\partial^\alpha v, \partial^\alpha w)_{L^2(G)}$$

which induces the norm

$$\|v\|_{H^1(G)}^2 = (v, v)_{H^1(G)} = \sum_{|\alpha| \leq 1} \|\partial^\alpha v\|_{L^2(G)}^2.$$

Now, the Lebesgue spaces $L^p(G)$ are extended to vector-valued functions $v : G \to \mathbb{R}^d$ with $d > 1$ and afterwards to networks. For this, we first introduce product spaces in general. Let $(\mathcal{W}_i, \|\cdot\|_i)$ be finitely many Banach spaces with $i = 1, \dots, d$. The product space $\mathcal{W}_1 \times \cdots \times \mathcal{W}_d$ is defined by

$$\mathcal{W}_1 \times \cdots \times \mathcal{W}_d := \{y = (y_1, \dots, y_d) : y_i \in \mathcal{W}_i \text{ for } i = 1 \dots, d\}.$$

Proposition 2.2
Let $(\mathcal{W}_i, \|\cdot\|_i)$ be finitely many Banach spaces with $i = 1, \dots, d$.

(i) The product space $\mathcal{W}_1 \times \cdots \times \mathcal{W}_d$ with any of the equivalent norms

$$\|y\|_\infty := \max_{i=1,\dots,d} \|y_i\|_i, \quad \|y\|_p := \left(\sum_{i=1}^{d} \|y_i\|_i^p \right)^{\frac{1}{p}} \quad \text{for } 1 \le p < \infty, \tag{2.1}$$

is a Banach space.

(ii) If $\mathcal{W}_i$ are Hilbert spaces with scalar products $(\cdot,\cdot)_i$, then the product space $\mathcal{W}_1 \times \cdots \times \mathcal{W}_d$ is a Hilbert space with the scalar product $(v, w) = \sum_{i=1}^{d} (v, w)_i$.

(iii) Let $(\mathcal{V}_i, \|\cdot\|_{\mathcal{V}_i})$ be d many normed spaces with $\mathcal{V}_i \subset \mathcal{W}_i$. Then, it holds the property $\mathcal{V}_1 \times \cdots \times \mathcal{V}_d \subset \mathcal{W}_1 \times \cdots \times \mathcal{W}_d$.

Proof. The considered properties of the underlying spaces transfer to the product space by construction. For (i) see [16, Chapter 3 and 5] and for (ii) see [5, Proposition 1.6.2]. □

Consequently, we define for the Lebesgue spaces the extension

$$L^p(G, \mathbb{R}^d) := \underbrace{L^p(G) \times \cdots \times L^p(G)}_{d-times} = \{v = (v_1, \dots, v_d) : G \to \mathbb{R}^d : v_i \in L^p(G) \text{ for } i = 1, \dots, d\}.$$

For the product space, we choose the norm which is a canonical extension of the norm of the underlying space:

$$\|v\|_{L^p(G,\mathbb{R}^d)} := \begin{cases} \left(\sum_{i=1}^{d} \|v_i\|_{L^p(G)}^p \right)^{1/p} & \text{for } p \in [1, \infty), \\ \max_{i=1,\dots,d} \|v_i\|_{L^\infty(G)} & \text{for } p = \infty. \end{cases}$$

This space also results from the construction as a Bochner space $L^p(G, \mathcal{W})$ with $\mathcal{W} = \mathbb{R}^d$ since $\mathbb{R}^d$ is a Banach space with any norm in (2.1). Note that the function space $L^p(G, \mathbb{R}^d)$ is a Banach space for $p \in [1, \infty]$ and even a Hilbert space for $p = 2$. The Hilbert space $H^1(G, \mathbb{R}^d)$ is analogously defined using Proposition 2.2 (ii).

For a domain $G \subset \mathbb{R}$, the total variation of a function $v \in L^1(G, \mathbb{R}^d)$ is defined as

$$TV(v) = \lim_{\epsilon \to 0} \sup \frac{1}{\epsilon} \int_G |v(x+\epsilon) - v(x)| \, \mathrm{d}x,$$

see for example [71]. We denote the space of all functions $v \in L^1(G, \mathbb{R}^d)$ with bounded total variation by $BV(G, \mathbb{R}^d)$.

Focused on one single edge $e_k \in \mathscr{E}$ and the time interval $(0, T)$ with $T > 0$, we denote the physical domain by $D_k := \Omega_k \times (0, T)$ and consider the function spaces $L^p(D_k, \mathbb{R}^d)$. For $1 \leq p < \infty$, this space is isomorphic to $L^p\big((0,T), L^p(\Omega_k, \mathbb{R}^d)\big)$ and the norm can be written as

$$\|v\|^p_{L^p(D_k, \mathbb{R}^d)} = \sum_{i=1}^{d} \int_0^T \int_{\Omega_k} |v_i(x,t)|^p \, \mathrm{d}x \, \mathrm{d}t,$$

due to Fubini's theorem. For $p = \infty$, the isomorphic relation does not hold and we only have $L^\infty\big((0,T), L^\infty(\Omega_k, \mathbb{R}^d)\big) \subset L^\infty(D_k, \mathbb{R}^d)$ [93].

Moreover, the space $C\big([0,T], L^p(\Omega_k, \mathbb{R}^d)\big)$ is required for the definition of weak entropy solutions of hyperbolic PDEs later. In order to consider functions on the whole network, we define the following product space.

Definition 2.3 (Function Spaces for Networks)
For a network described in the previous subsection, we define

$$\begin{aligned} \mathbb{L}^p &:= L^p(D_1, \mathbb{R}^d) \times \cdots \times L^p(D_P, \mathbb{R}^d) \\ &= \big\{ v = \big(v^{(1)}, \ldots, v^{(P)}\big) : v^{(j)} = \big(v_1^{(j)}, \ldots, v_d^{(j)}\big) \in L^p(D_j, \mathbb{R}^d) \text{ for } j = 1, \ldots, P \big\} \end{aligned}$$

with the corresponding norm

$$\|v\|_{\mathbb{L}^p} := \left(\sum_{j=1}^{P} \|v^{(j)}\|^p_{L^p(D_j, \mathbb{R}^d)} \right)^{1/p} \qquad \text{for } p \in [1, \infty),$$

$$\|v\|_{\mathbb{L}^\infty} := \max_{j=1,\ldots,P} \|v^{(j)}\|_{L^\infty(D_j, \mathbb{R}^d)} \qquad \text{for } p = \infty.$$

Additionally, we define functions on the network which are continuous on the time interval $[0,T]$. *The space*

$$\mathbb{C}^0 := C\big([0,T], L^1(\Omega_1, \mathbb{R}^d)\big) \times \cdots \times C\big([0,T], L^1(\Omega_P, \mathbb{R}^d)\big)$$

is equipped with the norm

$$\|v\|_{\mathbb{C}^0} := \max_{j=1,\ldots,P} \|v^{(j)}\|_{C\left([0,T], L^1(\Omega_j, \mathbb{R}^d)\right)}.$$

We denote the vector of the i-th components of $\{v^{(k)}\}_{k=1,\ldots,P}$ by $v_i = \big(v_i^{(1)}, v_i^{(2)}, \ldots, v_i^{(P)}\big)$. Due to the Propositions 2.1 and 2.2, the space $\mathbb{L}^p$ is a Banach space for $p \in [1, \infty]$ and $\mathbb{L}^2$ is a Hilbert space. Furthermore, it holds $\mathbb{L}^p \subset \mathbb{L}^q$ for $1 \leq q < p \leq \infty$ and Hölder's inequality can be generalized for the $\mathbb{L}^1$-Norm.

Lemma 2.4 (Generalized Hölder's inequality)
Let v be a function of the space $\mathbb{L}^2$. Then, it holds

$$\|v\|_{\mathbb{L}^1}^2 \leq C_{T,\Omega}\|v\|_{\mathbb{L}^2}^2$$

with $C_{T,\Omega} = d\,T\sum_{j=1}^{P}|\Omega_j|$.

Proof. We consider the term

$$\|v\|_{\mathbb{L}^1} = \sum_{j=1}^{P}\|v^{(j)}\|_{L^1(D_j,\mathbb{R}^d)} = \sum_{j=1}^{P}\sum_{i=1}^{d}\|v_i^{(j)}\|_{L^1(D_j)}.$$

Applying Hölder's inequality [2, Theorem 2.4] to the $L^1(D_j)$-norm and the Cauchy-Schwarz inequality, we obtain

$$\begin{aligned}\|v\|_{\mathbb{L}^1} &\leq \sum_{j=1}^{P}\sum_{i=1}^{d}\|1\|_{L^2(D_j)}\|v_i^{(j)}\|_{L^2(D_j)}\\ &\leq \left(\sum_{j=1}^{P}\sum_{i=1}^{d}\|1\|_{L^2(\Omega_j\times(0,T))}^2\right)^{1/2}\left(\sum_{j=1}^{P}\sum_{i=1}^{d}\|v_i^{(j)}\|_{L^2(\Omega_j\times(0,T))}^2\right)^{1/2}\\ &= \left(d\,T\sum_{j=1}^{P}|\Omega_j|\right)^{1/2}\|v\|_{\mathbb{L}^2}.\end{aligned}$$

□

2.3 Deterministic Hyperbolic Equations on Networks

On each interval $\Omega_j = (a_j, b_j) \subset \mathbb{R}$ representing an edge $e_j \in \mathscr{E}$ in the network $\mathscr{G} = (\mathscr{V}, \mathscr{E})$, we consider a one-dimensional hyperbolic partial differential equation

$$\partial_t u^{(j)}(x,t) + \partial_x F\big(u^{(j)}(x,t)\big) = G\big(u^{(j)}(x,t), x, t\big) \qquad \text{for } (x,t) \in \Omega_j \times \mathbb{R}_+ \tag{2.2}$$

with flux function $F : \mathbb{R}^d \to \mathbb{R}^d$ and source term $G : \mathbb{R}^d \times \mathbb{R} \times \mathbb{R}_+ \to \mathbb{R}^d$. For a general introduction to hyperbolic PDEs, we refer to [43]. In general, a classical (i.e. continuously differentiable) solution which solves the previous PDE pointwise does not exist so that weak solutions are typically considered. In order to define the solution $u^{(j)}$ at the boundary of Ω_j, we assume that $u^{(j)}(t)$ has bounded variation for all $t \in [0,T]$, i.e. $u^{(j)}(t) \in BV(\Omega_j, \mathbb{R}^d) \subset L^1(\Omega_j, \mathbb{R}^d)$. Therefore, the traces

$$\begin{aligned}u^{(j)}(a_j+,t) &= \lim_{x \searrow a_j} u^{(j)}(x,t),\\ u^{(j)}(b_j-,t) &= \lim_{x \nearrow b_j} u^{(j)}(x,t)\end{aligned}$$

exist [21, 43]. Since weak solutions are not necessarily unique, as discussed for example in [43, Subsection I.4.3], we identify the physically correct solution by the entropy condition and define the weak entropy solution similar to [75].

Definition 2.5 (Entropy Solution)
A function $u^{(j)} \in C\big([0,T], L^1(\Omega_j, \mathbb{R}^d)\big)$ *is a (weak) entropy solution of the PDE* (2.2) *if*

- *The function* $u^{(j)}$ *is a weak solution: It holds* $u^{(j)}(t) \in BV(\Omega_j, \mathbb{R}^d)$ *for all* $t \in [0,T]$ *as well as*

$$\int_0^T \int_{\Omega_j} \Big(u^{(j)} \partial_t \phi + F\big(u^{(j)}\big) \partial_x \phi + G\big(u^{(j)}\big)\phi \Big) \,\mathrm{d}x \,\mathrm{d}t \\ = \int_0^T \Big(F\big(u^{(j)}(a_j+,t)\big)\ \phi(a_j,t) - F\big(u^{(j)}(b_j-,t)\big)\ \phi(b_j,t) \Big) \,\mathrm{d}t$$

  for all $\phi \in C_c^\infty(\bar{\Omega}_j \times (0,T), \mathbb{R})$. *The space* $C_c^\infty(\bar{\Omega}_j \times (0,T), \mathbb{R})$ *contains all infinitely differentiable functions with compact support on* $\bar{\Omega}_j \times (0,T)$.

- *The entropy condition is fulfilled: For all non-negative* $\phi \in C_c^\infty(\Omega_j \times (0,T), \mathbb{R}_+)$, *it holds*

$$\int_0^T \int_{\Omega_j} \Big(\eta\big(u^{(j)}\big)\ \partial_t \phi + \psi\big(u^{(j)}\big)\ \partial_x \phi + \partial_{u^{(j)}} \eta\big(u^{(j)}\big) G\big(u^{(j)}\big)\ \phi \Big) \,\mathrm{d}x \,\mathrm{d}t \geq 0$$

  for all convex entropy-entropy flux pairs $\big(\eta(u^{(j)}), \psi(u^{(j)})\big)$.

Now, we supplement the hyperbolic PDE with initial and boundary conditions which yield the following initial-boundary value problem

$$\partial_t u^{(j)}(x,t) + \partial_x F\big(u^{(j)}(x,t)\big) = G\big(u^{(j)}(x,t), x, t\big), \quad (x,t) \in \Omega_j \times \mathbb{R}_+, \tag{2.3}$$

$$u^{(j)}(x,0) = u_0^{(j)}(x), \qquad x \in \Omega_j, \tag{2.4}$$

$$\mathcal{B}_1\big(u^{(j)}(a_j+,t)\big) = H_1(t), \qquad t \in \mathbb{R}_+, \tag{2.5}$$

$$\mathcal{B}_2\big(u^{(j)}(b_j-,t)\big) = H_2(t), \qquad t \in \mathbb{R}_+. \tag{2.6}$$

The task to pose boundary conditions on $\{a_j, b_j\}$ in (2.5) and (2.6) such that the previous problem is well-posed is in general difficult.

In order to choose proper boundary conditions, the characteristics of the hyperbolic equation [43, Section I.4, Section II.5] and thus the eigenvalues of the Jacobian $\partial_u F(u)$ have to be investigated. Note that, in general, the eigenvalues determine the slope of the characteristics. The boundary conditions need to be chosen depending on the direction of the characteristics at the boundary of Ω_j. For each characteristic which points inwards at the boundary a_j or b_j, we need to pose one boundary condition at the corresponding boundary since we prescribe the initial value in (2.4). Of course, it can occur that no conditions on the boundary are allowed. In this case, the functions $\mathcal{B}_i$ and H_i vanish. Considering equations (2.5) and (2.6), the function $\mathcal{B}_i$ maps the solution to components of the solution $u^{(j)}$ or to linear combinations of the components for which we prescribe the boundary values H_i. Since the direction of the characteristics can depend on time or even on the solution, the number of boundary conditions can vary over time such that the dimension of the image spaces of $\mathcal{B}_i$ as well as of H_i changes or even no condition is allowed. For the standard treatment of boundary conditions and well-posedness results of initial-boundary value problems considering scalar hyperbolic equation as well as linear hyperbolic systems, we refer to [43, Chapter VI] and [66, Chapter 6].

If the considered physical phenomenon can be described by models with different fidelity represented by hyperbolic PDEs, we allow to choose the model for each interval Ω_j separately. In regions of the network of complex dynamics, models with high fidelity should be chosen and in regions of low dynamics, simplified models should be applied. We denote the resulting hierarchy of models with decreasing fidelity by $\mathcal{M} := \{\mathcal{M}_1, \dots, \mathcal{M}_H\}$ and indicate the models by the parameter $m_j \in \mathcal{M}$.

Proceeding with the whole network, we need to pose conditions in order to couple the edges at the inner nodes and to prescribe values on specific boundary nodes by the so-called coupling and boundary conditions. For $t \in \mathbb{R}_+$, we get

$$\partial_t u^{(j)}(x,t) + \partial_x F_{m_j}\big(u^{(j)}(x,t)\big) = G_{m_j}\big(u^{(j)}(x,t), x, t\big), \qquad x \in \Omega_j,\ j = 1, \dots, P \tag{2.7}$$

$$u^{(j)}(x,0) = u_0^{(j)}(x), \qquad x \in \Omega_j,\ j = 1, \dots, P \tag{2.8}$$

$$\mathcal{B}_b\big(u^{(j)}(v_b,t)\big) = H_b(t), \qquad b \in \mathcal{I}_{BC} \subset \mathcal{I}_\partial,\ j \in \delta(v_b), \tag{2.9}$$

$$\mathcal{C}_c\big(u^{(j_1)}(v_c,t), \dots, u^{(j_{N_c})}(v_c,t)\big) = \Pi_c(t), \qquad c \in \mathcal{I}_0,\ j_k \in \delta(v_c),\ k = 1, \dots, N_c, \tag{2.10}$$

with $N_c := |\delta(v_c)|$, solution $u^{(j)} : D_j \to \mathbb{R}^d$ and the physical domain $D_j := \Omega_j \times (0,T)$. As mentioned above, we assume that $u^{(j)}(t) \in BV(\Omega_j, \mathbb{R}^d)$ so that the traces

$$u^{(j)}(a_j+,t) := \lim_{x \searrow a_j} u^{(j)}(x,t) \quad \text{and} \quad u^{(j)}(b_j-,t) := \lim_{x \nearrow b_j} u^{(j)}(x,t)$$

exist [21, 43]. Now, we define the solution $u^{(j)}$ at the nodes belonging to edge e_j by

$$u^{(j)}(v_i,t) := u^{(j)}(a_j+,t) \qquad \text{if } j \in \delta^+(v_i),$$
$$u^{(j)}(v_i,t) := u^{(j)}(b_j-,t) \qquad \text{if } j \in \delta^-(v_i).$$

The index set $\mathcal{I}_{BC} \subset \mathcal{I}_\partial$ describes the indices of the boundary nodes $\mathcal{I}_\partial$ in the network where we have to pose boundary conditions. As mentioned above, the function $\mathcal{B}_b$ specify the components of the solution or a linear combination of it for which we prescribe values H_b at the boundary node v_b. The indices of the coupling nodes, or inner nodes, are denoted by $\mathcal{I}_0$. The function $\mathcal{C}_c$ couples the solutions of all edges which belong to the node v_c and function Π_c is the prescribed coupling constant depending only on time. The coupling of edges can be interpreted as a boundary value problem: If no additional auxiliary variables are introduced, the number of coupling conditions at the inner node v_c has to be equal to the sum of the number of the boundary conditions which are required on each adjacent edge at node v_c. Hence, the number of coupling conditions depends on the direction of the characteristics of adjacent edges and can change over time. Moreover, we assume that the initial values $u_0^{(j)}$, the functions $\mathcal{B}_b$ and H_b as well as the coupling functions $\mathcal{C}_c$ and Π_c are chosen in such a way that the initial-boundary value problem (2.7)–(2.10) is well-posed. In the context of gas networks, we refer to [9, 19, 48, 69, 88] for details.

Beside hyperbolic PDEs, we can also add a quasi-stationary equation to the model hierarchy where the solution can be computed analytically. Typically, this equation is used as the simplest model which should always be chosen if the loss of accuracy is acceptable. We denote the set of hyperbolic models with $\mathcal{M}_{\text{PDE}}$ and the quasi-stationary model with $\mathcal{M}_{\text{QS}}$. For each individual interval Ω_j we choose either a hyperbolic PDE contained in $\mathcal{M}_{\text{PDE}}$ or the quasi-

stationary equation in $\mathcal{M}_{\mathrm{QS}}$. Thus, the full network problem is given by

$$\left.\begin{aligned} \partial_t u^{(j)}(x,t) + \partial_x F_{m_j}\big(u^{(j)}(x,t)\big) &= G_{m_j}\big(u^{(j)}(x,t),x,t\big), && m_j \in \mathcal{M}_{\mathrm{PDE}}, \quad (2.11)\\ \partial_x F_{m_j}\big(u^{(j)}(x,t)\big) &= G_{m_j}\big(u^{(j)}(x,t),x,t\big), && m_j \in \mathcal{M}_{\mathrm{QS}}, \quad (2.12)\end{aligned}\right\} x \in \Omega_j,$$

$$u^{(j)}(x,0) = u_0^{(j)}(x), \qquad x \in \Omega_j, \quad (2.13)$$

for $j = 1,\dots,P$ and $t \in \mathbb{R}_+$. The problem is complemented by boundary and coupling conditions

$$\mathcal{B}_b\big(u^{(j)}(v_b,t)\big) = H_b(t), \qquad b \in \mathcal{I}_{BC},\ j \in \delta(v_b), \quad (2.14)$$

$$\mathcal{C}_c\big(u^{(j_1)}(v_c,t),\dots,u^{(j_{N_c})}(v_c,t)\big) = \Pi_c(t), \qquad c \in \mathcal{I}_0,\ j_k \in \delta(v_c),\ k = 1,\dots,N_c, \quad (2.15)$$

for $t \in \mathbb{R}_+$ and $N_c := |\delta(v_c)|$. For this problem, we extend the definition of a weak entropic solution in [75], where a single branching node is considered, to the whole network and supplement it with the quasi-stationary model.

Definition 2.6 (Entropy Network Solution)
A function $u = \big(u^{(1)},\dots,u^{(P)}\big)$ is called an entropy network solution of the previous problem, if

- *for $m_j \in \mathcal{M}_{\mathrm{PDE}}$, the function $u^{(j)} \in C\big([0,T],L^1(\Omega_j,\mathbb{R}^d)\big)$ is a weak entropy solution of the corresponding PDE, see Definition 2.5,*
- *for $m_j \in \mathcal{M}_{\mathrm{QS}}$, the function $u^{(j)} \in C\big([0,T],C^1(\Omega_j,\mathbb{R}^d)\big)$ is the classical solution of the corresponding quasi-stationary equation,*
- *the initial condition $u^{(j)}(x,0) = u_0^{(j)}(x)$ in (2.13) is satisfied for $j = 1,\dots,P$,*
- *the boundary conditions (2.14) and the coupling conditions (2.15) are satisfied for almost all $t > 0$.*

In order to ensure the continuity in time of the solution of the quasi-stationary equation, we require that the boundary and the coupling conditions are continuous in time. Additionally, the given initial values should be consistent with the boundary conditions and satisfy the quasi-stationary equation for $t = 0$.

2.4 Numerical Methods for Simulation

The exact solution u cannot be computed analytically in general. Therefore, a numerical approximation of the solution is necessary. In order to solve hyperbolic PDEs numerically, several numerical methods are available. A very common approach are finite volume methods [43, 66, 71] where the space-time domain is split into cells, the so-called finite volumes. Integrating the PDE over the cells leads to a formula for computing the average solution inside the cells. A necessary condition for the stability of the scheme is the CFL condition which states that the numerical information needs to be propagated faster than the analytic information. This condition usually provides an upper bound for the time step size. In the case of networks with regions with high dynamics, the CFL condition would determine a very small time step for the whole network. If simulations over a large time span are needed, as in the case of gas transport, this restriction would be disadvantageous since the computational effort would be relatively high. Therefore, we

focus on an implicit box scheme which allows arbitrary large time step sizes since it requires no CFL condition.

Since the behavior of the solution may change dynamically in space and time, we want to adjust the model and the time-space discretization according to the dynamics. Adaptive algorithms provide such an automatic control of the discretizations which reduces the computational cost while ensuring a prescribed accuracy of the simulation. In Subsection 2.4.2, we present the general aspects of adaptive strategies.

2.4.1 An Implicit Box Scheme

In the following, we present a symmetric implicit box (IBOX) scheme which was introduced in [63] for subsonic compressible flow. We use this scheme for a discretization of the isothermal Euler equations in Section 2.5. In [107], numerical tests with a higher order finite volume method showed that the IBOX scheme provides approximations with comparable accuracy for the gas transport in networks.

We consider a general system of balance laws

$$\partial_t u(x,t) + \partial_x F\big(u(x,t)\big) = G\big(u(x,t)\big), \quad (x,t) \in \mathbb{R} \times \mathbb{R}_+$$

with initial condition

$$u(x,0) = u_0(x), \quad x \in \mathbb{R}.$$

Let Δx be the spatial step size and Δt the temporal step size. The implicit box scheme provides the pointwise approximation $U_j^n \approx u(j\Delta x, n\Delta t)$ for $j \in \mathbb{Z}, n \in \mathbb{N}_0$ computed by

$$\frac{U_{j-1}^{n+1} + U_j^{n+1}}{2} = \frac{U_{j-1}^n + U_j^n}{2} - \frac{\Delta t}{\Delta x}\Big(F(U_j^{n+1}) - F(U_{j-1}^{n+1})\Big) + \Delta t \frac{G(U_{j-1}^{n+1}) + G(U_j^{n+1})}{2}.$$

For the initial condition we set

$$U_j^0 = \frac{1}{\Delta x}\int_{X_j} u_0(x)\ \mathrm{d}x$$

with $X_j = [(j-0.5)\Delta x, (j+0.5)\Delta x)$.

Finally, we approximate the weak solution by the piecewise constant function

$$\tilde{u}(x,t) := U_j^n \quad \text{for} \quad (x,t) \in X_j \times T_n$$

with $T_n = [n\Delta t, (n+1)\Delta t)$. This implicit box scheme has order 1 in time and 2 in space [31]. In contrast to time discretizations via explicit methods, the CFL condition which yields an upper bound on the time step size does not have to be satisfied. Therefore, this method allows large time steps which is preferred in simulations over a long time period. However, we need to fulfill an inverse CFL condition:

$$\Delta t \geq \frac{\Delta x}{2\lambda_{min}},$$

where λ_{min} is the smallest absolute value of the eigenvalues of the Jacobian matrix of the flux function $F(u)$.

For scalar hyperbolic PDEs with dissipative source term, the following properties of the IBOX scheme are shown in [63]: The scheme admits a unique solution in every time step, the approximate solutions converge to the entropy solution and the scheme is stable.

As already mentioned in [63], we need to assume that the slope of each characteristic does not change its sign over the considered spatial and temporal interval in order to get a well working numerical method. Considering a one-dimensional interval $(a, b) \subset \mathbb{R}$ with positive length $L = b - a$ which represents an edge in a network, the spatial step size Δx has to be chosen such that $N_x := L/\Delta x$ is an integer. Then, we discretize the interval in N_x subintervals: $a < a + \Delta x < \cdots < a + N_x \Delta x = b$. The IBOX scheme provides $d\,N_x$ equations for $d(N_x + 1)$ unknown variables when $u(x, t) \in \mathbb{R}^d$. Therefore, we have to impose d conditions on the boundary depending on the sign of the characteristic directions: The number of conditions needed on the left boundary $x = a$ and on the right boundary $x = b$ is equal to the number of characteristics entering the interval at the corresponding boundary, i.e. the characteristics having a positive and negative slope, respectively.

2.4.2 Abstract Adaptive Strategy

The aim of adaptive strategies is to control the local resolution in time, space and in model hierarchy such that we can ensure a prescribed accuracy of the simulation and reduce the computational cost. To this end, we perform the classical adaptive loop illustrated in Figure 2.2. First, we solve the hyperbolic problem by a numerical method. Then, we estimate the total error as well as the local temporal, spatial and model error of the computed numerical solution. If the total error estimate is below a given tolerance, we accept the numerical solution. Otherwise, we perform a suitable refinement strategy: We mark the regions where at least one local estimator is too high. For each marked region, we refine the corresponding discretization in time, space and/or model hierarchy and solve the PDE problem using the new discretization. Finally, we repeat the described process.

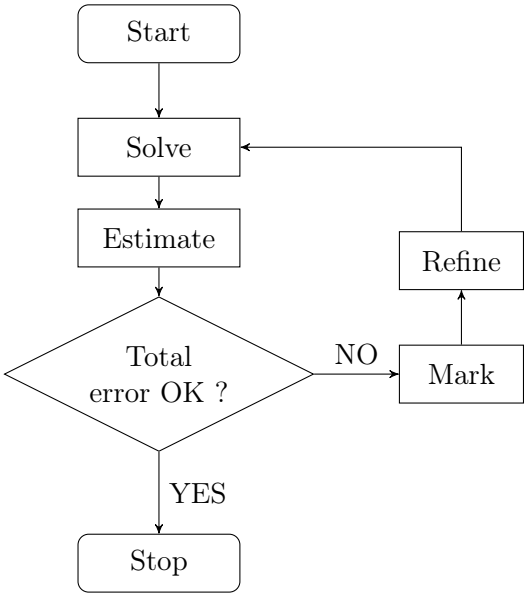


Figure 2.2: Classical adaptation loop for adaptive solvers.

In adaptive schemes, a posteriori error estimators are frequently used since no a priori knowledge about the solution is needed. The error estimates are based on already computed solutions. If the error estimator does not provide a strict upper bound on the exact error, we cannot guarantee that the accuracy of the exact solution is smaller than the prescribed tolerance. In this case, the estimator is only heuristic and is often called error indicator. For nonlinear conservation laws, details on the adaptive scheme and a good review of a posteriori error control based on Kruzkov's doubling of variables technique is given in [84]. Another error estimator is introduced in [57, 58]: A heuristic error indicator based on the weak local truncation error of the numerical solution serves as smoothness indicator in adaptive algorithms.

Furthermore, the influence of the resolution can also be measured in a user-defined functional $\mathcal{F}$, for example the solution integrated over time and space. In the setting of gas networks, we focus on such solution functionals and on error estimators based on solutions of adjoint equations, see Subsection 2.5.4.

Usually, the adaptive scheme performed for a given tolerance $TolH$ provides

$$\|u_h - u\|_X \leq C_h \, TolH,$$

where C_h is a constant and $\|\cdot\|_X$ the corresponding norm. We denote the numerical approximation of the solution u by u_h. For the functional case, we obtain

$$|\mathcal{F}[u_h] - \mathcal{F}[u]| \leq C_h \, TolH.$$

In summary, adaptive strategies control the error for a given tolerance and they usually reduce the computational cost since the discretization is only locally refined where it is necessary and not globally. Especially, fine uniform discretizations for complex problems can lead to a huge computational effort. In the next chapters, we suppose that we have an adaptive deterministic black box solver at hand since we only consider uncertainty quantification methods which reuse the deterministic solver. Therefore, the computational costs are reduced immediately.

2.5 Gas Network Modeling

In this section, we describe the gas transportation in pipelines by isothermal Euler equations in one spatial dimension. We derive a model hierarchy based on [24, 29] and supplement the equations with appropriate boundary and coupling conditions. For more details on gas transport in networks, we refer to [26]. Finally, we present an adaptive procedure in space, time and model hierarchy developed for gas transport in [24, 29]. For the simulation therein, the IBOX scheme discussed in Subsection 2.4.1 is used.

Considering the network described in Section 2.1, edges represent pipes and inner nodes are junctions of pipes or network components as compressor stations or valves. The boundary nodes are either sources where gas is injected into the network or sinks where gas is extracted from the network. The flow direction of the gas does not have to coincide with the orientation of the edges, but it is coupled to this. If the gas flows against the direction of the edge, then the flow has a negative sign.

2.5.1 Model Hierarchy for the Flow of Gas in Pipelines

Since gas pipelines are usually located in the ground, it is reasonable to assume that the temperature is constant along the pipes. Therefore, we model the gas transport in each pipe by isothermal Euler equations. For a general overview of the different models for gas transport in pipelines and their applications we refer to [26, 61].

First, we consider the nonlinear isothermal Euler equations for a horizontal pipe:

$$\begin{aligned} \partial_t \varrho + \partial_x(\varrho\, v) &= 0, \\ \partial_t(\varrho\, v) + \partial_x(p + \varrho\, v^2) &= -\frac{\lambda\, \varrho\, v|v|}{2D_p}, \end{aligned}$$

where ϱ is the density, v the velocity and p the pressure of the gas. We denote the diameter of the pipe by D_p and the friction coefficient is represented by λ. In order to enhance the readability, we omit the space and time dependence of the variables ϱ, v and p. Since we have 2 equations for 3 variables, the equations are supplemented by the equation of state for real gases

$$\varrho = \frac{p}{z(p,T) R_g\, T_g},$$

where R_g denotes the specific gas constant. The compressibility factor $z = z(p, T_g)$ depends in general on the constant temperature T_g and the pressure of the gas. Note that we have $z = 1$ for ideal gases. In the following, we assume that the compressibility factor z is constant. Therefore, we obtain a constant speed of sound $c = \sqrt{p/q} = \sqrt{z R_g\, T_g}$ and, hence, the pressure law

$$p = c^2 \varrho.$$

The friction coefficient λ depends on the Reynolds number R_e and the roughness of the pipe. We focus on [26, Section 2.2] which provides a detailed overview about this topic. If $R_e < 2320$, the flow is laminar and the friction coefficient is computed by $\lambda = 64/R_e$. For turbulent flow, i.e. $R_e \geq 2320$, there exist several models to compute the friction coefficient. We use the recommended formula of Colebrook which determines the friction coefficient implicitly by

$$\frac{1}{\sqrt{\lambda}} = -2 \log_{10}\left(\frac{2.51}{R_e\sqrt{\lambda}} + \frac{k_p}{3.71 D_p}\right),$$

where k_p is the roughness of the pipe. Note that the friction coefficient computed by the above formula depends on the flow rate q, since the Reynolds number also depends on this quantity.

Now, we rewrite the equation in terms of the pressure p and the (volume) flow rate q at standard conditions (1 atm air pressure, temperature of 0°C = 273.15K) given by

$$q = \frac{\varrho\, v A}{\varrho_0} = \frac{q_m}{\varrho_0}.$$

Here, q_m is the mass flow, A the cross-sectional area of the pipe and ϱ_0 the density under standard conditions. In Table 2.1, we list the gas quantities together with their units used in the presented Euler models.

Gas Quantity	Variable	Unit
Density	$\varrho(x,t)$	$[\mathrm{kg/m^3}]$
Pressure	$p(x,t)$	$[\mathrm{Pa}] = 10^{-5}[\mathrm{bar}]$
Flow rate at standard conditions	$q(x,t)$	$[\mathrm{m^3/s}]$
Mass flow	$q_m(x,t)$	$[\mathrm{kg/s}]$
Velocity	$v(x,t)$	$[\mathrm{m/s}]$
Density at standard conditions	ϱ_0	$[\mathrm{kg/m^3}]$
Speed of sound	c	$[\mathrm{m/s}]$
Friction coefficient	$\lambda(q)$	$[-]$
Reynolds number	R_e	$[-]$
Specific gas constant	R_g	$[\mathrm{J/(kg \cdot K)}]$
Diameter of the pipe	D_p	$[\mathrm{m}]$
Cross-sectional area of the pipe	A	$[\mathrm{m^2}]$
Roughness of the pipe	k_p	$[\mathrm{m}]$

Table 2.1: The main physical quantities in gas transport, variables as well as constants.

Using the pressure law $p = c^2\varrho$ and the relation $v = \varrho_0 c^2 q/(pA)$, the nonlinear isothermal Euler equations which is our first and most complex model $\mathcal{M}_1$ has the form:

$$\mathcal{M}_1 : \quad \begin{cases} \partial_t p + \dfrac{\varrho_0 c^2}{A}\partial_x q = 0, \\ \partial_t q + \partial_x \left(\dfrac{A}{\varrho_0} p + \dfrac{\varrho_0 c^2}{A}\dfrac{q^2}{p} \right) = -\dfrac{\lambda \varrho_0 c^2}{2AD_p}\dfrac{q|q|}{p}\,. \end{cases} \tag{2.16}$$

We can rewrite the term in the spatial derivative of the second equation of (2.16) as

$$\frac{A}{\varrho_0}p + \frac{\varrho_0 c^2}{A}\frac{q^2}{p} = \frac{Ap}{\varrho_0}\left(1 + \frac{v^2}{c^2}\right).$$

For small flow rates $|v| \ll c$, we can approximate the term $1 + v^2/c^2$ by 1 as suggested in [85] which leads to the semilinear isothermal Euler equations

$$\mathcal{M}_2 : \quad \begin{cases} \partial_t p + \dfrac{\varrho_0 c^2}{A}\,\partial_x q = 0, \\ \partial_t q + \dfrac{A}{\varrho_0}\,\partial_x p = -\dfrac{\lambda \varrho_0 c^2}{2AD_p}\dfrac{q|q|}{p}\,. \end{cases}$$

For stationary flow, the gas transport is described by the stationary semilinear isothermal Euler equations

$$\begin{aligned} \frac{\varrho_0 c^2}{A}\partial_x q(x) &= 0, \\ \frac{A}{\varrho_0}\partial_x p(x) &= -\frac{\lambda \varrho_0 c^2}{2AD_p}\frac{q(x)|q(x)|}{p(x)}. \end{aligned}$$

These equations depend only on the spatial parameter x and can be solved analytically since the

pressure is always positive. The solution on the interval $[a, b]$ is given by

$$q(x) = const.,$$
$$p(x) = \sqrt{p(x_0)^2 + \frac{\lambda \varrho_0^2 c^2 (x_0 - x)}{D_p A^2} |q(x)| q(x)},$$

where x_0 is an arbitrary point in $[a, b]$. Representing now the flow by the pressure difference yields the Weymouth equation [26]. We consider this stationary model with time-dependent variables resulting in the so-called quasi-stationary semilinear isothermal Euler equations as the most simplest model

$$\mathcal{M}_3 : \quad \begin{cases} \dfrac{\varrho_0 c^2}{A} \partial_x q = 0, \\ \dfrac{A}{\varrho_0} \partial_x p = -\dfrac{\lambda \varrho_0 c^2}{2AD_p} \dfrac{q|q|}{p}. \end{cases} \tag{2.17}$$

In practice, quasi-stationary models are frequently used [24, 47].

Considering the network problem (2.11)–(2.15) for gas transport, we use one of the previously presented models on each pipe Ω_j separately. Therefore, we obtain the pipe-dependent quantities $u^{(j)} = \big(p^{(j)}, q^{(j)}\big), A^{(j)}, D_p^{(j)}$ and $\lambda^{(j)}$. The model hierarchy is given by $\mathcal{M} = \{\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3\}$ and has decreasing fidelity. We recapitulate the models $\mathcal{M}_{\text{PDE}} = \{\mathcal{M}_1, \mathcal{M}_2\}$ and $\mathcal{M}_{\text{QS}} = \{\mathcal{M}_3\}$ in the compact form used in the network problem (2.11)–(2.15):

- $\mathcal{M}_1$: Nonlinear isothermal Euler equations

$$F_{\mathcal{M}_1}\big(u^{(j)}\big) = \begin{pmatrix} \dfrac{\varrho_0 c^2}{A^{(j)}} q^{(j)} \\ \dfrac{A^{(j)}}{\varrho_0} p^{(j)} + \dfrac{\varrho_0 c^2}{A^{(j)}} \dfrac{(q^{(j)})^2}{p^{(j)}} \end{pmatrix}, \quad G_{\mathcal{M}_1}\big(u^{(j)}, x, t\big) = \begin{pmatrix} 0 \\ -\dfrac{\varrho_0 c^2 \lambda^{(j)}}{2A^{(j)} D_p^{(j)}} \dfrac{q^{(j)}|q^{(j)}|}{p^{(j)}} \end{pmatrix}.$$

- $\mathcal{M}_2$: Semilinear isothermal Euler equations

$$F_{\mathcal{M}_2}\big(u^{(j)}\big) = \begin{pmatrix} \dfrac{\varrho_0 c^2}{A^{(j)}} q^{(j)} \\ \dfrac{A^{(j)}}{\varrho_0} p^{(j)} \end{pmatrix}, \quad G_{\mathcal{M}_2}\big(u^{(j)}, x, t\big) = G_{\mathcal{M}_1}\big(u^{(j)}, x, t\big).$$

- $\mathcal{M}_3$: Quasi-stationary semilinear isothermal Euler equations

$$F_{\mathcal{M}_3}\big(u^{(j)}\big) = F_{\mathcal{M}_2}\big(u^{(j)}\big), \quad G_{\mathcal{M}_3}\big(u^{(j)}, x, t\big) = G_{\mathcal{M}_1}\big(u^{(j)}, x, t\big).$$

In order to obtain the complete network problem (2.11)–(2.15) describing the gas transport in pipelines, we now specify the boundary and coupling conditions.

2.5.2 Boundary Conditions

For the aforementioned nonlinear, semilinear and quasi-stationary Euler equations, the boundary conditions are analyzed in [24]. For each gas model, we have to pose exactly one condition at each boundary node: Either the pressure p or the gas flow q is prescribed. Considering the nonlinear and semilinear Euler equations, we always have one entering and one leaving characteristic at each boundary. In a gas network, the boundary nodes are either inflow nodes (sources) $\mathscr{V}_\partial^{in}$, where gas is injected into the network or outflow nodes (sinks) $\mathscr{V}_\partial^{out}$, where gas is taken out of the network.

Using the incidence matrix $\boldsymbol{A}$, we define

$$\mathscr{V}_\partial^{in} := \big\{v_i \in \mathscr{V}_\partial : q^{(j)}(v_i,t)\cdot \boldsymbol{A}_{ij} > 0 \text{ for } j \in \delta(v_i)\big\},$$
$$\mathscr{V}_\partial^{out} := \big\{v_i \in \mathscr{V}_\partial : q^{(j)}(v_i,t)\cdot \boldsymbol{A}_{ij} < 0 \text{ for } j \in \delta(v_i)\big\}$$

as well as the corresponding index sets

$$\mathcal{I}_\partial^{in} := \big\{i : v_i \in \mathscr{V}_\partial^{in}\big\} \quad \text{and} \quad \mathcal{I}_\partial^{out} := \big\{i : v_i \in \mathscr{V}_\partial^{out}\big\}.$$

Therefore, we have $\mathscr{V}_\partial = \mathscr{V}_\partial^{in} \,\dot\cup\, \mathscr{V}_\partial^{out}$ and $\mathcal{I}_\partial = \mathcal{I}_\partial^{in} \,\dot\cup\, \mathcal{I}_\partial^{out}$. A common choice is to prescribe the pressure at the inflow nodes and the gas flow at the outflow nodes for $t > 0$. Then, the boundary conditions written in the form of the operator $\mathcal{B}_b : \mathbb{R}^2 \to \mathbb{R}$ in (2.14) are given by

$$\mathcal{B}_b\big(u^{(j)}(v_b,t)\big) := \begin{cases} u_1^{(j)}(v_b,t) = p^{(j)}(v_b,t) & \text{for } b \in \mathcal{I}_\partial^{in},\ j \in \delta(v_b), \\ u_2^{(j)}(v_b,t) = q^{(j)}(v_b,t) & \text{for } b \in \mathcal{I}_\partial^{out},\ j \in \delta(v_b). \end{cases}$$

Consequently, we have $\mathcal{I}_{BC} = \mathcal{I}_\partial$. The function $H_b : \mathbb{R}_+ \to \mathbb{R}$ prescribes the boundary values depending on the time. For easier readability, we use the notation

$$H_b(t) \equiv p^{v_b}(t) \quad \text{and} \quad H_b(t) \equiv q^{v_b}(t), \text{ respectively.}$$

2.5.3 Coupling Conditions

In order to complete the network problem (2.11)–(2.15), we need to specify the coupling conditions at the inner nodes $\mathscr{V}_0$ following [9, 24]. If no additional auxiliary variables are used, the number of coupling conditions at an inner node is equal to the number of pipes connected to the considered node since we need exactly one boundary condition at each boundary of each pipe. In gas networks, we distinguish between junctions $\mathscr{V}_{ju}$, compressor stations $\mathscr{V}_{cs}$ and valves $\mathscr{V}_{va}$. Compressor stations are used to increase the pressure if the pressure loss caused by friction in the pipes gets too high. Valves control the gas flow in parts of the gas network.

Junctions. Let v_c be a junction with ingoing edges $\{e_k : k \in \delta^-(v_c)\}$ and outgoing edges $\{e_k : k \in \delta^+(v_c)\}$. Junctions $\mathscr{V}_{ju}$ are either pure branching nodes, sources or sinks. At branching nodes, no gas is injected or extracted from the network.

First, we consider branching nodes and require the conservation of mass (Kirchhoff's first rule) by

$$\sum_{j\in\delta^+(v_c)} q^{(j)}(v_c,t) - \sum_{j\in\delta^-(v_c)} q^{(j)}(v_c,t) = 0 \qquad \forall t > 0.$$

This condition implies that the sum of the fluxes going into the node v_c is equal to the sum of fluxes going out of this node. Furthermore, we pose the continuity of pressure

$$p^{(j)}(v_c,t) = p^{(k)}(v_c,t) \qquad \forall j \in \delta^+(v_c),\ k \in \delta^-(v_c)$$

for all time $t > 0$. This condition is frequently used [9, 14, 35, 44, 48] and very common in the engineering community. Other coupling conditions like the equality of enthalpy are discussed in [19, 69, 75, 88]. The previous coupling conditions can be written in the form (2.15) by defining the function $\mathcal{C}_c : \mathbb{R}^{2N_c} \to \mathbb{R}^{N_c}$ as

$$\mathcal{C}_c\big(u^{(j_1)}(v_c,t),\dots,u^{(j_{N_c})}(v_c,t)\big) := \begin{pmatrix} \sum\limits_{j\in\delta^+(v_c)} q^{(j)}(v_c,t) - \sum\limits_{j\in\delta^-(v_c)} q^{(j)}(v_c,t) \\ p^{(j_1)}(v_c,t) - p^{(j_2)}(v_c,t) \\ p^{(j_1)}(v_c,t) - p^{(j_3)}(v_c,t) \\ \vdots \\ p^{(j_1)}(v_c,t) - p^{(j_{N_c})}(v_c,t) \end{pmatrix}$$

and $\Pi_c(t) := 0 \in \mathbb{R}^{N_c}$, where $j_k \in \delta(v_c)$ and $N_c = |\delta(v_c)|$.

Now, we extend the previous equation to inner nodes which are either sources or sinks. We denote the gas flow going into or out of the network at node v_c and the pressure at this node by additional auxiliary variables $q^{v_c}(t)$ and $p^{v_c}(t)$, respectively. The conservation of mass is now ensured by the coupling condition

$$\sum_{j\in\delta^+(v_c)} q^{(j)}(v_c,t) - \sum_{j\in\delta^-(v_c)} q^{(j)}(v_c,t) = q^{v_c}(t).$$

For the continuity of pressure, we require

$$p^{(j)}(v_c,t) = p^{v_c}(t), \qquad j \in \delta^+(v_c), \tag{2.18}$$

$$p^{(j)}(v_c,t) = p^{v_c}(t), \qquad j \in \delta^-(v_c). \tag{2.19}$$

Consequently, the coupling functions $\mathcal{C}_c : \mathbb{R}^{N_c} \to \mathbb{R}^{N_c+1}$ and $\Pi_c : \mathbb{R}^{N_c} \to \mathbb{R}^{N_c+1}$ in (2.15) have the form

$$\mathcal{C}_c\big(u^{(j_1)}(v_c,t),\dots,u^{(j_{N_c})}(v_c,t)\big) := \begin{pmatrix} \sum_{j\in\delta^+(v_c)} q^{(j)}(v_c,t) - \sum_{j\in\delta^-(v_c)} q^{(j)}(v_c,t) \\ p^{(j_1)}(v_c,t) \\ \vdots \\ p^{(j_{N_c})}(v_c,t) \end{pmatrix}$$

and

$$\Pi_c(t) := \begin{pmatrix} q^{v_c}(t) \\ p^{v_c}(t) \\ \vdots \\ p^{v_c}(t) \end{pmatrix}$$

with $j_k \in \delta(v_c)$ and $k = 1,\dots,|\delta(v_c)| = N_c$. In total, we have $N_c + 1$ equations, but $N_c + 2$ unknowns. Therefore, we need to prescribe either the gas flow $q^{v_c}(t)$ or the pressure $p^{v_c}(t)$ at node v_c.

Inserting edges of length 0 (i.e. short edges) can result in a network where sources and sinks are always boundary nodes and no inner nodes. For the considered inner node v_c, a source or a sink, we would insert a short edge $e_k = (v_c, v_l)$ where the new node v_l is a duplicate of v_c. The node v_c would be transformed to a pure branching node and we would have to fulfill the algebraic equations

$$p^{(k)}(v_c, t) = p^{(k)}(v_l, t) \quad \text{and} \quad q^{(k)}(v_c, t) = q^{(k)}(v_l, t), \tag{2.20}$$

since the edge e_k has length 0.

Compressor Stations. In this work, we concentrate on pressure-controlled compressor stations which have an electric motor consuming only electric power and no gas. For simplicity, we consider a compressor station $v_c \in \mathscr{V}_{cs}$ connecting two pipes e_k and e_l. Moreover, we assume that the gas flows from e_k to e_l. The pressure increase generated by the compressor station is described by the equation

$$p^{(l)}(v_c, t) = p^{(k)}(v_c, t) + \Delta p_c(t)$$

with a prescribed profile $\Delta p_c(t)$ of the pressure difference. Due to the electric motor, the same amount of gas which flows into the compressor station leaves it:

$$\boldsymbol{A}_{ck} q^{(k)}(v_c, t) + \boldsymbol{A}_{cl} q^{(l)}(v_c, t) = 0.$$

In summary, we have

$$\mathcal{C}_c\big(u^{(k)}(v_c, t), u^{(l)}(v_c, t)\big) := \begin{pmatrix} p^{(l)}(v_c, t) - p^{(k)}(v_c, t) \\ \boldsymbol{A}_{ck} q^{(k)}(v_c, t) + \boldsymbol{A}_{cl} q^{(l)}(v_c, t) \end{pmatrix}, \quad \Pi_c(t) := \begin{pmatrix} \Delta p_c(t) \\ 0 \end{pmatrix}.$$

Valves. Let $v_c \in \mathscr{V}_{va} \subset \mathscr{V}_0$ be a valve which connects two edges e_k and e_l. In practice, valves are opened or closed slowly in order to regulate the gas flow in a smooth way. Due to modeling issues, we model a valve with only two states: open or closed. An open valve behaves like a branching node and is described by

$$\mathcal{C}_c\big(u^{(k)}(v_c, t), u^{(l)}(v_c, t)\big) := \begin{pmatrix} p^{(k)}(v_c, t) - p^{(l)}(v_c, t) \\ \boldsymbol{A}_{ck} q^{(k)}(v_c, t) + \boldsymbol{A}_{cl} q^{(l)}(v_c, t) \end{pmatrix}, \quad \Pi_c(t) := \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

For a closed valve, we need to impose the condition

$$\mathcal{C}_c\big(u^{(k)}(v_c, t), u^{(l)}(v_c, t)\big) := \begin{pmatrix} q^{(k)}(v_c, t) \\ q^{(l)}(v_c, t) \end{pmatrix}, \quad \Pi_c(t) := \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

In the case that more than two edges are connected to a valve or compressor station, the coupling conditions can be easily adapted.

2.5.4 Adaptive Simulation in Space, Time and Model Hierarchy

In the following, we present the main aspects of the adaptive simulation strategy developed and applied in several papers [24, 25, 27–31]. Based on a posteriori error estimates, the algorithm provides an automatic control of the model distribution and the discretization in space and time such that the model and the discretization errors are reduced up to a prescribed tolerance. In regions where the dynamic of the solution is very low, the simplest model should be sufficient

and in regions where we observe a high dynamical behavior, the model with the highest fidelity should be needed.

Let an arbitrary initial distribution of gas transport models $\{m_1, \dots, m_P\} \subset \{\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3\}^P$ be given. We solve the nonlinear and the semilinear Euler equations on each pipe using the IBOX scheme, see Subsection 2.4.1. The quasi-stationary Euler equations $\mathcal{M}_3$ can be solved analytically. However, we can also apply the IBOX scheme concerning the space discretization, as discussed in [27, Section 6]. On each pipe $\Omega_j = (a_j, b_j)$, we are able to use one of the three gas transport models and a uniform space discretization with an individual spatial step size. For the time discretization, we use for all pipes the same global temporal step size Δt. Together with the initial value, the coupling conditions and the boundary conditions, we obtain a nonlinear systems of equations which has to be solved. Of course, the model and the spatial step size on each pipe as well as the global temporal time step size can also be chosen to vary over time.

Now, the aim is to automatically control the resolution in space, time and model hierarchy such that a prescribed accuracy of the simulation is achieved. We measure the influence of the resolution on a user-defined output functional

$$\mathcal{F}[u] = \int_{\Omega \times (0,T)} N(u) \, \mathrm{d}(x,t) + \sum_{v \in \mathscr{V}} \int_0^T N_v(u) \, \mathrm{d}t \qquad \in \mathbb{R} \tag{2.21}$$

with $\Omega = \prod_{e_j \in \mathscr{E}} \Omega_j = \prod_{e_j \in \mathscr{E}} (a_j, b_j)$. The functions N and N_v are tracking-type cost functions and depend on the solution u. Typical choices of this functional are the compressor costs and the pressure at some sinks integrated over time. In order to control the resolution, we use a posteriori error estimators η_x, η_t and η_m which approximate the discretization and the model errors. The estimators are based on adjoint solutions of the discretized model equations [24, 29].

In order to reduce the effort, the simulation time $[0, T]$ is divided equidistantly into N_t time blocks. For each time block $[T_{k-1}, T_k]$, the model as well as the spatial discretization of each pipe and the time step size do not vary over time. In order to determine the discretization and the models successively, we compute the error estimators in $[T_{k-1}, T_k]$ which provides an approximation of the exact error

$$\mathcal{F}_k[u] - \mathcal{F}_k[u_h] \approx \eta_x^k + \eta_t^k + \eta_m^k.$$

We denote the exact functional on time block $[T_{k-1}, T_k]$ by $\mathcal{F}_k[u]$ and the functional of the numerical solution u_h by $\mathcal{F}_k[u_h]$. In the adaptation strategy, the relative error

$$\frac{|\mathcal{F}_k[u] - \mathcal{F}_k[u_h]|}{|\mathcal{F}_k[u]|} \approx \frac{\left|\eta_x^k + \eta_t^k + \eta_m^k\right|}{|\mathcal{F}_k[u_h]|}$$

was considered so far. However, we consider the absolute error since in the stochastic setting this error is the natural one due to error splitting in a deterministic and stochastic part, see Chapters 4 and 5. Therefore, we modify the existing algorithm appropriately. Let $TolH$ be the total tolerance for the absolute error. Now, we need to distribute this tolerance over the time blocks in order to obtain local tolerances for the error in each time block. We decide to split the tolerance equally: $TolH_k = TolH/N_t$ for $k = 1, \dots, N_t$. In each time block, we successively refine the resolution in space, time and model hierarchy until

$$\left|\eta_x^k + \eta_t^k + \eta_m^k\right| \leq TolH_k$$

is satisfied. Therefore, we obtain at the end of the algorithm

$$|\mathcal{F}[u] - \mathcal{F}[u_h]| \leq \sum_{k=1}^{N_t} |\mathcal{F}_k[u] - \mathcal{F}_k[u_h]| \approx \sum_{k=1}^{N_t} \left|\eta_x^k + \eta_t^k + \eta_m^k\right| \leq \sum_{k=1}^{N_t} TolH_k = TolH.$$

The steps of the adaptive algorithm described in [24] are illustrated in Figure 2.3 and are based on the classical adaption loop shown in Figure 2.2 in Subsection 2.4.2. For each time interval $[T_{k-1}, T_k]$, we evaluate the error estimators. First, we divide the total tolerance $TolH_k$ into a tolerance for the temporal-spatial discretization error and a tolerance for the model error. If the discretization error is too high, we check the temporal and the spatial discretization error separately. The temporal discretization error is considered globally for the whole network. In the case that this error exceeds the given bound, we mark the global time step size for refinement. Then, we compute the spatial discretization error locally for each pipe and mark only the pipes for uniform spatial refinement where the local estimator is not small enough. If any refinements are required, the simulation for the current time block is recomputed and the errors are checked again. If the discretization error is accepted whereas the model error exceeds the given tolerance, we compute the model error estimator locally for each pipe. Then, we mark the pipes where the corresponding model error estimate is greater than the given tolerance so that a model enhancement is needed. In case of marked pipes, we adjust the model on the corresponding pipes and recompute the simulation with the unchanged discretization in space and time. In the case that the total error is smaller than the given tolerance, we check if the discretization meshes can be coarsened and if the models can be switch down. Using the new discretization and model distribution, the simulation for the next time block starts and the process is repeated. More details on the refinement strategy are given in [24, Section 3.3] and [29, Section 5]. In [25], a greedy-like refinement strategy is investigated and compared with the presented strategy in view of computational cost.

The chosen a posteriori error estimators are only heuristic indicators such that no strict bounds on the error are satisfied [29]. However, we can usually determine a constant $C_h > 0$ such that the algorithm provides an upper bound on the error by

$$|\mathcal{F}[u] - \mathcal{F}[u_h]| \leq C_h \, TolH,$$

where C_h is usually close to 1. Eventually, the approximate solution u_h refers to the given tolerance $TolH$.

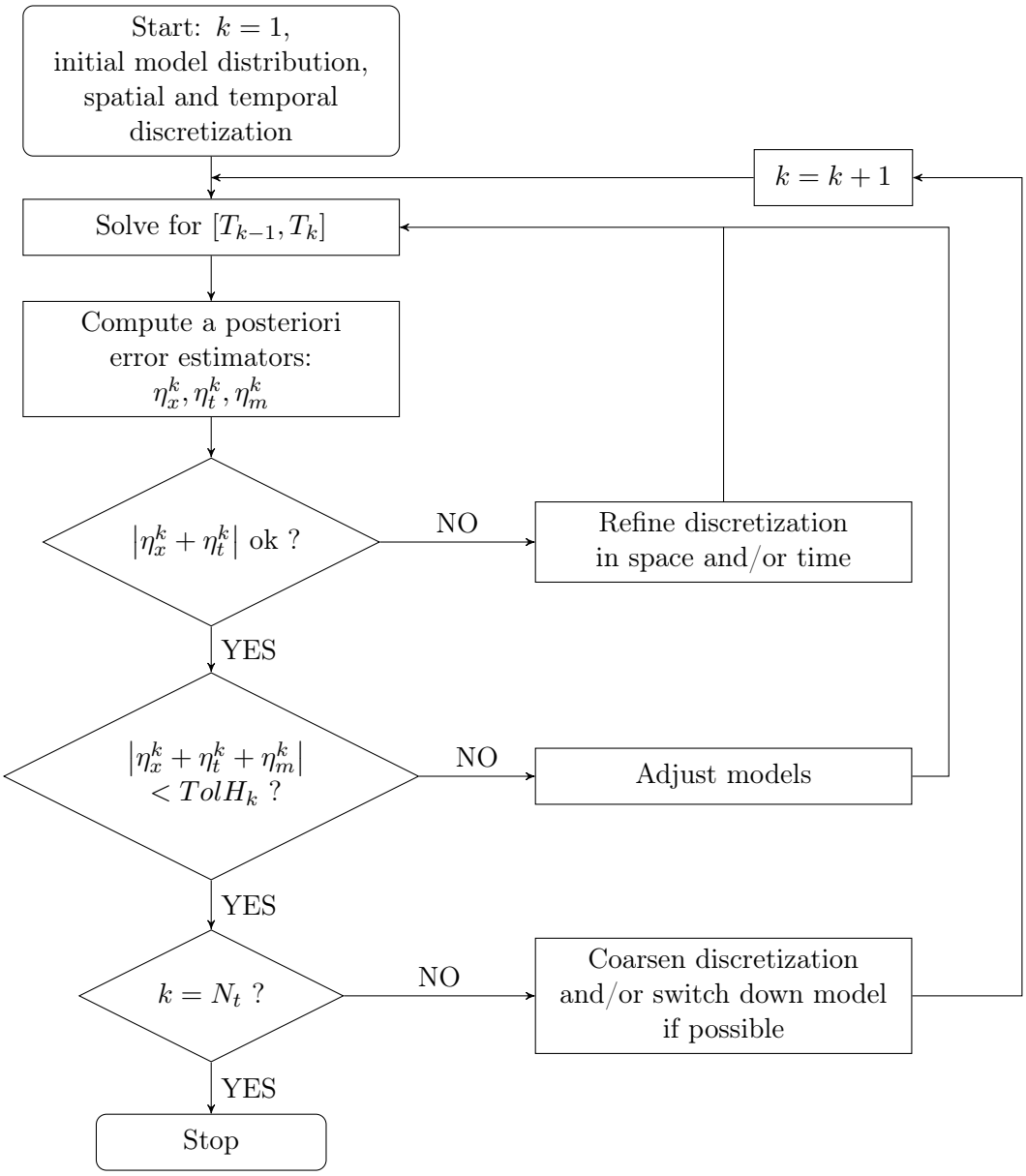


Figure 2.3: Scheme of the adaptive algorithm [24] for error control.

Chapter 3

Stochastic Problem Setting

In this chapter, we extend the deterministic network problem given in the previous section by uncertain data. First, we introduce the solution spaces and other mathematical preliminaries in order to formulate the stochastic problem and its random entropy solution. Then, we parametrize the problem such that the uncertain data is represented by a finite-dimensional space and we define the corresponding stochastic entropy solution.

3.1 Preliminaries

For a brief introduction to probability theory and more details about the solution spaces, we refer to [73]. In order to extend the previous network problem (2.11)–(2.15) by uncertainties, we consider a complete probability space $(\Theta, \Sigma, \mathbb{P})$. The sample space Θ contains all possible outcomes θ and $\Sigma \subset \mathcal{P}(\Theta)$ denotes the sigma algebra, where $\mathcal{P}(\Theta)$ is the power set of Θ. The probability measure $\mathbb{P} : \Sigma \to [0, 1]$ maps every measurable set in Σ onto the interval $[0, 1]$.

Let $(\Psi, \mathcal{Y})$ be a measurable space. A *Ψ-valued random variable* (RV) is defined as a measurable function X from Θ into Ψ. For a given $\theta \in \Theta$, the value $X(\theta) \in \Psi$ is called a *realization* of the RV X. For $\Psi = \mathbb{R}$, we call the mapping X *real-valued* and, for $\Psi = \mathbb{R}^d$ with $d > 1$, the $\mathbb{R}^d$-valued RV is a *random vector*. If Ψ is a function space, X is a function-valued RV which we also call a *random field*. In literature, the definitions of random variables and random fields are not necessarily consistent, see for example [13, 73, 76]. The previous definitions are mainly based on [76].

Let $(\mathcal{W}, \|\cdot\|_{\mathcal{W}})$ be a Banach space. The expected value of a Banach space-valued random variable $X : \Theta \to \mathcal{W}$ is defined as

$$\mathbb{E}[X] = \int_{\Theta} X(\theta) \, \mathrm{d}\mathbb{P}(\theta) \ \in \mathcal{W}.$$

Similar to the Lebesgue-Bochner spaces, the space of mean-square integrable RVs, also called stochastic Bochner space, is defined as

$$L^2(\Theta, \mathcal{W}) = \left\{ X : \Theta \to \mathcal{W} : X \text{ measurable and } \mathbb{E}\left[\|X\|_{\mathcal{W}}^2\right] < \infty \right\}.$$

The space $L^2(\Theta, \mathcal{W})$ with the norm

$$\|X\|_{L^2(\Theta,\mathcal{W})} = \mathbb{E}\left[\|X\|_{\mathcal{W}}^2\right]^{1/2} = \left(\int_{\Theta} \|X\|_{\mathcal{W}}^2 \, \mathrm{d}\mathbb{P}(\theta) \right)^{\frac{1}{2}}$$

is a Banach space [54].

For $\mathcal{W} = \mathbb{R}$, the space $L^2(\Theta, \mathbb{R})$ contains all real-valued RV with finite first and second moment and is usually denoted by $L^2(\Theta)$. Furthermore, the following property holds

$$\left\| \int_\Theta X(\theta) \, \mathrm{d}\mathbb{P}(\theta) \right\|_{\mathcal{W}} \leq \int_\Theta \|X(\theta)\|_{\mathcal{W}} \, \mathrm{d}\mathbb{P}(\theta), \tag{3.1}$$

see [54, 73]. Applying the previous property and Jensen's inequality leads to

$$\|\mathbb{E}[X]\|_{\mathcal{W}} \leq \mathbb{E}\left[\|X\|_{\mathcal{W}}\right] \leq \mathbb{E}\left[\|X\|_{\mathcal{W}}^2\right]^{\frac{1}{2}} = \|X\|_{L^2(\Theta,\mathcal{W})} < \infty$$

for a RV $X \in L^2(\Theta, \mathcal{W})$.

If a Hilbert space $(\mathcal{H}, (\cdot,\cdot)_{\mathcal{H}})$ is used, then the space $L^2(\Theta, \mathcal{H})$ is a Hilbert space [73] with inner product

$$(X, Y)_{L^2(\Theta,\mathcal{H})} = \mathbb{E}\big[(X, Y)_{\mathcal{H}}\big] \qquad \text{for } X, Y \in L^2(\Theta, \mathcal{H}).$$

The variance-like term

$$V[X] := \|\mathbb{E}[X] - X\|_{L^2(\Theta,\mathcal{W})}$$

is finite for RVs $X \in L^2(\Theta, \mathcal{W})$. Similar to [76], we use the inequality $\|a - b\|^2 \leq 2(\|a\|^2 + \|b\|^2)$, property (3.1) and Jensen's inequality to show that

$$\begin{aligned} V[X] = \|\mathbb{E}[X] - X\|_{L^2(\Theta,\mathcal{W})}^2 &\leq 2\Big(\|\mathbb{E}[X]\|_{\mathcal{W}}^2 + \|X\|_{L^2(\Theta,\mathcal{W})}^2\Big) \leq 2\left(\mathbb{E}\left[\|X\|_{\mathcal{W}}\right]^2 + \|X\|_{L^2(\Theta,\mathcal{W})}^2\right) \\ &\leq 4\|X\|_{L^2(\Theta,\mathcal{W})}^2 < \infty. \end{aligned} \tag{3.2}$$

In the case of $\mathcal{W} = \mathbb{R}$, the variance-like term $V[X]$ coincides with the variance

$$\mathbb{V}[X] = \mathbb{E}\left[(X - \mathbb{E}[X])^2\right].$$

3.1.1 Finite-Dimensional Noise

In this part, we present the definition of finite-dimensional noise which will be needed to transform the hyperbolic problem with random data in Section 3.2 from the abstract sample space Θ to a finite-dimensional space. After a change of variable, we obtain a weighted solution space. The following theory is based on [73].

Definition 3.1 (Finite-Dimensional Noise)
Let $(\mathcal{W}, \|\cdot\|_{\mathcal{W}})$ be a Banach space and let $\xi_i : \Theta \to \mathbb{R}$ be real-valued RVs with $i = 1, \ldots, N < \infty$. We denote the image of the RVs by $\Gamma_i := \xi_i(\Theta)$. Let $\xi : \Theta \to \Gamma$ be the resulting finite-dimensional random vector $\xi := (\xi_1, \ldots, \xi_N)$ with $\Gamma := \Gamma_1 \times \cdots \times \Gamma_N \subset \mathbb{R}^N$. Then, a function $v \in L^2(\Theta, \mathcal{W})$ of the form $v\big(\xi(\theta)\big) \in \mathcal{W}$ for $\theta \in \Theta$ is called finite-dimensional noise.

In summary, a finite-dimensional noise is a random field which depends on a finite number of real-valued RVs. Obviously, real-valued RVs are finite-dimensional noise. A typical possibility to obtain finite-dimensional noise is to truncate Karhunen-Loève expansions.

Additionally, we assume that the random vector $\xi := (\xi_1, \dots, \xi_N) : \Theta \to \Gamma \subset \mathbb{R}^N$ has an absolutely continuous distribution. Therefore, the random vector ξ has a joint probability density function (PDF) $\rho : \Gamma \to \mathbb{R}_+$. In order to obtain a straightforward computation of the expected value of v, we can perform a change of variable

$$\mathbb{E}[v] = \int_\Theta v\big(\xi(\theta)\big)\,\mathrm{d}\mathbb{P}(\theta) = \int_\Gamma v(\mathbf{y})\,\mathrm{d}\mathbb{P}_\xi(\mathbf{y}) = \int_\Gamma v(\mathbf{y})\rho(\mathbf{y})\,\mathrm{d}\mathbf{y} \ \in \mathcal{W},$$

where $\mathbb{P}_\xi$ is the probability distribution of ξ. For a finite-dimensional noise $v \in L^2(\Theta, \mathcal{W})$, the previous change of variable provides

$$\|v\|^2_{L^2(\Theta,\mathcal{W})} = \int_\Theta \|v\big(\xi(\theta)\big)\|^2_{\mathcal{W}}\,\mathrm{d}\mathbb{P}(\theta) = \int_\Gamma \|v(\mathbf{y})\|^2_{\mathcal{W}}\,\mathrm{d}\mathbb{P}_\xi(\mathbf{y}) = \int_\Gamma \|v(\mathbf{y})\|^2_{\mathcal{W}}\ \rho(\mathbf{y})\,\mathrm{d}\mathbf{y}.$$

This identity motivates to consider the weighted Bochner space

$$L^2_\rho(\Gamma, \mathcal{W}) = \big\{v : \Gamma \to \mathcal{W} : v \text{ measurable and } \int_\Gamma \|v(\mathbf{y})\|^2_{\mathcal{W}}\ \rho(\mathbf{y})\,\mathrm{d}\mathbf{y} < \infty\big\}.$$

The weighted Bochner space equipped with the norm

$$\|v\|^2_{L^2_\rho(\Gamma,\mathcal{W})} = \int_\Gamma \|v(\mathbf{y})\|^2_{\mathcal{W}}\ \rho(\mathbf{y})\,\mathrm{d}\mathbf{y} =: \mathbb{E}\left[\|v\|^2_{\mathcal{W}}\right]$$

is a Banach space. Usually, we denote $L^2_\rho(\Gamma, \mathbb{R})$ by $L^2_\rho(\Gamma)$. Due to the change of variable, we can associate the finite-dimensional noise $v \in L^2(\Theta, \mathcal{W})$ with $v \in L^2_\rho(\Gamma, \mathcal{W})$ by defining $y_i := \xi_i(\theta)$ and $\mathbf{y} = (y_1, \dots, y_N)$.

If $\rho \in L^\infty(\Gamma)$, then it holds $C(\Gamma, \mathcal{W}) \subset L^2_\rho(\Gamma, \mathcal{W})$: Considering the function $v \in C(\Gamma, \mathcal{W})$, then applying Hölder's inequality provides

$$\|v\|_{L^2_\rho(\Gamma,\mathcal{W})} = \big\|\|v\|^2_{\mathcal{W}}\,\rho\big\|_{L^1(\Gamma)} \le \big\|\|v\|^2_{\mathcal{W}}\big\|_{L^1(\Gamma)}\ \|\rho\|_{L^\infty(\Gamma)}.$$

Both terms on the right hand side of the above inequality are bounded, since $\rho \in L^\infty(\Gamma)$ and $\|v\|^2_{\mathcal{W}} \in C(\Gamma) \subset L^1(\Gamma)$. Since Γ is compact, function v is measurable [54, Chapter 1].

3.2 Hyperbolic Equations with Random Data

In mathematical models, the input data can be subject to uncertainty due to measurements, a lack of knowledge or imprecise data. In this work, we consider the case that the initial data u_0, the boundary data H_b and/or the coupling data Π_c at inner network components in the problem (2.11)–(2.15) are uncertain. The coupling conditions at pure branching nodes remain deterministic. For example, if we consider gas networks with uncertain gas demands at exits which are inner nodes, we have uncertain coupling conditions at these nodes. The uncertainty of the input data propagates to the solution of the model such that the solution is uncertain as well. We model the input data as random fields with given distributions. Moreover, the flux function F_{m_j}, the source term G_{m_j}, the boundary operator $\mathcal{B}_b$ and the coupling function $\mathcal{C}_c$ remain deterministic.

Consequently, we obtain the following random problem:

$$\left.\begin{aligned} \partial_t u^{(j)}(x,t,\theta) + \partial_x F_{m_j}\big(u^{(j)}(x,t,\theta)\big) &= G_{m_j}\big(u^{(j)}(x,t,\theta),x,t\big), && m_j \in \mathcal{M}_{\mathrm{PDE}}, \\ \partial_x F_{m_j}\big(u^{(j)}(x,t,\theta)\big) &= G_{m_j}\big(u^{(j)}(x,t,\theta),x,t\big), && m_j \in \mathcal{M}_{\mathrm{QS}}, \end{aligned}\right\} x \in \Omega_j, \tag{3.3, 3.4}$$

$$u^{(j)}(x,0,\theta) = u_0^{(j)}(x,\theta), \qquad x \in \Omega_j, \tag{3.5}$$

for $j = 1,\dots,P$, $t \in \mathbb{R}_+$ and $\theta \in \Theta$. The problem is complemented by boundary and coupling conditions

$$\mathcal{B}_b\big(u^{(j)}(v_b,t,\theta)\big) = H_b(t,\theta), \qquad b \in \mathcal{I}_{BC},\ j \in \delta(v_b), \tag{3.6}$$

$$\mathcal{C}_c\big(u^{(j_1)}(v_c,t,\theta),\dots,u^{(j_{N_c})}(v_c,t,\theta)\big) = \Pi_c(t,\theta), \qquad c \in \mathcal{I}_0,\ j_k \in \delta(v_c),\ k = 1,\dots,N_c, \tag{3.7}$$

for $t \in \mathbb{R}_+$, $\theta \in \Theta$ and $N_c := |\delta(v_c)|$. The solution is represented as $u^{(j)}(x,t,\theta) : D_j \times \Theta \to \mathbb{R}^d$ with the deterministic physical domain $D_j = \Omega_j \times (0,T)$. All input parameters in the problem and the solution now depend on the additional variable $\theta \in \Theta$. On the one hand, we can interpret the solution as a set of $\mathbb{R}^d$-valued RVs $u^{(j)}(x,t,\cdot) : \Theta \to \mathbb{R}^d$ with $(x,t) \in D_j$. On the other hand, the solution is a function-valued random variable (random field): For a given $\theta \in \Theta$, $u^{(j)}(\cdot,\cdot,\theta) : D_j \to \mathbb{R}^d$ is a deterministic function of space and time which is also called a realization or sample of the random field $u^{(j)}$. Typically, the second interpretation is used. This stochastic formulation of our network problem is the basis for the Monte Carlo approach in Chapter 4. Similar to [76], we define the random solution of the previous problem by extending Definition 2.5 of the deterministic problem in a path-wise manner. Furthermore, we assume the existence and uniqueness of the random solution.

Definition 3.2 (Random Entropy Network Solution)
A random field $u = \big(u^{(1)},\dots,u^{(P)}\big) : \Theta \to C([0,T],L^1(\Omega_1,\mathbb{R}^d)) \times \cdots \times C([0,T],L^1(\Omega_P,\mathbb{R}^d))$ *is a random entropy network solution of the stochastic problem* (3.3)–(3.7) *if, for* $\mathbb{P}$*-almost all* $\theta \in \Theta$*,* $u(\theta)$ *is an entropy network solution of the deterministic problem which results from inserting the fixed variable* θ *into problem* (3.3)–(3.7)*.*

Assumption 3.3
There exists a unique random entropy network solution u *of the problem* (3.3)–(3.7) *such that* $u \in L^2(\Theta,\mathbb{Y})$ *with* $\mathbb{Y} = C\big([0,T],L^1(\Omega_1,\mathbb{R}^d)\big) \times \cdots \times C\big([0,T],L^1(\Omega_P,\mathbb{R}^d)\big)$*.*

For scalar hyperbolic conservation laws with uncertain initial data, Mishra and Schwab [76] have shown the existence and uniqueness of a random entropy solution. In [78], the setting was extended by an uncertain flux function. Recently, well-posedness results for one-dimensional systems of hyperbolic conservation laws with uncertain initial data and uncertain flux function were presented in [41]. For multi-dimensional systems, the existence of the random entropy solution is an open problem since global results about existence and uniqueness of the deterministic solution are not available.

3.3 Parametrized Hyperbolic Equations with Random Data

Next, we want to reduce the infinite-dimensional probability space to an N-dimensional space. We perform the well-established approach including the common assumptions in order to obtain the starting network problem for stochastic collocation methods, see for example [6, 73, 106, 114]. As mentioned in Subsection 3.1.1, we need the following assumption.

Assumption 3.4 (Finite-Dimensional Noise & joint PDF)
The random input fields of the network problem, in our case u_0, H_b and Π_c, are finite-dimensional noise (Definition 3.2): They are described by a finite number N of real-valued random variables $\xi_i : \Theta \to \Gamma_i \subset \mathbb{R}$. The resulting finite-dimensional random vector $\xi := (\xi_1, \dots, \xi_N) : \Theta \to \Gamma$, with $\Gamma := \Gamma_1 \times \cdots \times \Gamma_N \subset \mathbb{R}^N$, has an absolutely continuous distribution, i.e. it has a joint PDF $\rho : \Gamma \to \mathbb{R}_+$. Additionally, it holds $\rho \in L^\infty(\Gamma)$.

If the input data is modeled directly by a finite set of real-valued RVs, then the finite-noise assumption is trivially satisfied. The random vector ξ generates a sigma algebra $\sigma(\xi) \subset \Sigma$ on the sample space Θ. Regarding the stochastic part, the random input fields are measurable with respect to this sigma algebra because of the finite-dimensional noise assumption [73, Lemma 9.40]. Considering the random network solution u of (3.3)–(3.7), we assume that each component $u^{(j)}(x,t,\cdot)$ is measurable with respect to the sigma algebra for every $(x,t) \in \Omega_j \times (0,T)$. Now, the Doob-Dynkin lemma [73, Lemma 4.46] provides that every solution component $u^{(j)}(x,t,\cdot)$ is a function of the random vector ξ, i.e. $u^{(j)}(x,t,\theta) = u^{(j)}(x,t,\xi(\theta))$. Therefore, the solution u is finite-dimensional noise and we can replace the outcome $\theta \in \Theta$ by the realization $\xi(\theta)$ and rewrite the network problem (3.3)–(3.7). For convenience, we omit the dependence on θ in the rewritten problem

$$\left.\begin{aligned} \partial_t u^{(j)}(x,t,\xi) + \partial_x F_{m_j}\big(u^{(j)}(x,t,\xi)\big) &= G_{m_j}\big(u^{(j)}(x,t,\xi),x,t\big), && m_j \in \mathcal{M}_{\mathrm{PDE}},\\ \partial_x F_{m_j}\big(u^{(j)}(x,t,\xi)\big) &= G_{m_j}\big(u^{(j)}(x,t,\xi),x,t\big), && m_j \in \mathcal{M}_{\mathrm{QS}},\end{aligned}\right\} x \in \Omega_j,$$
$$u^{(j)}(x,0,\xi) = u_0^{(j)}(x,\xi), \qquad x \in \Omega_j,$$

for $j = 1, \dots, P$, $t \in \mathbb{R}_+$ and $\theta \in \Theta$. The problem is complemented by boundary and coupling conditions

$$\begin{aligned} \mathcal{B}_b\big(u^{(j)}(v_b,t,\xi)\big) &= H_b(t,\xi), && b \in \mathcal{I}_{BC},\ j \in \delta(v_b),\\ \mathcal{C}_c\big(u^{(j_1)}(v_c,t,\xi),\dots,u^{(j_{N_c})}(v_c,t,\xi)\big) &= \Pi_c(t,\xi), && c \in \mathcal{I}_0,\ j_k \in \delta(v_c),\\ & && k = 1,\dots,N_c,\end{aligned}$$

for $t \in \mathbb{R}_+$, $\theta \in \Theta$ and $N_c := |\delta(v_c)|$. The probability space corresponding to the rewritten network problem is $(\Theta, \sigma(\xi), \mathbb{P})$. Note that the equations are still stochastic and that the abstract definition of the expectation is not suitable for numerical computations.

Due to the finite-dimensional noise assumption, we can perform a change of variable as described in Subsection 3.1.1. Defining $\mathbf{y} := \xi(\theta) \in \Gamma$ leads to the purely deterministic, parametrized problem

$$\left.\begin{aligned} \partial_t u^{(j)}(x,t,\mathbf{y}) + \partial_x F_{m_j}\big(u^{(j)}(x,t,\mathbf{y})\big) &= G_{m_j}\big(u^{(j)}(x,t,\mathbf{y}),x,t\big), && m_j \in \mathcal{M}_{\mathrm{PDE}}, && (3.8)\\ \partial_x F_{m_j}\big(u^{(j)}(x,t,\mathbf{y})\big) &= G_{m_j}\big(u^{(j)}(x,t,\mathbf{y}),x,t\big), && m_j \in \mathcal{M}_{\mathrm{QS}}, && (3.9)\end{aligned}\right\} x \in \Omega_j,$$
$$u^{(j)}(x,0,\mathbf{y}) = u_0^{(j)}(x,\mathbf{y}), \qquad x \in \Omega_j, \qquad (3.10)$$

for $j = 1, \dots, P$, $t \in \mathbb{R}_+$ and $\mathbf{y} \in \Gamma$. The problem is complemented by boundary and coupling conditions

$$\begin{aligned} \mathcal{B}_b\big(u^{(j)}(v_b,t,\mathbf{y})\big) &= H_b(t,\mathbf{y}), && b \in \mathcal{I}_{BC},\ j \in \delta(v_b), && (3.11)\\ \mathcal{C}_c\big(u^{(j_1)}(v_c,t,\mathbf{y}),\dots,u^{(j_{N_c})}(v_c,t,\mathbf{y})\big) &= \Pi_c(t,\mathbf{y}), && c \in \mathcal{I}_0,\ j_k \in \delta(v_c),\\ & && k = 1,\dots,N_c, && (3.12)\end{aligned}$$

for $t \in \mathbb{R}_+$, $\mathbf{y} \in \Gamma$ and $N_c := |\delta(v_c)|$. In addition to space and time, this equation is also parametrized with the stochastic parameter $\mathbf{y} \in \Gamma$. Finally, we have transferred the problem formulated for an abstract stochastic sample space to the finite-dimensional image space Γ of the random vector ξ.

In this work, we restrict the setting to the common assumptions that the RVs ξ_i are independent and have bounded support Γ_i.

Assumption 3.5 (Boundedness)
The image space Γ_i of the RV ξ_i is a bounded interval in $\mathbb{R}$ for all $i = 1, \dots, N$. Without loss of generality, we assume $\Gamma_i = [-1, 1]$.

Assumption 3.6 (Stochastic Independence)
The RVs ξ_i with image space Γ_i are independent RVs and have the PDFs $\rho_i : \Gamma_i \to \mathbb{R}_+$ for $i = 1, \dots, N$.

Because of the previous assumption, the joint PDF $\rho : \Gamma \to \mathbb{R}_+$ of the random vector ξ is given by $\rho(\mathbf{y}) = \prod_{i=1}^{N} \rho_i(y_i)$. The problem including the assumptions is the basis for stochastic collocation methods. Finally, we define the solution of the parametrized problem (3.8)–(3.12) similar to Definition 3.2 and we assume the existence and uniqueness of this solution.

Definition 3.7 (Stochastic Entropy Network Solution)
The function $u = \big(u^{(1)}, \dots, u^{(P)}\big) : \Gamma \to C([0,T], L^1(\Omega_1, \mathbb{R}^d)) \times \cdots \times C([0,T], L^1(\Omega_P, \mathbb{R}^d))$ is a stochastic entropy network solution of the parametrized problem (3.8)–(3.12) if, for all $\mathbf{y} \in \Gamma$, $u(\mathbf{y})$ is an entropy network solution of the deterministic problem which results from inserting the fixed variable $\mathbf{y}$ into problem (3.8)–(3.12).

Assumption 3.8
There exists a unique stochastic entropy network solution u of problem (3.8)–(3.12) such that $u \in L^2_\rho(\Gamma, \mathbb{Y})$ with $\mathbb{Y} = C\big([0,T], L^1(\Omega_1, \mathbb{R}^d)\big) \times \cdots \times C\big([0,T], L^1(\Omega_P, \mathbb{R}^d)\big)$.

Chapter 4

Adaptive Multi-Level Monte Carlo Method

In this chapter, we present a fully adaptive multi-level Monte Carlo method. Monte Carlo methods are used to approximate the expected value of a quantity of interest depending on the solution of problem (3.3)–(3.7). Moreover, these methods are standard sampling methods and are very robust because no further regularity assumptions on the quantity of interest (QoI) are necessary. However, we make some assumptions on the QoI for the convergence and complexity analysis.

First, we describe the fully adaptive single-level Monte Carlo (SLMC) approach where the QoI is the solution of problem (3.3)–(3.7). The developed strategy is called fully adaptive because of its adaptive manner in the stochastic as well in the physical space. We combine an adaptive choice of the number of samples with adaptive computations of each sample computed for a given physical tolerance. Then, we extend the method by a multi-level (ML) structure and analyze its complexity. Both approaches are controlled by a prescribed total accuracy. Finally, the expected value of functionals of the solution are considered and the implementation of the method is explained.

A similar approach was presented for random elliptic problems in [64] where only Hilbert space-valued random variables were considered and all samples on the coarsest level were computed with the same initial discretization. Therefore, the number of levels usually increases with the prescribed total accuracy. In contrast to [64], we analyze the more general case of Banach space-valued random variables and choose a fixed number of levels independently from the total accuracy, see Section 4.4.

We recall that problem (3.3)–(3.7) in Section 3.2 is given by

$$\left.\begin{aligned} \partial_t u^{(j)}(x,t,\theta) + \partial_x F_{m_j}\big(u^{(j)}(x,t,\theta)\big) &= G_{m_j}\big(u^{(j)}(x,t,\theta),x,t\big), && m_j \in \mathcal{M}_{\mathrm{PDE}},\\ \partial_x F_{m_j}\big(u^{(j)}(x,t,\theta)\big) &= G_{m_j}\big(u^{(j)}(x,t,\theta),x,t\big), && m_j \in \mathcal{M}_{\mathrm{QS}},\end{aligned}\right\} x \in \Omega_j,$$
$$u^{(j)}(x,0,\theta) = u_0^{(j)}(x,\theta), \qquad x \in \Omega_j,$$

for $j = 1,\dots,P$, $t \in \mathbb{R}_+$ and $\theta \in \Theta$. The problem is complemented by boundary and coupling conditions

$$\begin{aligned} \mathcal{B}_b\big(u^{(j)}(v_b,t,\theta)\big) &= H_b(t,\theta), && b \in \mathcal{I}_{BC},\ j \in \delta(v_b),\\ \mathcal{C}_c\big(u^{(j_1)}(v_c,t,\theta),\dots,u^{(j_{N_c})}(v_c,t,\theta)\big) &= \Pi_c(t,\theta), && c \in \mathcal{I}_0,\ j_k \in \delta(v_c),\\ & && k = 1,\dots,N_c, \end{aligned}$$

for $t \in \mathbb{R}_+$, $\theta \in \Theta$ and $N_c := |\delta(v_c)|$. The random solution field $u = \big(u^{(1)},\dots,u^{(P)}\big)$ with $u^{(j)}(x,t,\theta) : D_j \times \Theta \to \mathbb{R}^d$ is defined on the underlying complete probability space $(\Theta,\Sigma,\mathbb{P})$.

4.1 Single-Level Structure

The idea of the Monte Carlo estimator is to compute the average of deterministic samples of the considered QoI (see e.g. [13] or [18]). We start with the general definition of the Monte Carlo estimator considering an arbitrary random variable.

Definition 4.1 (Monte Carlo Estimator)
Let $\mathcal{W}$ be a Banach space and let $v \in L^2(\Theta, \mathcal{W})$. The Monte Carlo (MC) estimator of the random variable v is defined as

$$E_M[v] = \frac{1}{M} \sum_{n=1}^{M} v_n$$

with $M \in \mathbb{N}_+$ and random variables $v_n \in L^2(\Theta, \mathcal{W})$ which are independent of each other and have the same probability distribution as v (i.i.d.).
A Monte Carlo estimate is a realization of the MC estimator denoted by

$$\widehat{E_M}[v] := E_M[v](\bar{\theta}) = \frac{1}{M} \sum_{n=1}^{M} v_n(\bar{\theta}) \in \mathcal{W}$$

for a given $\bar{\theta} \in \Theta$ and realizations $v_n(\bar{\theta})$ of v_n. The realizations $v_1(\bar{\theta}), \dots, v_M(\bar{\theta})$ are also called independent and identically distributed (i.i.d.) samples of the random variable v.
To enhance the readability, we use the more common notation: M i.i.d. samples are denoted by $v(\theta_1), \dots, v(\theta_M)$ with pairwise distinct $\theta_1, \dots, \theta_M \in \Theta$.

From the previous definition, we conclude directly that the MC estimator $E_M[v]$ is a mean-square integrable random variable and is unbiased:

$$E_M[v] \in L^2(\Theta, \mathcal{W}), \qquad \mathbb{E}\big[E_M[v]\big] = \mathbb{E}[v].$$

Since the random variables in Definition 4.1 are independent and identically distributed, the strong law of large numbers [54, 73] guarantees that the MC estimator $E_M[v]$ converges $\mathbb{P}$-almost surely to the mean $\mathbb{E}[v]$. In Definition 4.1, we strictly distinguish between estimator and estimate, because the estimator is a random variable and is analyzed in the theory. In contrast, the MC estimate is an element of $\mathcal{W}$ and is computed in the implementation.

Now, we consider the random entropy solution u of the stochastic problem (3.3)–(3.7). The expected value of u could be approximated by the MC estimator $E_M[u]$. However, the exact random entropy solution u and thus its samples are in general not available. Therefore, we consider an approximate random field u_h of the solution u and compute the MC estimator $E_M[u_h]$. Each realization $u_h(\theta)$ is a finite-dimensional physical approximation of the solution $u(\theta)$ of a deterministic network problem which has the same form as (2.11)–(2.15) and arises from inserting the fixed θ into (3.3)–(3.7). Since the MC estimate contains M i.i.d samples of u_h, we get M decoupled deterministic problems which differ in the input data. For the computation of $u_h(\theta_n)$, we usually solve the deterministic network problem by applying a numerical method (e.g. a finite volume method or an implicit box scheme as described in Section 2.4) to each edge of the network while taking the imposed boundary and coupling conditions into account. A very common approach is to predefine a uniform discretization in time and space, see for example [76]. Now, we assume that we have an adaptive solver which adaptively refines the spatial, temporal and model discretizations until the error estimate is less than a prescribed physical tolerance $TolH > 0$. Consequently, each realization $u_h(\theta)$ has a sample-adaptive resolution in the physical space, i.e. in space, time and model hierarchy. Therefore, the samples could have different

resolutions in time and space. In this case, we need to interpolate all samples onto a common fine spatial-temporal grid. The approximate random field u_h refers to a given tolerance $TolH$ and is also called random numerical solution. The general aspects of adaptive strategies were described in Subsection 2.4.2 and an adaptive algorithm developed for gas transport was presented in Subsection 2.5.4.

Assumption 4.2 (Adaptive Physical Approximation)

(i) Let u be the random entropy solution of problem (3.3)–(3.7). For $\theta \in \Theta$, let $u_h(\theta)$ be the physical approximation of $u(\theta)$ computed for a given tolerance $TolH > 0$ by an adaptive solver using a space-time-model discretization. Then, there exists a physical constant $c_h : \Theta \to \mathbb{R}_+$ with $\mathbb{E}[c_h] < \infty$ such that $u_h(\theta)$ satisfies

$$\|u(\theta) - u_h(\theta)\|_{\mathbb{L}^1} \leq c_h(\theta)\, TolH \tag{4.1}$$

$\mathbb{P}$-almost surely.

(ii) It holds $u_h \in L^2(\Theta, \mathbb{L}^1)$.

From this assumption, it follows

$$\mathbb{E}\big[\|u - u_h\|_{\mathbb{L}^1}\big] \leq C_h\, TolH \tag{4.2}$$

with $C_h := \mathbb{E}[c_h] = \int_\Theta c_h(\theta)\, \mathrm{d}\mathbb{P}(\theta)$ by integrating the inequality (4.1) over the sample space. Note that the constant C_h is independent of the stochastic parameter $\theta \in \Theta$.

In order to show that the approximation of the expected value of the random entropy solution u converges to the MC estimator of the approximate solution u_h, we investigate the overall error $\mathbb{E}[u] - E_M[u_h]$ in more detail. Using the triangle inequality, we obtain the splitting

$$\|\mathbb{E}[u] - E_M[u_h]\|_{L^2(\Theta,\mathbb{L}^1)} \leq \underbrace{\|\mathbb{E}[u] - \mathbb{E}[u_h]\|_{L^2(\Theta,\mathbb{L}^1)}}_{I)\ physical\ error} + \underbrace{\|\mathbb{E}[u_h] - E_M[u_h]\|_{L^2(\Theta,\mathbb{L}^1)}}_{II)\ stochastic\ error}.$$

The first term on the right-hand side of the above inequality represents the deterministic physical error. This error results from the approximation of the random entropy solution u by a random numerical solution u_h and depends on the chosen resolution in space, time and model hierarchy. The second term is the stochastic sampling error which we get by using the MC estimator to approximate the expected value. We can bound the error terms as stated in the following two Lemmata. Hence, we can formulate an estimate for the overall error, see Theorem 4.6.

Lemma 4.3

Suppose Assumptions 3.3 and 4.2 hold. Then, the deterministic physical error is bounded by

$$\|\mathbb{E}[u] - \mathbb{E}[u_h]\|_{L^2(\Theta,\mathbb{L}^1)} \leq C_h\, TolH$$

with constant $C_h = \mathbb{E}[c_h]$.

Proof. For Banach space-valued random variables, it holds (3.1):

$$\|\mathbb{E}[u] - \mathbb{E}[u_h]\|_{L^2(\Theta,\mathbb{L}^1)} = \|\mathbb{E}[u - u_h]\|_{\mathbb{L}^1} \leq \mathbb{E}[\|u - u_h\|_{\mathbb{L}^1}].$$

By Assumption 4.2, we can use inequality (4.2) and obtain

$$\|\mathbb{E}[u] - \mathbb{E}[u_h]\|_{L^2(\Theta,\mathbb{L}^1)} \leq \mathbb{E}[c_h]\, TolH. \qquad \square$$

The following analysis of the stochastic sampling error is based on [76, 89]. We observe that the physical space $\mathbb{L}^1$ will not lead directly to a convergence bound. From the Definition 4.1 of the MC estimator, it follows

$$\|\mathbb{E}[u_h] - E_M[u_h]\|^2_{L^2(\Theta,\mathbb{L}^1)} = \left\|\mathbb{E}[u_h] - \sum_{n=1}^{M} \frac{1}{M} u_{h,n}\right\|^2_{L^2(\Theta,\mathbb{L}^1)} = \frac{1}{M^2}\left\|\sum_{n=1}^{M} \left(\mathbb{E}[u_h] - u_{h,n}\right)\right\|^2_{L^2(\Theta,\mathbb{L}^1)},$$

where the random variables $u_{h,n}$ are independent of each other and have the same probability distribution as u_h. Then, using the triangle inequality and the property that the random variables $u_{h,n}$ are i.i.d., we obtain

$$\frac{1}{M^2}\left\|\sum_{n=1}^{M}(\mathbb{E}[u_h] - u_{h,n})\right\|^2_{L^2(\Theta,\mathbb{L}^1)} \leq \frac{1}{M^2}\left(\sum_{n=1}^{M}\|\mathbb{E}[u_h] - u_{h,n}\|_{L^2(\Theta,\mathbb{L}^1)}\right)^2 = \|\mathbb{E}[u_h] - u_h\|^2_{L^2(\Theta,\mathbb{L}^1)}.$$

Unfortunately, the resulting bound is independent of the number M of samples and cannot provide the convergence of the stochastic error for $M \to \infty$. However, in the case of a random variable $v_h \in L^2(\Theta, \mathcal{H})$ with a Hilbert space $\mathcal{H}$, the error satisfies the identity [13]

$$\|\mathbb{E}[v_h] - E_M[v_h]\|^2_{L^2(\Theta,\mathcal{H})} = \frac{1}{M}\|\mathbb{E}[v_h] - v_h\|^2_{L^2(\Theta,\mathcal{H})}. \tag{4.3}$$

This equation is mainly used in the MC theory of elliptic PDEs for the Hilbert spaces L^2 or H^1. The idea to guarantee the convergence of the stochastic error with respect to M is to bound the $L^2(\Theta,\mathbb{L}^1)$-norm by the $L^2(\Theta,\mathbb{L}^2)$-norm and to use the previous identity. Therefore, we make the following assumption on u_h.

Assumption 4.4
The approximate solution u_h satisfies

$$u_h \in L^2(\Theta, \mathbb{L}^\infty)$$

with $\mathbb{L}^\infty = L^\infty((0,T)\times\Omega_1,\mathbb{R}^d)\times\cdots\times L^\infty((0,T)\times\Omega_P,\mathbb{R}^d)$.

Lemma 4.5
Suppose Assumptions 4.2 (ii) and 4.4 hold. Then, the stochastic error satisfies the inequality

$$\|\mathbb{E}[u_h] - E_M[u_h]\|_{L^2(\Theta,\mathbb{L}^1)} \leq \sqrt{C_{T,\Omega}}\ \frac{1}{\sqrt{M}}\|\mathbb{E}[u_h] - u_h\|_{L^2(\Theta,\mathbb{L}^2)}$$

with $C_{T,\Omega} = d\,T\,\sum_{j=1}^{P}|\Omega_j|$.

Proof. Since Assumption 4.4 is fulfilled, it follows immediately

$$\mathbb{E}[u_h] \in \mathbb{L}^\infty \quad \text{and} \quad E_M[u_h] \in L^2(\Theta,\mathbb{L}^\infty).$$

Hence, it holds $\mathbb{E}[u_h] - E_M[u_h] \in L^2(\Theta,\mathbb{L}^\infty) \subset L^2(\Theta,\mathbb{L}^2)$, especially the error $\mathbb{E}[u_h] - E_M[u_h](\theta)$ is $\mathbb{P}$-almost surely an element of $\mathbb{L}^2$. Applying Lemma 2.4 $\mathbb{P}$-almost surely leads to

$$\begin{aligned}
\|\mathbb{E}[u_h] - E_M[u_h]\|^2_{L^2(\Theta,\mathbb{L}^1)} &= \int_\Theta \|\mathbb{E}[u_h] - E_M[u_h](\theta)\|^2_{\mathbb{L}^1}\,\mathrm{d}\mathbb{P}(\theta) \\
&\leq C_{T,\Omega}\int_\Theta \|\mathbb{E}[u_h] - E_M[u_h](\theta)\|^2_{\mathbb{L}^2}\,\mathrm{d}\mathbb{P}(\theta) \\
&= C_{T,\Omega}\,\|\mathbb{E}[u_h] - E_M[u_h]\|^2_{L^2(\Theta,\mathbb{L}^2)}.
\end{aligned}$$

Since $\mathbb{L}^2$ is a Hilbert space, we can apply the identity (4.3) and obtain

$$\|\mathbb{E}[u_h] - E_M[u_h]\|^2_{L^2(\Theta,\mathbb{L}^1)} \leq C_{T,\Omega}\,\frac{1}{M}\|\mathbb{E}[u_h] - u_h\|^2_{L^2(\Theta,\mathbb{L}^2)}. \qquad \square$$

Combining the Lemmata 4.3 and 4.5 gives us the following estimation of the overall error.

Theorem 4.6
Suppose Assumptions 3.3, 4.2 and 4.4 are fulfilled. Then, it holds

$$\|\mathbb{E}[u] - E_M[u_h]\|_{L^2(\Theta,\mathbb{L}^1)} \leq C_h\,TolH + \sqrt{C_{T,\Omega}}\frac{1}{\sqrt{M}}\|\mathbb{E}[u_h] - u_h\|_{L^2(\Theta,\mathbb{L}^2)}. \tag{4.4}$$

Note that the variance-like term $\|\mathbb{E}[u_h] - u_h\|_{L^2(\Theta,\mathbb{L}^2)}$ in the estimation of the MC error is finite, see (3.2).

In our convergence analysis, the goal is to guarantee that the overall error $\|\mathbb{E}[u]-E_M[u_h]\|_{L^2(\Theta,\mathbb{L}^1)}$ satisfies a user-prescribed accuracy $\epsilon > 0$. We balance the terms on the right-hand side of inequality (4.4) such that each term is equal to $\epsilon/2$. Consequently, the physical tolerance and the number of samples depend on ϵ and are given by

$$TolH = \frac{\epsilon}{2\,C_h} \quad \text{and} \quad M = \left\lceil \frac{4\,C_{T,\Omega}\,\|\mathbb{E}[u_h] - u_h\|^2_{L^2(\Theta,\mathbb{L}^2)}}{\epsilon^2} \right\rceil. \tag{4.5}$$

4.1.1 Complexity Analysis

The total computational cost C^{MC} of the single-level MC method is the sum of the single costs to compute each sample $u_h(\theta_n)$ of u_h for tolerance $TolH$. For a small accuracy ϵ, the computational effort of the MC method is usually very high since we need to strongly refine the physical discretization of each sample while significantly increasing the number of samples, see (4.5). These two processes lead to higher costs. Let W_h be an upper bound for the cost of each sample of u_h. Then, we get

$$C^{\mathrm{MC}} := \sum_{n=1}^{M} cost\big(u_h(\theta_n)\big) \leq M\,W_h,$$

where $cost\big(u_h(\theta_n)\big)$ denotes the cost to compute the n-th sample $u_h(\theta_n)$. Precisely, the quantity C^{MC} is the cost of a fixed realization of the MC estimator and therefore depends on different $\theta_1, \ldots, \theta_M \in \Theta$. The right side of the inequality is independent of the considered realization and thus the following analysis is valid for every realization of the MC estimator and especially for the expected value of the total computational cost. If only the expected cost is interesting, then it is sufficient to have an upper bound W_h for the expected cost $\mathbb{E}[cost(u_h)]$.

In order to link the computational complexity with the total accuracy ϵ, we require the following assumption.

Assumption 4.7
Let W_h be an upper bound for the cost to compute an MC sample of the approximate solution u_h for tolerance $TolH > 0$. Then, there exists a constant $\gamma > 0$ such that

$$W_h \leq C_W\,TolH^{-\gamma}$$

with constant C_W independent of γ and $TolH$.

Using the previous theorem for the overall error of the MC method, we can estimate the computational cost C_ϵ^{MC}, also called complexity, needed to achieve the accuracy ϵ for the MC estimator. For the complexity analysis, we introduce the notation $a \lesssim b$ to denote the relation $a \leq Cb$ with a constant C independent of the physical tolerance $TolH$, the number of samples M and the accuracy ϵ. In the case that $a \lesssim b$ and $a \gtrsim b$, we write $a \eqsim b$.

Lemma 4.8
Let Assumptions 3.3, 4.2, 4.4 and 4.7 be fulfilled. Then, for any $\epsilon < 1$, there exists a physical tolerance $TolH > 0$ and $M \in \mathbb{N}_+$ such that

$$\|\mathbb{E}[u] - E_M[u_h]\|_{L^2(\Theta,\mathbb{L}^1)} \leq \epsilon$$

and

$$C_\epsilon^{\mathrm{MC}} \lesssim \epsilon^{-2-\gamma}.$$

Proof. As stated in (4.5), the error satisfies the given accuracy ϵ if we choose

$$TolH = \frac{\epsilon}{2\,C_h} \quad \text{and} \quad M = \left\lceil 4\,C_{T,\Omega}\,\epsilon^{-2}\,\|\mathbb{E}[u_h] - u_h\|_{\mathbb{L}^2}^2 \right\rceil.$$

Inserting the formula of the number of samples and Assumption 4.7 provide the inequality

$$C_\epsilon^{\mathrm{MC}} \leq M\,W_h \leq \left(4\,C_{T,\Omega}\,\epsilon^{-2}\,\|\mathbb{E}[u_h] - u_h\|_{\mathbb{L}^2}^2 + 1\right)\,C_W\,(2\,C_h)^\gamma \epsilon^{-\gamma} \lesssim \epsilon^{-2-\gamma}$$

because $\epsilon < 1$ and $\gamma > 0$. □

Remark 4.9
Instead of the full space-time solution u, we can also choose a possibly nonlinear function of the solution as our QoI. Some possible options are a single component of the solution, the solution integrated over the time or the spatial space and the solution evaluated at a fixed point in time $\bar{t} \in [0,T]$ or at a fixed spatial point $\bar{x} \in \bar{\Omega}_j$.
For the error of the i-th components u_i, all previous results are valid without additional assumptions because it holds

$$\|\mathbb{E}[u_i] - E_M[(u_h)_i](\theta)\|_{L^1((0,T)\times\Omega_1)\times\cdots\times L^1((0,T)\times\Omega_P)} \leq \|\mathbb{E}[u] - E_M[u_h](\theta)\|_{\mathbb{L}^1}$$

$\mathbb{P}$-almost surely. The same argument applies to the quantities $\sum_{j=1}^P \int_{[0,T]} u^{(j)}(\cdot,t,\cdot)\,\mathrm{d}t$ and $\int_{\Omega_j} u^{(j)}(x,\cdot,\cdot)\,\mathrm{d}x$ using Fubini's theorem and the linearity of the MC estimator.
Analyzing the MC error of $u^{(j)}(x,\bar{t},\theta)$ for a fixed $\bar{t} \in [0,T]$ requires that the pointwise evaluation in time of the approximate solution is defined $\mathbb{P}$-almost surely. Therefore, we assume

$$u_h \in L^2\big(\Theta, L^1(\Omega_1, H^1((0,T),\mathbb{R}^d)) \times \cdots \times L^1(\Omega_P, H^1((0,T),\mathbb{R}^d))\big).$$

Assumptions 4.2, 4.4 and 4.7 have to hold for the considered quantity $u_h^{(j)}(x,\bar{t},\theta)$. Then, the convergence and complexity analysis of the error

$$\|\mathbb{E}[u^{(j)}(\cdot,\bar{t},\cdot)] - E_M[u_h^{(j)}(\cdot,\bar{t},\cdot)]\|_{L^2(\Theta,L^1(\Omega_j,\mathbb{R}^d))}$$

can be performed analogously to this section. The constant $C_{T,\Omega}$ in Lemma 4.5 will change to $C_\Omega = d\,|\Omega_j|$.
In the case of the approximate solution $u_h^{(j)}(\bar{x},t,\theta)$ evaluated at a fixed spatial point $x \in \bar{\Omega}_j$, we need to assume that u and u_h belong to the space

$$L^2\big(\Theta, L^1((0,T), H^1(\Omega_1,\mathbb{R}^d)) \times \cdots \times L^1((0,T), H^1(\Omega_P,\mathbb{R}^d))\big).$$

If Assumptions 4.2, 4.4 and 4.7 are fulfilled for $u_h^{(j)}(\bar{x},t,\theta)$*, we can analyze the error*

$$\|\mathbb{E}[u^{(j)}(\bar{x},\cdot,\cdot)] - E_M[u_h^{(j)}(\bar{x},\cdot,\cdot)]\|_{L^2(\Theta,L^1((0,T),\mathbb{R}^d))}$$

as shown in this section. In the stochastic error estimate in Lemma 4.5, we will get the constant $C_T = d\,T$ *instead of* $C_{T,\Omega}$*.*

Remark 4.10
Following the previous analysis, we get the same results for the mean squared error (MSE) $\|\mathbb{E}[u] - E_M[u_h]\|^2_{L^2(\Theta,\mathbb{L}^1)}$*. If we consider the* $\mathbb{L}^2$*-norm instead of the* $\mathbb{L}^1$*-norm, then the MSE is usually considered. In the error analysis [13], equality now holds in the splitting inequality of the MSE into the squared physical and the squared sampling error:*

$$\|\mathbb{E}[u] - E_M[u_h]\|^2_{L^2(\Theta,\mathbb{L}^2)} = \|\mathbb{E}[u] - \mathbb{E}[u_h]\|^2_{L^2(\Theta,\mathbb{L}^2)} + \frac{1}{M}\|\mathbb{E}[u_h] - u_h\|^2_{L^2(\Theta,\mathbb{L}^2)}.$$

In addition, the constant $C_{T,\Omega}$ *in Lemma 4.5 vanishes because identity* (4.3) *holds. Finally, we obtain the physical tolerance* $\sqrt{2}\,TolH$ *and* $M/(2\,C_{T,\Omega})$ *as the number of samples, where* $TolH$ *and* M *are defined by* (4.5)*. The estimation of the complexity remains the same.*

4.2 Multi-Level Structure

The Monte Carlo method is now extended by a multi-level structure in order to reduce the computational effort. At first, Heinrich [53] and Giles [42] independently developed a multi-level strategy for MC methods which was then applied to PDEs with random data by Barth et al. [10], Charrier et al. [15] and Cliffe et al. [18]. The basic idea is to couple approximate solutions of different physical accuracies such that approximate solutions with lower physical accuracies are sampled more often whereas the ones with higher accuracies are sampled less often. Typically, the computational cost for computing one sample increases with the accuracy of the sample.

Let $\{TolH_k\}_{k=0}^K$ be a sequence of physical tolerances with

$$1 \geq TolH_0 > TolH_1 > \cdots > TolH_K > 0$$

and $K \in \mathbb{N}_+$. Note that we would obtain the single-level (SL) approach for $K = 0$. Each integer $k \in \{0,\ldots,K\}$ refers to an approximate random field $u_{h_k} : \Theta \to \mathbb{L}^1$. Each realization $u_{h_k}(\theta)$ is computed with a sample-dependent space-time-model resolution which is controlled by $TolH_k$ via an adaptive strategy and gets finer with increasing k and fixed θ. We formulate Assumptions 4.2 and 4.4 for the sequence of these approximate solutions.

Assumption 4.11 (Sequence of Adaptive Physical Approximations)

(i) Let u *be the random entropy solution of problem* (3.3)–(3.7) *and* $u_{h_k}(\theta)$ *be the numerical approximation of* $u(\theta)$ *with a space-time-model discretization provided by an adaptive solver for a given* $TolH_k > 0$ *and* $\theta \in \Theta$*. Then, for all* $k = 0,\ldots,K$ *there exists a physical constant* $c_k : \Theta \to \mathbb{R}_+$ *with* $\mathbb{E}[c_k] < \infty$ *such that the approximate solution* $u_{h_k}(\theta)$ *satisfies*

$$\|u(\theta) - u_{h_k}(\theta)\|_{\mathbb{L}^1} \leq c_k(\theta)\,TolH_k \tag{4.6}$$

$\mathbb{P}$*-almost surely.*

(ii) It holds $u_{h_k} \in L^2(\Theta,\mathbb{L}^\infty)$ *for* $k = 0,\ldots,K$*.*

We can formulate the approximate solution u_{h_K} with the highest accuracy as the telescoping sum of approximations with lower physical accuracies

$$u_{h_K} = \sum_{k=0}^{K} u_{h_k} - u_{h_{k-1}}$$

with $u_{-1} := 0$. Considering the expected value of u_{h_K}, we obtain the following simplification using the previous equation and the linearity of the mean:

$$\mathbb{E}[u_{h_K}] = \mathbb{E}\left[\sum_{k=0}^{K}(u_{h_k} - u_{h_{k-1}})\right] = \sum_{k=0}^{K}\mathbb{E}[u_{h_k} - u_{h_{k-1}}].$$

Now, we estimate each expected value of the difference by a Monte Carlo estimator introduced in Definition 4.1. This provides the following definition of the multi-level Monte Carlo estimator.

Definition 4.12 (Multi-Level Monte Carlo Estimator)
Let $u_{h_k} \in L^2(\Theta, \mathbb{L}^1)$ be approximate solutions and $u_{h_{-1}} = 0$. The multi-level Monte Carlo (MLMC) estimator is defined as

$$E^K\left[u_{h_K}\right] = \sum_{k=0}^{K} E_{M_k}[u_{h_k} - u_{h_{k-1}}]$$

with $M_k \in \mathbb{N}_+$. The sequence $k = 0, \dots, K$ referring to M_k as well as to the corresponding MC estimator E_{M_k} of the difference $u_{h_k} - u_{h_{k-1}}$ is called levels. The number of levels is given by $K + 1$.
A multi-level Monte Carlo estimate is a realization of the MLMC estimator denoted by

$$\widehat{E^K}[u_{h_K}] := E^K\left[u_{h_K}\right](\bar{\theta}) = \sum_{k=0}^{K} E_{M_k}[u_{h_k} - u_{h_{k-1}}](\bar{\theta}) = \sum_{k=0}^{K}\frac{1}{M_k}\sum_{n=1}^{M_k} d_{k,n}(\bar{\theta}) \in \mathbb{L}^1,$$

where $\bar{\theta} \in \Theta$ is given and $d_{k,1}, \dots, d_{k,M_k}$ are i.i.d. random variables of $d_k := u_{h_k} - u_{h_{k-1}}$ for each level k. In order to enhance the readability as in Definition 4.1, we denote the n-th sample $d_{k,n}(\bar{\theta})$ by $d_k(\theta_n^k)$. Using this common notation, the MLMC estimate has the form

$$\widehat{E^K}[u_{h_K}] = \sum_{k=0}^{K}\frac{1}{M_k}\sum_{n=1}^{M_k} d_k(\theta_n^k) = \sum_{k=0}^{K}\frac{1}{M_k}\sum_{n=1}^{M_k} u_{h_k}(\theta_n^k) - u_{h_{k-1}}(\theta_n^k)$$

with different $\theta_n^k \in \Theta$.

Note that the MC estimators E_{M_k} used in the previous definition are pairwise independent. This means that the random variables $d_{k,i}$ on level $k \in \{0, \dots, K\}$ and $d_{k+1,j}$ on level $k+1$ are independent for all $i = 1, \dots, M_k$ and $j = 1, \dots, M_{k+1}$. Therefore, the samples used on level k are different from those on the other levels. More precisely, each sample computed for the MLMC estimate corresponds to a different, unique deterministic network problem (2.11)–(2.15). However, on level $k > 0$, the sample $d_k(\theta_i^k) = u_{h_k}(\theta_i^k) - u_{h_{k-1}}(\theta_i^k)$ is the difference between two approximate solutions of the same deterministic problem computed for two different physical tolerances, namely $TolH_k$ and $TolH_{k-1}$. Since the samples of all levels do not have the same discretization in time and space, we need to interpolate them onto an underlying fine spatial-temporal grid. Similar to the single-level approach, it follows directly from the definition that

the multi-level Monte Carlo estimator $E^K[u_{h_K}]$ is a mean-square integrable random variable and unbiased:

$$E^K[u_{h_K}] \in L^2(\Theta, \mathbb{L}^1), \quad \mathbb{E}\left[E^K[u_{h_K}]\right] = \mathbb{E}[u_{h_K}].$$

For the convergence analysis, we consider the error between the MLMC estimator $E^K[u_{h_K}]$ and the expected value of the random entropy solution. As in the single-level MC approach, we split the error into a deterministic physical and a stochastic sampling error:

$$\|\mathbb{E}[u] - E^K[u_{h_K}]\|_{L^2(\Theta,\mathbb{L}^1)} \leq \underbrace{\|\mathbb{E}[u] - \mathbb{E}[u_{h_K}]\|_{L^2(\Theta,\mathbb{L}^1)}}_{I)\ physical\ error} + \underbrace{\|\mathbb{E}[u_{h_K}] - E^K[u_{h_K}]\|_{L^2(\Theta,\mathbb{L}^1)}}_{II)\ stochastic\ error}. \tag{4.7}$$

The physical and the sampling error can be estimated further and we obtain the following estimation on the overall MLMC error.

Theorem 4.13

Let Assumptions 3.3 and 4.11 be fulfilled. Then, the error of the MLMC estimator can be bounded by

$$\|\mathbb{E}[u] - E^K[u_{h_K}]\|_{L^2(\Theta,\mathbb{L}^1)} \leq C_H\, TolH_K + \sqrt{C_{T,\Omega} \sum_{k=0}^{K} \frac{1}{M_k} \|\mathbb{E}[d_k] - d_k\|^2_{L^2(\Theta,\mathbb{L}^2)}}$$

with $d_k = u_{h_k} - u_{h_{k-1}}$.

Proof. We start to consider the first term on the right-hand side of (4.7), the deterministic physical error. Integrating the assumed estimation (4.6) over the sample space Θ leads to

$$\mathbb{E}[\|u - u_{h_k}\|_{\mathbb{L}^1}] \leq C_H\, TolH_k \quad \text{for } k = 0, \dots, K$$

with the constant $C_H := \max_{k=0,\dots,K} \mathbb{E}[c_k]$ which is independent of θ. Using the same arguments as in the proof of Lemma 4.3, we obtain

$$\|\mathbb{E}[u] - \mathbb{E}[u_{h_K}]\|_{L^2(\Theta,\mathbb{L}^1)} \leq \mathbb{E}\big[\|u - u_{h_K}\|_{\mathbb{L}^1}\big] \leq C_H\, TolH_K.$$

Next, the stochastic sampling error is estimated. Due to Assumption 4.11 (ii), it holds $\mathbb{P}$-almost surely that $\mathbb{E}[u_{h_K}] - E^K[u_{h_K}](\theta) \in \mathbb{L}^\infty \subset \mathbb{L}^2$. Therefore, Lemma 2.4 can be applied $\mathbb{P}$-almost surely:

$$\begin{aligned}
\|\mathbb{E}[u_{h_K}] - E^K[u_{h_K}]\|^2_{L^2(\Theta,\mathbb{L}^1)} &= \int_\Theta \|\mathbb{E}[u_{h_K}] - E^K[u_{h_K}](\theta)\|^2_{\mathbb{L}^1}\, \mathrm{d}\mathbb{P}(\theta) \\
&\leq C_{T,\Omega} \int_\Theta \|\mathbb{E}[u_{h_K}] - E^K[u_{h_K}](\theta)\|^2_{\mathbb{L}^2}\, \mathrm{d}\mathbb{P}(\theta) \\
&= C_{T,\Omega}\, \|\mathbb{E}[u_{h_K}] - E^K[u_{h_K}]\|^2_{L^2(\Theta,\mathbb{L}^2)}.
\end{aligned}$$

Then, it holds [13, Theorem 3.1]

$$\|\mathbb{E}[u_{h_K}] - E^K[u_{h_K}]\|^2_{L^2(\Theta,\mathbb{L}^2)} = \sum_{k=0}^{K} \frac{1}{M_k} \|\mathbb{E}[u_{h_k} - u_{h_{k-1}}] - (u_{h_k} - u_{h_{k-1}})\|^2_{L^2(\Theta,\mathbb{L}^2)}$$

because $\mathbb{L}^2$ is a Hilbert space and the random variables $(u_{h_k} - u_{h_{k-1}})_{n_k}$ in Definition 4.12 are i.i.d. for all $k = 0, \dots, K$ and $n_k = 1, \dots, M_k$. □

As in the single-level context, the variance-like term

$$V[u_{h_k} - u_{h_{k-1}}] := \|\mathbb{E}[u_{h_k} - u_{h_{k-1}}] - (u_{h_k} - u_{h_{k-1}})\|^2_{L^2(\Theta,\mathbb{L}^2)}$$

is finite, since $u_{h_k} - u_{h_{k-1}} \in L^2(\Theta, \mathbb{L}^2)$ and therefore (3.2) holds.

One approach to distribute the samples over the levels is to choose the number of samples M_k depending on the physical tolerance $TolH_K$ such that the stochastic error has the same size as the physical error. This strategy was used in [10] to obtain the same order of the error parts and in [67, 106] for an ML stochastic collocation method. Due to the previous theorem, we balance the two terms on the right-hand side by choosing

$$M_k \leq C_{T,\Omega}\, V[u_{h_k} - u_{h_{k-1}}](K+1)(C_H\, TolH_K)^{-2}.$$

It follows $\|\mathbb{E}[u] - E^K[u_{h_K}]\|_{L^2(\Theta,\mathbb{L}^2)} \leq 2\, C_H\, TolH_K$ so that the MLMC error converges for $TolH_K \to 0$. Let $\epsilon > 0$ be a user-prescribed accuracy for the MLMC error. In order to achieve this accuracy, both terms on the right-hand side of the estimate in Theorem 4.13 have to be bounded by $\epsilon/2$. For the considered approach, we obtain the physical tolerance $TolH_K = \epsilon/(2\, C_H)$ with the number of samples M_k defined as above.

Another approach is to choose the number M_k of samples on each level so that the computational cost of the method is minimized and the MLMC error satisfies the desired accuracy ϵ [18, 106]. The resulting complexity theorems are often formulated for a different setting where only uniform spatial refinements are considered instead of an adaptive and error-controlled algorithm. Therefore, a starting resolution for all samples of u_{h_0} has to be chosen beforehand, see for example [13, 18, 106]. This approach is adapted to our fully adaptive method and is shown in the following part.

4.2.1 Complexity Analysis

Let W_k be an upper bound for the cost of a single sample of u_{h_k} for $k = 0, \ldots, K$. The total computational cost of the multi-level Monte Carlo estimator is given by

$$C^{\mathrm{MLMC}} = \sum_{k=0}^{K} \sum_{n=1}^{M_k} cost\big(u_{h_k}(\theta_n^k)\big) + cost\big(u_{h_{k-1}}(\theta_n^k)\big) \leq \sum_{k=0}^{K} M_k\,(W_k + W_{k-1}) \tag{4.8}$$

with $W_{-1} := 0$ and different $\theta_n^k \in \Theta$.

We want to choose the number of samples on each level optimally in such a way that the total cost $C_\epsilon^{\mathrm{MLMC}}$ needed to achieve the accuracy ϵ for the MLMC estimator is minimized. For the following analysis, the physical tolerance $TolH_k$ needs to converge to zero for $k \to \infty$.

Theorem 4.14
Let $\{TolH_k\}_{k=0,1,\ldots}$ be a strictly decreasing null sequence of physical tolerances $TolH_k \in (0, 1]$. Further, let Assumptions 3.3 and 4.11 be fulfilled. Then, for any $\epsilon > 0$, there exists a number $K = K(\epsilon) \in \mathbb{N}_+$ and a sequence of number of samples $\{M_k\}_{k=0,\ldots,K}$ such that

$$\|\mathbb{E}[u] - E^K[u_{h_K}]\|_{L^2(\Theta,\mathbb{L}^1)} \leq \epsilon$$

and

$$C_\epsilon^{\mathrm{MLMC}} \leq 4\, C_{T,\Omega}\, \epsilon^{-2} \Big(\sum_{k=0}^{K} (W_k + W_{k-1})^{1/2}\, V[u_{h_k} - u_{h_{k-1}}]^{1/2} \Big)^2 + \sum_{k=0}^{K} (W_k + W_{k-1}).$$

The optimal choice for the number of samples M_k is given by

$$M_k = \left\lceil \frac{4}{\epsilon^2} C_{T,\Omega} \left(\frac{V[u_{h_k} - u_{h_{k-1}}]}{W_k + W_{k-1}} \right)^{1/2} \sum_{j=0}^{K} (W_j + W_{j-1})^{1/2}\; V[u_{h_j} - u_{h_{j-1}}]^{1/2} \right\rceil.$$

Proof. Considering the ML error, the above Theorem 4.13 provides the estimation

$$\| \mathbb{E}[u] - E^K\,[u_{h_K}] \|_{L^2(\Theta, \mathbb{L}^1)} \leq C_H\, TolH_K + \sqrt{C_{T,\Omega} \sum_{k=0}^{K} \frac{1}{M_k} V[u_{h_k} - u_{h_{k-1}}]}$$

with

$$V[u_{h_k} - u_{h_{k-1}}] = \| \mathbb{E}[u_{h_k} - u_{h_{k-1}}] - (u_{h_k} - u_{h_{k-1}}) \|^2_{L^2(\Theta, \mathbb{L}^2)}.$$

In order to achieve an accuracy $\epsilon > 0$, we first choose an appropriate $K \in \mathbb{N}_+$ such that $C_H\, TolH_K \leq \epsilon/2$. This choice determines the number $K = K(\epsilon)$ as a function of ϵ and can always be made since $\{TolH_k\}_{k=0,1,\dots}$ is a null sequence.

Now, we want to minimize the overall costs $C_\epsilon^{\mathrm{MLMC}}$ needed to achieve the accuracy ϵ. Due to our choice of K, we impose the equation

$$\sqrt{C_{T,\Omega} \sum_{k=0}^{K} \frac{1}{M_k} V[u_{h_k} - u_{h_{k-1}}]} = \epsilon/2$$

as a constraint. Together with the upper bound (4.8) for the total cost of the MLMC estimator, this leads to the minimization problem

$$\begin{aligned} \min_{M_0,\dots,M_K} \quad & \sum_{k=0}^{K} M_k\, (W_k + W_{k-1}) \\ \text{s.t.} \quad & C_{T,\Omega} \sum_{k=0}^{K} \frac{1}{M_k} V[u_{h_k} - u_{h_{k-1}}] = \frac{\epsilon^2}{4}, \end{aligned}$$

which we solve by the Lagrangian multiplier method as in [106]. Note that M_k is treated as a continuous variable. The Lagrange function is given by

$$\mathcal{L}(M_0, \dots, M_K, \alpha) = \sum_{k=0}^{K} M_k\, (W_k + W_{k-1}) + \alpha \left(C_{T,\Omega} \sum_{j=0}^{K} \frac{1}{M_j} V[u_{h_j} - u_{h_{j-1}}] - \frac{\epsilon^2}{4} \right)$$

with the Lagrange multiplier $\alpha \in \mathbb{R}$. In order to find an extremum, we impose the condition

$\nabla\mathcal{L}(M_0,\dots,M_K,\alpha) \stackrel{!}{=} 0$ which implies the equations

$$\frac{\partial\mathcal{L}}{\partial M_k} = W_k + W_{k-1} - \alpha C_{T,\Omega} M_k^{-2}\, V[u_{h_k} - u_{h_{k-1}}] = 0, \quad k = 0,\dots,K, \tag{4.9}$$

$$\frac{\partial\mathcal{L}}{\partial \alpha} = C_{T,\Omega}\sum_{j=0}^{K}\frac{1}{M_j}V[u_{h_j} - u_{h_{j-1}}] - \frac{\epsilon^2}{4} = 0. \tag{4.10}$$

First, we solve the equations (4.9) for M_k:

$$M_k = \left(\alpha\, C_{T,\Omega}\, V[u_{h_k} - u_{h_{k-1}}]\,(W_k + W_{k-1})^{-1}\right)^{1/2}, \quad k = 0,\dots,K. \tag{4.11}$$

Inserting this relation into (4.10) and solving the resulting equation for $\sqrt{\alpha}$ lead to

$$\alpha^{1/2} = \frac{4}{\epsilon^2}\sqrt{C_{T,\Omega}}\sum_{j=0}^{K} V[u_{h_j} - u_{h_{j-1}}]^{1/2}(W_j + W_{j-1})^{1/2}.$$

Then, substituting this formula for the parameter α into equation (4.11) results in the optimal number of samples

$$M_k = \frac{4}{\epsilon^2}C_{T,\Omega}\left(\frac{V[u_{h_k} - u_{h_{k-1}}]}{W_k + W_{k-1}}\right)^{1/2}\sum_{j=0}^{K}(W_j + W_{j-1})^{1/2}\, V[u_{h_j} - u_{h_{j-1}}]^{1/2},$$

which we round up to the next integer $M_k^* = \lceil M_k \rceil$. Using the chosen number of samples M_k^* satisfying $M_k \le M_k^* \le M_k + 1$, we obtain for the ϵ-cost the estimation

$$\begin{aligned} C_\epsilon^{\mathrm{MLMC}} &\le \sum_{k=0}^{K}(M_k + 1)(W_k + W_{k-1}) \\ &= 4\,\epsilon^{-2}\, C_{T,\Omega}\left(\sum_{k=0}^{K}(W_k + W_{k-1})^{1/2}\, V[u_{h_k} - u_{h_{k-1}}]^{1/2}\right)^2 + \sum_{k=0}^{K}(W_k + W_{k-1}). \end{aligned} \tag{4.12}$$

Note that the rounded number of samples M_k^* ensures that the stochastic error is bounded by $\epsilon/2$ since

$$\sqrt{C_{T,\Omega}\sum_{k=0}^{K}\frac{1}{M_k^*}V[u_{h_k} - u_{h_{k-1}}]} \le \sqrt{C_{T,\Omega}\sum_{k=0}^{K}\frac{1}{M_k}V[u_{h_k} - u_{h_{k-1}}]} = \frac{\epsilon}{2}. \qquad \square$$

Next, we assume that the upper bound W_k is bounded in the same way as in Assumption 4.7 in the single-level approach. This additional condition provides a slightly modified formula for the number of samples on each level which we use in the implementation of the MLMC method in Section 4.4.

Assumption 4.15
Let W_k be an upper bound for the cost to compute an MC sample of the approximate solution u_{h_k} for tolerance $TolH_k > 0$. There exists a constant $\gamma > 0$ with

$$W_k \le C_W\, TolH_k^{-\gamma}$$

for all $k = 0,\dots,K$. The constant $C_W > 0$ is independent of k, θ and γ.

Corollary 4.16
Let $\{TolH_k\}_{k=0,1,\dots}$ be a strictly decreasing null sequence of physical tolerances $TolH_k \in (0,1]$. Further, suppose Assumptions 3.3, 4.11 and 4.15 hold. Then, for any $\epsilon > 0$, there exists a number $K = K(\epsilon) \in \mathbb{N}_+$ and a sequence of number of samples $\{M_k\}_{k=0,\dots,K}$ such that

$$\|\mathbb{E}[u] - E^K[u_{h_K}]\|_{L^2(\Theta,\mathbb{L}^1)} \leq \epsilon$$

and

$$C_\epsilon^{\mathrm{MLMC}} \leq C_1 \epsilon^{-2} \left(\sum_{k=0}^{K} G_k(\gamma)^{1/2} V[u_{h_k} - u_{h_{k-1}}]^{1/2} \right)^2 + C_W \sum_{k=0}^{K} G_k(\gamma)$$

with $C_1 := 4\, C_{T,\Omega}\, C_W$ and

$$G_k(\gamma) := TolH_k^{-\gamma} + (1 - \delta_{k0})\, TolH_{k-1}^{-\gamma}. \tag{4.13}$$

As usual, the Kronecker delta δ_{k0} is defined as $\delta_{k0} = 1$ for $k = 0$ and $\delta_{k0} = 0$ otherwise. The optimal choice for the number M_k of samples is given by

$$M_k = \left\lceil 4\, \epsilon^{-2}\, C_{T,\Omega}\, V[u_{h_k} - u_{h_{k-1}}]^{1/2}\, G_k(\gamma)^{-1/2} \sum_{j=0}^{K} G_j(\gamma)^{1/2}\, V[u_{h_j} - u_{h_{j-1}}]^{1/2} \right\rceil, \quad k = 0, \dots, K. \tag{4.14}$$

Proof. We prove the stated results analogously to the proof of Theorem 4.14. Due to Assumption 4.15, the considered optimization problem modifies to

$$\begin{aligned} \min_{M_0,\dots,M_K} \quad & C_W \sum_{k=0}^{K} M_k \left(TolH_k^{-\gamma} + (1 - \delta_{k0})\, TolH_{k-1}^{-\gamma} \right) \\ \text{s.t.} \quad & C_{T,\Omega} \sum_{k=0}^{K} \frac{1}{M_k} V[u_{h_k} - u_{h_{k-1}}] = \frac{\epsilon^2}{4}. \end{aligned}$$

□

Here, we use the Kronecker delta because on the coarsest level $k = 0$ there are only samples of u_{h_0} and the overall cost on this level is bounded by $C_W\, M_0\, TolH_0^{-\gamma}$. Alternatively, we could set $TolH_{-1} = 0$ analogously to the cost bound $W_{-1} = 0$ in (4.8), however we define $TolH_{-1}$ in Chapter 5 differently and we want to keep the notation consistent in this work.

In the following, we concentrate on physical tolerances characterized by a geometric design such that $TolH_k = q^k\, TolH_0$ with a positive reduction factor $q < 1$. Furthermore, we suppose that the variance-like term $V[u_{h_k} - u_{h_{k-1}}]$ decrease with level k and then we prove the improvement in the asymptotic cost over the single-level approach.

Assumption 4.17 (Variance Reduction)
There exits a constant $\beta > 0$ with

$$V[u_{h_k} - u_{h_{k-1}}] \leq C_V\, TolH_k^{\beta} \quad \text{for } k > 0,$$

where C_V is a positive constant independent of k, θ and β.

Theorem 4.18
Let a null sequence of physical tolerances $\{TolH_k\}_{k=0,1,\dots}$ be given by $TolH_k = q^k\, TolH_0$ with $TolH_0 \in (0,1]$ and a reduction factor $q \in (0,1)$. Further, let Assumptions 3.3, 4.11, 4.15 and 4.17 be fulfilled. Suppose that the rates γ of Assumption 4.15 and β of Assumption 4.17 satisfy the condition $2 \geq \min(\gamma, \beta)$. Then, for any $\epsilon \in (0, 1/e]$, there exists a number $K = K(\epsilon) \in \mathbb{N}_+$ and a sequence $\{M_k\}_{k=0,\dots,K}$ such that

$$\|\mathbb{E}[u] - E^K\left[u_{h_K}\right]\|_{L^2(\Theta,\mathbb{L}^1)} \leq \epsilon$$

and

$$C_\epsilon^{\mathrm{MLMC}} \lesssim \begin{cases} \epsilon^{-2} & if\ \beta > \gamma, \\ \epsilon^{-2}(\log \epsilon)^2 & if\ \beta = \gamma, \\ \epsilon^{-2-(\gamma-\beta)} & if\ \beta < \gamma. \end{cases}$$

Proof. Due to the supposed assumptions, the first statement follows from Corollary 4.16. For the choice of the value K, we get the condition $C_H\, TolH_K = C_H\, q^K TolH_0 \leq \epsilon/2$. We choose

$$K := \left\lceil \log_q \left(\frac{\epsilon}{2\, C_H\, TolH_0} \right) \right\rceil + K_0 \tag{4.15}$$

with the smallest possible constant $K_0 \in \mathbb{N}$ such that $K \in \mathbb{N}_+$. It follows

$$C_H\, TolH_K \leq q^{K_0+1}\epsilon/2 \leq \epsilon/2$$

since $q^{K_0+1} < 1$ and $K \leq \log_q(\epsilon/(2\, C_H\, TolH_0)) + 1 + K_0$. In addition, Corollary 4.16 provides

$$C_\epsilon^{\mathrm{MLMC}} \leq \underbrace{4\, C_{T,\Omega}\, C_W\, \epsilon^{-2}\, F(\gamma)^2}_{=:\ (I)} + \underbrace{C_W \sum_{k=0}^{K} G_k(\gamma)}_{=:\ (II)} \tag{4.16}$$

with

$$F(\gamma) := \sum_{k=0}^{K} G_k(\gamma)^{1/2}\, V[u_{h_k} - u_{h_{k-1}}]^{1/2} \quad \text{and} \quad G_k(\gamma) := TolH_k^{-\gamma} + (1-\delta_{k0})\, TolH_{k-1}^{-\gamma}.$$

For further estimation of the complexity $C_\epsilon^{\mathrm{MLMC}}$, we need the following property of a geometric sum with $q \in (0,1)$ and $\alpha > 0$:

$$\sum_{k=0}^{K} q^{\alpha k} = \frac{1 - q^{\alpha(K+1)}}{1 - q^\alpha} < \frac{1}{1 - q^\alpha} \tag{4.17}$$

since $q^\alpha < 1$. Together with the formula (4.15) for K, it follows

$$\begin{aligned} \sum_{k=0}^{K} q^{-\alpha k} = \sum_{k=0}^{K} (q^\alpha)^{k-K} = q^{-\alpha K} \sum_{k=0}^{K} \left(q^\alpha\right)^k < \frac{q^{-\alpha K}}{1 - q^\alpha} \\ \lesssim q^{-\alpha\,(\log_q(\epsilon/(2C_H TolH_0))+1+K_0)} \eqsim \epsilon^{-\alpha}. \end{aligned} \tag{4.18}$$

First, we consider the second term (II) on the right-hand side of (4.16). The assumed geometric

design of $TolH_k = q^k\, TolH_0$ and inequality

$$(II) = C_W \sum_{k=0}^{K} TolH_k^{-\gamma} + (1-\delta_{k0})\, TolH_{k-1}^{-\gamma} = C_W\, TolH_0^{-\gamma} \sum_{k=0}^{K} q^{-\gamma k} + (1-\delta_{k0})\, q^{-\gamma(k-1)}$$
$$\leq C_W\, TolH_0^{-\gamma}\big(1 + (1-\delta_{k0})\, q^{\gamma}\big) \sum_{k=0}^{K} q^{-\gamma k} \lesssim \frac{q^{-\gamma K}}{1-q^{\gamma}} \eqsim \epsilon^{-\gamma}. \tag{4.19}$$

For the sum $F(\gamma)$ in the term (I) of estimation (4.16), the relation $TolH_k = q^k\, TolH_0$ and Assumption 4.17 yield

$$\begin{aligned} F(\gamma) &= \sum_{k=0}^{K} \Big(TolH_k^{-\gamma} + (1-\delta_{k0})\, TolH_{k-1}^{-\gamma}\Big)^{1/2}\ V[u_{h_k} - u_{h_{k-1}}]^{1/2} \\ &\leq TolH_0^{-\gamma} \left(V[u_0]^{1/2} + \sum_{k=1}^{K} \Big(q^{-\gamma k} + q^{-\gamma(k-1)}\Big)^{1/2} \sqrt{C_V TolH_k^{\beta}} \right) \\ &= TolH_0^{-\gamma} \left(V[u_0]^{1/2} + C_V^{1/2}\, TolH_0^{\beta/2}\, (1+q^{\gamma})^{1/2} \sum_{k=1}^{K} q^{-\gamma k/2} q^{\beta k/2} \right) \\ &\lesssim 1 + \sum_{k=1}^{K} q^{k(\beta-\gamma)/2} \eqsim \sum_{k=0}^{K} q^{k(\beta-\gamma)/2}. \end{aligned}$$

Consequently, we get

$$(I) \lesssim \epsilon^{-2} \left(\sum_{k=0}^{K} q^{k(\beta-\gamma)/2} \right)^2. \tag{4.20}$$

Now, we consider the geometric sum $(G) := \sum_{k=0}^{K} \big(q^{(\beta-\gamma)/2}\big)^k$ for the following three different cases:

(i) $\beta - \gamma > 0$:
For this case, inequality (4.17) provides that the geometric sum (G) is bounded by a constant independent of K. Therefore, we have $(I) \lesssim \epsilon^{-2}$ and it follows

$$C_{\epsilon}^{\mathrm{MLMC}} \lesssim \epsilon^{-2} + \epsilon^{-\gamma} \lesssim \epsilon^{-2}$$

since $\epsilon < 1$ and $2 \geq \min(\beta, \gamma) = \gamma$.

(ii) $\beta - \gamma = 0$:
Inserting formula (4.15) into the geometric sum (G) gives

$$(G) = \sum_{k=0}^{K} \Big(q^{(\beta-\gamma)/2}\Big)^k = K + 1 \leq \log_q \left(\frac{\epsilon}{2\, C_H\, TolH_0} \right) + 2 + K_0 \lesssim |\log(\epsilon)|$$

because it holds $|\log(\epsilon)| \geq 1$ due to $\epsilon \leq e^{-1}$. It follows

$$(I) \lesssim \epsilon^{-2} \log(\epsilon)^2$$

and finally

$$C_{\epsilon}^{\mathrm{MLMC}} \lesssim \epsilon^{-2} \log(\epsilon)^2 + \epsilon^{-\gamma} \lesssim \epsilon^{-2} \log(\epsilon)^2$$

since $\beta = \gamma \leq 2$.

(iii) $\beta - \gamma < 0$:
Using inequality (4.18), we obtain $(G) \lesssim \epsilon^{-(\gamma-\beta)/2}$. Therefore, we have

$$(I) \lesssim \epsilon^{-2}\epsilon^{-(\gamma-\beta)}.$$

Combining the estimations for the terms (I) and (II) yields

$$C_\epsilon^{\mathrm{MLMC}} \lesssim \epsilon^{-2-(\gamma-\beta)} + \epsilon^{-\gamma} \lesssim \epsilon^{-2-(\gamma-\beta)}$$

because $\beta = \min(\gamma, \beta) \leq 2$ and thus $2 + (\gamma - \beta) > \gamma$. □

As in [18, 106], we interpret the complexity result of the previous theorem by considering (4.12) or directly (4.20). Moreover, we show the benefit of the ML approach by comparing the results with the complexity of the single-level MC approach proved in Lemma 4.8. In the optimal case $\beta > \gamma$, the variance-like term $V[u_{h_k} - u_{h_{k-1}}]$ decreases faster with k than the cost W_k increases. Hence, the terms $(W_k + W_{k-1})\, V[u_{h_k} - u_{h_{k-1}}]$ in (4.12) and $q^{k(\beta-\gamma)/2}$ in (4.20) are the largest for $k = 0$. Therefore, the most of the computational effort will be on the coarsest level $k = 0$ computing $E_{M_0}[u_{h_0}]$. Comparing with the single-level MC approach, the cost savings are $C_\epsilon^{\mathrm{MC}}/C_\epsilon^{\mathrm{MLMC}} \eqsim \epsilon^{-\gamma}$. If the variance-like term $V[u_{h_k} - u_{h_{k-1}}]$ decreases slower with k than the cost W_k increases, i.e. $\beta < \gamma$, the most of the computational effort is required on the finest level $k = K$. The savings in cost are $C_\epsilon^{\mathrm{MC}}/C_\epsilon^{\mathrm{MLMC}} \eqsim \epsilon^{-\beta}$. In the last case $\beta = \gamma$, the computational effort is distributed equally across the levels and the cost savings are up to a log factor of the order $\epsilon^{-\gamma}$.

If we neglect the condition $2 \geq \min(\gamma, \beta)$ in the previous theorem, then the term $\epsilon^{-\gamma}$ does not vanish and remains in the complexity estimates [13, 94]. Note that this term is generated by rounding up the number M_k of samples, see (4.12) and (4.19).

The previous complexity theorem is formulated for a given sequence of physical tolerances and determines the number $K \in \mathbb{N}_+$. In our case, we have access to an adaptive physical solver which works with an arbitrary physical tolerance as input parameter. Therefore, we can first choose a fixed $K \in \mathbb{N}_+$ and then the physical tolerance $TolH_K$ depending on C_H and the desired total accuracy ϵ:

$$TolH_K = \frac{\epsilon}{2\, C_H}.$$

The remaining tolerances are defined by $TolH_k = q^{k-K} TolH_K$. The choice of the number of samples and the complexity analysis are performed as before.

Remark 4.19
In order to generalize the previous theorem using an arbitrary Banach space $\mathcal{W}$ for the physical space, Assumption 4.17 has to be replaced by

$$\|\mathbb{E}[u_{h_k} - u_{h_{k-1}}] - E_{M_k}[u_{h_k} - u_{h_{k-1}}]\|^2_{L^2(\Theta,\mathcal{W})} \leq C\, M_k^{-1}\, TolH_k^\beta \quad k = 1, \ldots, K$$

with a constant $C > 0$ independent of M_k and u_{h_k}. Since the MC estimator is unbiased, the term on the left-hand side is equal to $V[E_{M_k}[u_{h_k} - u_{h_{k-1}}]]$ and can be interpreted as a variance-like term of the Monte Carlo estimator of the difference $u_{h_k} - u_{h_{k-1}}$.

If Assumption 4.11 also holds for the $\mathbb{L}^2$-norm, then we can obtain $\beta = 2$ by following the proof of Theorem 3.1 in [64].

4.3 Real-Valued Quantity of Interest

In application, a functional, also possibly nonlinear, of the solution is a typical QoI. In this section, we mainly follow the work of Cliffe et al. [18] and adapt the complexity analysis from the previous sections of this chapter to real-valued QoIs.

We consider a functional $\mathcal{F} : \mathbb{L}^1 \to \mathbb{R}$ and define the real-valued QoI $\Phi : \Theta \to \mathbb{R}$ by

$$\Phi : \theta \mapsto \mathcal{F}[u(\theta)],$$

where u is the random entropy solution. The functional of a random numerical approximation u_h of the solution u is denoted by Φ_h with $\Phi_h(\theta) := \mathcal{F}[u_h(\theta)]$ for $\theta \in \theta$ and it obviously approximates Φ.

4.3.1 Single-Level Structure

Analogously to the single-level approach considering the random entropy solution u, the expected value of the QoI Φ is approximated by the Monte Carlo estimator of the approximation Φ_h. For the definition of the MC estimator, we refer to Definition 4.1.

Similar to the solution case, we compute numerical approximations $\Phi_h(\theta)$ by using an adaptive physical solver which computes the approximate solution $u_h(\theta)$ with a sample-dependent resolution in space, time and model hierarchy. This resolution is refined until the error estimate for Φ is smaller than a given tolerance $TolH$. For the convergence and complexity analysis, we assume therefore the following properties.

Assumption 4.20 (Adaptive Physical Approximation)

(i) Let u be the random entropy solution of problem (3.3)–(3.7). For $\theta \in \Theta$, let $\Phi_h(\theta)$ be the functional Φ of the physical approximation $u_h(\theta)$ computed for a given tolerance $TolH > 0$ by an adaptive solver using a space-time-model discretization. Then, there exists a physical constant $c_h : \Theta \to \mathbb{R}_+$ with $\mathbb{E}[c_h] < \infty$ such that $\Phi_h(\theta)$ satisfies

$$|\Phi(\theta) - \Phi_h(\theta)| \leq c_h(\theta)\, TolH \tag{4.21}$$

$\mathbb{P}$-almost surely.

(ii) The functions Φ and Φ_h belong to the space $L^2(\Theta)$.

From this assumption, it follows

$$\mathbb{E}\left[|\Phi - \Phi_h|\right] \leq C_h\, TolH$$

with $C_h := \mathbb{E}[c_h]$ by integrating the inequality (4.21) over the sample space. Note that the constant C_h is finite and independent of $\theta \in \Theta$. In addition, Jensen's inequality yields

$$|\mathbb{E}[\Phi - \Phi_h]| \leq \mathbb{E}\left[|\Phi - \Phi_h|\right] \leq C_h\, TolH. \tag{4.22}$$

For the convergence analysis, we consider the mean squared error (MSE) of the MC estimator of the QoI Φ and expand the error to

$$\|\mathbb{E}[\Phi] - E_M[\Phi_h]\|^2_{L^2(\Theta)} = \mathbb{E}\Big[\big(\mathbb{E}[\Phi] - E_M[\Phi_h]\big)^2\Big] = \underbrace{\big(\mathbb{E}[\Phi] - \mathbb{E}[\Phi_h]\big)^2}_{I)\ deterministic\ error} + \underbrace{\mathbb{E}\Big[\big(\mathbb{E}[\Phi_h] - E_M[\Phi_h]\big)^2\Big]}_{II)\ stochastic\ error}. \tag{4.23}$$

The previous identity holds because the MC estimator is unbiased. If we consider the root mean squared error (RMSE), the previous splitting is an inequality. Therefore, in contrast to the Sections 4.1 and 4.2, we consider the MSE in this section. Investigating both error terms and supposing the above assumption give the following theorem.

Theorem 4.21
Suppose Assumptions 3.3 and 4.20 hold. Then, the MSE can be bounded by

$$\mathbb{E}\Big[\big(\mathbb{E}[\Phi] - E_M[\Phi_h]\big)^2\Big] \leq C_h^2\, TolH^2 + \frac{1}{M}\mathbb{V}[\Phi_h]. \tag{4.24}$$

Proof. First, we consider the deterministic physical error. Due to inequality (4.22), we have

$$\big(\mathbb{E}[\Phi] - \mathbb{E}[\Phi_h]\big)^2 = \mathbb{E}[\Phi - \Phi_h]^2 \leq C_h^2\, TolH^2.$$

Next, the stochastic error depends on the variance of the MC estimator:

$$\begin{aligned}\mathbb{E}\Big[\big(\mathbb{E}[\Phi_h] - E_M[\Phi_h]\big)^2\Big] &= \mathbb{E}\Big[\big(\mathbb{E}[E_M[\Phi_h]] - E_M[\Phi_h]\big)^2\Big] = \mathbb{V}\big[E_M[\Phi_h]\big] \\ &= \frac{1}{M^2}\mathbb{V}\Big[\sum_{n=1}^{M}\Phi_{h,n}\Big] = \frac{1}{M}\mathbb{V}[\Phi_h]\end{aligned} \tag{4.25}$$

because the random variables $\Phi_{h,n}$ are independent and identically distributed. Combining both error parts results in

$$\mathbb{E}\Big[\big(\mathbb{E}[\Phi] - E_M[\Phi_h]\big)^2\Big] = \mathbb{E}[\Phi - \Phi_h]^2 + \frac{1}{M}\mathbb{V}[\Phi_h] \leq C_h^2\, TolH^2 + \frac{1}{M}\mathbb{V}[\Phi_h].$$

Since $\Phi \in L^2(\Theta)$, inequality (3.2) holds such that $\mathbb{V}[\Phi_h] < \infty$. □

In order to guarantee an accuracy $\epsilon^2 > 0$ of the MSE, we balance both terms on the right-hand side of inequality (4.24) and obtain

$$TolH = \frac{\epsilon}{\sqrt{2}\, C_h} \quad \text{and} \quad M = \left\lceil \frac{2\,\mathbb{V}[\Phi_h]}{\epsilon^2} \right\rceil. \tag{4.26}$$

Note that this choice is not exactly the same as in (4.5) derived for the MC error of the full solution. The difference arises from the identity in (4.23).

The analysis of the total computational cost of the MC method for the QoI Φ is performed analogously to the solution case, in Section 4.1. Before we can analyze the computational complexity C_ϵ^{MC} required to achieve the desired accuracy ϵ^2, we need the following assumption on the cost of each sample of the approximated QoI Φ_h.

Assumption 4.22
Let W_h be an upper bound for the cost to compute an MC sample of Φ_h for tolerance $TolH > 0$. Then, there exists a constant $\gamma > 0$ such that

$$W_h \leq C_W \, TolH^{-\gamma}$$

with a constant $C_W > 0$ independent of γ and $TolH$.

Lemma 4.23
Let Assumptions 3.3, 4.20 and 4.22 be fulfilled. Then, for any $\epsilon < 1$, there exists a physical tolerance $TolH > 0$ and an integer $M \in \mathbb{N}_+$ such that

$$\mathbb{E}\Big[\big(\mathbb{E}[\Phi] - E_M[\Phi_h]\big)^2\Big] \leq \epsilon^2$$

and

$$C_\epsilon^{\mathrm{MC}} \leq \epsilon^{-2-\gamma}.$$

Proof. The proof is analogous to the proof of Lemma 4.8. □

Obviously, the RMSE also achieves the prescribed accuracy ϵ. Due to Jensen's inequality and Lemma 4.23, the mean absolute error satisfies

$$\mathbb{E}\Big[\big|\mathbb{E}[\Phi] - E_M[\Phi_h]\big|\Big] \leq \mathbb{E}\Big[\big(\mathbb{E}[\Phi] - E_M[\Phi_h]\big)^2\Big]^{1/2} \leq \epsilon. \tag{4.27}$$

4.3.2 Multi-Level Structure

In the following, we apply the multi-level Monte Carlo estimator defined in Definition 4.12 to a sequence of functionals Φ_{h_k} of the approximate solution u_{h_k}, i.e. $\Phi_{h_k}(\theta) := \mathcal{F}[u_{h_k}(\theta)]$ for $\theta \in \Theta$. As in Section 4.2, the physical tolerances fulfill

$$1 \geq TolH_0 > \cdots > TolH_K > 0$$

and each approximation Φ_{h_k} is computed for tolerance $TolH_k$ using an adaptive solver.

Assumption 4.24 (Sequence of Adaptive Physical Approximations)

(i) Let u be the entropy solution of problem (3.3)–(3.7). For $\theta \in \Theta$, let $\Phi_{h_k}(\theta)$ be the functional of the numerical approximation $u_{h_k}(\theta)$ computed for a given $TolH_k > 0$ by an adaptive solver. Then, for all $k = 0, \ldots, K$ there exists a physical constant $c_k : \Theta \to \mathbb{R}_+$ with $\mathbb{E}[c_k] < \infty$ such that $\Phi_{h_k}(\theta)$ satisfies

$$|\Phi(\theta) - \Phi_{h_k}(\theta)| \leq c_k(\theta) \, TolH_k \tag{4.28}$$

$\mathbb{P}$-almost surely.

(ii) It holds $\Phi, \Phi_{h_k} \in L^2(\Theta) \quad$ for $k = 0, \ldots, K$.

Analogously to the single-level approach, we split the MSE into a deterministic and the stochastic error. Due to the previous assumption, we can state the following theorem mainly based on [18].

Theorem 4.25
Let Assumptions 3.3 and 4.24 be fulfilled. Then, the error of the MLMC estimator of Φ satisfies

$$\mathbb{E}\Big[\big(\mathbb{E}[\Phi] - E^K[\Phi_{h_K}]\big)^2\Big] \leq C_H^2\, TolH_K^2 + \sum_{k=0}^{K} \frac{1}{M_k} \mathbb{V}[\Phi_{h_k} - \Phi_{h_{k-1}}]. \tag{4.29}$$

Proof. This proof is mainly based on [18]. Using the unbiasedness of the MLMC estimator, the independence of the MC estimators E_{M_k} and the linearity of the expected value implies that

$$\begin{aligned}\mathbb{E}\Big[\big(\mathbb{E}[\Phi] - E^K[\Phi_{h_K}]\big)^2\Big] &= \mathbb{E}\Big[\big(\mathbb{E}[\Phi] - \mathbb{E}[\Phi_{h_K}]\big)^2\Big] + \mathbb{E}\Big[\big(\mathbb{E}[\Phi_{h_K}] - E^K[\Phi_{h_K}]\big)^2\Big] \\ &= \big(\mathbb{E}[\Phi] - \mathbb{E}[\Phi_{h_K}]\big)^2 + \sum_{k=0}^{K} \mathbb{E}\Big[\big(\mathbb{E}[\Phi_{h_k} - \Phi_{h_{k-1}}] - E_{M_k}[\Phi_{h_k} - \Phi_{h_{k-1}}]\big)^2\Big].\end{aligned} \tag{4.30}$$

Analogous to equation (4.25), we obtain

$$\begin{aligned}\mathbb{E}\Big[\big(\mathbb{E}[\Phi] - E^K[\Phi_{h_K}]\big)^2\Big] &= \mathbb{E}[\Phi - \Phi_{h_K}]^2 + \sum_{k=0}^{K} \mathbb{V}\big[E_{M_k}[\Phi_{h_k} - \Phi_{h_{k-1}}]\big] \\ &= \mathbb{E}[\Phi - \Phi_{h_K}]^2 + \sum_{k=0}^{K} \frac{1}{M_k} \mathbb{V}\big[\Phi_{h_k} - \Phi_{h_{k-1}}\big].\end{aligned}$$

The variance $\mathbb{V}[\Phi_{h_k} - \Phi_{h_{k-1}}]$ is finite since $\Phi_{h_k} - \Phi_{h_{k-1}}$ belongs to $L^2(\Theta)$, see (3.2). Similar to (4.22), integrating the assumed inequality (4.28) over the sample space Θ and using Jensen's inequality lead to

$$|\mathbb{E}[\Phi - \Phi_{h_k}]| \leq \mathbb{E}[|\Phi - \Phi_{h_k}|] \leq C_H\, TolH_k \quad \text{for } k = 0, \dots, K$$

with $C_H := \max_{k=0,\dots,K} \mathbb{E}[c_k]$. Therefore, the deterministic error satisfy the estimation

$$\mathbb{E}[\Phi - \Phi_{h_K}]^2 \leq C_H^2\, TolH_K^2. \qquad \square$$

In order to achieve a total accuracy $\epsilon^2 < 1$ of the MSE, we have to bound both terms on the right-hand side of (4.29) by $\epsilon^2/2$. We can reformulate the error and complexity statements of the MLMC error of the full solution for the solution functional Φ. Nevertheless, the upper bound for the complexity and the number of samples differ by a factor of 2 from the results in Section 4.2 since the analysis here is based on the identity (4.30) instead of the inequality (4.7). Now, let W_k be an upper bound for the cost of a single sample of Φ_{h_k} for $k = 0, \dots, K$.

Theorem 4.26
Let $\{TolH_k\}_{k=0,1,\dots}$ be a strictly decreasing null sequence of physical tolerances $TolH_k \in (0, 1]$. Further, let Assumptions 3.3 and 4.24 be fulfilled. Then, for any $\epsilon > 0$, there exists a number $K = K(\epsilon) \in \mathbb{N}_+$ and a sequence of number of samples $\{M_k\}_{k=0,\dots,K}$ such that

$$\mathbb{E}\Big[\big(\mathbb{E}[\Phi] - E^K[\Phi_{h_K}]\big)^2\Big] \leq \epsilon^2$$

and

$$C_\epsilon^{\mathrm{MLMC}} \leq 2\,\epsilon^{-2} \left(\sum_{k=0}^{K} (W_k + W_{k-1})^{1/2}\, \mathbb{V}[\Phi_{h_k} - \Phi_{h_{k-1}}]^{1/2} \right)^2 + \sum_{k=0}^{K} (W_k + W_{k-1}).$$

The optimal choice for the number of samples M_k is given by

$$M_k = \left\lceil \frac{2}{\epsilon^2} \left(\frac{\mathbb{V}[\Phi_{h_k} - \Phi_{h_{k-1}}]}{W_k + W_{k-1}} \right)^{1/2} \sum_{j=0}^{K} (W_j + W_{j-1})^{1/2} \, \mathbb{V}[\Phi_{h_j} - \Phi_{h_{j-1}}]^{1/2} \right\rceil.$$

Proof. We proceed similar as in the proof of Theorem 4.14 by considering the error estimate (4.29) of the previous theorem. We choose $K \in \mathbb{N}_+$ such that $C_H \, TolH_K \leq \epsilon/\sqrt{2}$. □

Next, we assume that the upper bound W_k of the cost of one sample of Φ_{h_k} increases with level k. The resulting formula for M_k is used in the implementation of the MLMC method in Subsection 4.4.1. In order to complete the complexity analysis, the variance of the difference $\Phi_{h_k} - \Phi_{h_{k-1}}$ is assumed to converge with rate β.

Assumption 4.27 (Variance Reduction)

(i) Let W_k be an upper bound for the cost to compute an MC sample of Φ_{h_k} for tolerance $TolH_k > 0$. There exists a constant $\gamma > 0$ with

$$W_k \leq C_W \, TolH_k^{-\gamma}$$

for all $k = 0, \ldots, K$. The constant $C_W > 0$ is independent of k, θ and γ.

(ii) There exists a constant $\beta > 0$ with

$$\mathbb{V}[\Phi_{h_k} - \Phi_{h_{k-1}}] \leq C_V \, TolH_k^{\beta} \quad \text{for } k > 0,$$

where C_V is a positive constant independent of k, θ and β.

Corollary 4.28

Let $\{TolH_k\}_{k=0,1,\ldots}$ be a strictly decreasing null sequence of physical tolerances $TolH_k \in (0, 1]$. Further, suppose Assumptions 3.3, 4.24 and 4.27 (i) hold. Then, for any $\epsilon > 0$, there exists a number $K = K(\epsilon) \in \mathbb{N}_+$ and a sequence of number of samples $\{M_k\}_{k=0,\ldots,K}$ such that

$$\mathbb{E}\Big[\big(\mathbb{E}[\Phi] - E^K[\Phi_{h_K}] \big)^2 \Big] \leq \epsilon^2$$

and

$$C_\epsilon^{\mathrm{MLMC}} \leq 2 \, C_W \, \epsilon^{-2} \Big(\sum_{k=0}^{K} G_k(\gamma)^{1/2} \, \mathbb{V}[\Phi_{h_k} - \Phi_{h_{k-1}}]^{1/2} \Big)^2 + \sum_{k=0}^{K} G_k(\gamma)$$

with

$$G_k(\gamma) := TolH_k^{-\gamma} + (1 - \delta_{k0}) \, TolH_{k-1}^{-\gamma}.$$

The optimal choice for the number of samples M_k is given by

$$M_k = \left\lceil 2 \, \epsilon^{-2} \left(\frac{\mathbb{V}[\Phi_{h_k} - \Phi_{h_{k-1}}]}{G_k(\gamma)} \right)^{1/2} \sum_{j=0}^{K} G_j(\gamma)^{1/2} \, \mathbb{V}[\Phi_{h_j} - \Phi_{h_{j-1}}]^{1/2} \right\rceil. \tag{4.31}$$

Proof. This statement is proved analogously to the proof of Corollary 4.16 by considering the error estimate (4.29). □

Theorem 4.29
Let a null sequence of physical tolerances $\{TolH_k\}_{k=0,1,\dots}$ be given by $TolH_k = q^k\, TolH_0$ with $TolH_0 \in (0,1]$ and a reduction factor $q \in (0,1)$. Further, let Assumptions 3.3, 4.24 and 4.27 be fulfilled. Suppose that the rates γ and β of Assumption 4.27 satisfy the condition $2 \geq \min(\gamma, \beta)$. Then, for any $\epsilon \in (0, 1/e]$, there exists a number $K = K(\epsilon) \in \mathbb{N}_+$ and a sequence $\{M_k\}_{k=0,\dots,K}$ such that

$$\mathbb{E}\Big[\big(\mathbb{E}[\Phi] - E^K[\Phi_{h_K}]\big)^2\Big] \leq \epsilon^2$$

and

$$C_\epsilon^{\mathrm{MLMC}} \lesssim \begin{cases} \epsilon^{-2} & if\ \beta > \gamma, \\ \epsilon^{-2}(\log \epsilon)^2 & if\ \beta = \gamma, \\ \epsilon^{-2-(\gamma-\beta)} & if\ \beta < \gamma. \end{cases}$$

Proof. This follows from the proof of Theorem 4.18 by continuing with the results of the previous corollary. □

For details about the interpretation of the previous complexity result and the cost savings compared to the single-level approach, we refer to the discussion in Subsection 4.2.1.

Since we have a physical adaptive solver for the functional Φ of the solution u in the implementation, we choose first the number $K \in \mathbb{N}_+$ and then we define the physical tolerance $TolH_K$ for the first level by

$$TolH_K = \frac{\epsilon}{\sqrt{2}\, C_H}.$$

The remaining tolerances $\{TolH_k\}_{k=0}^{K-1}$ follow from the relation $TolH_k = q^{k-K} TolH_0$. The choice of the number of samples and the complexity analysis remain the same.

Furthermore, we can even show that Assumption 4.24 provides that Assumption 4.27 (ii) is fulfilled with $\beta = 2$.

Lemma 4.30
Suppose Assumption 3.3 holds. Further, let Assumption 4.24 be fulfilled with $\mathbb{E}[c_k^2] < \infty$. If the physical tolerances have a geometric design $TolH_k = q^k\, TolH_0$, then there exists a constant $C_V > 0$ independent of k such that

$$\mathbb{V}[\Phi_{h_k} - \Phi_{h_{k-1}}] \leq C_V\, TolH_k^2 \quad for\ k > 0.$$

Proof. We consider in this proof always $k > 0$ since $\mathbb{V}[\Phi_{h_0} - \Phi_{h_{-1}}] = \mathbb{V}[\Phi_{h_0}]$. The variance can be bounded by

$$\begin{aligned} \mathbb{V}[\Phi_{h_k} - \Phi_{h_{k-1}}] &= \mathbb{E}\Big[\big(\mathbb{E}[\Phi_{h_k} - \Phi_{h_{k-1}}] - (\Phi_{h_k} - \Phi_{h_{k-1}})\big)^2\Big] \\ &= \mathbb{E}\Big[\big(\Phi_{h_k} - \Phi_{h_{k-1}}\big)^2\Big] - \mathbb{E}[\Phi_{h_k} - \Phi_{h_{k-1}}]^2 \\ &\leq \mathbb{E}\Big[\big(\Phi_{h_k} - \Phi_{h_{k-1}}\big)^2\Big] \end{aligned} \tag{4.32}$$

since $\Phi_{h_k}, \Phi_{h_{k-1}} \in L^2(\Theta)$ and thus $|\mathbb{E}[\Phi_{h_k} - \Phi_{h_{k-1}}]| \leq |\mathbb{E}[\Phi_{h_k}]| + |\mathbb{E}[\Phi_{h_{k-1}}]| < \infty$. Due to the triangle inequality, Assumption 4.24 and the geometric design of the tolerances $TolH_k$, we obtain for almost all $\theta \in \Theta$

$$\begin{aligned} |\Phi_{h_k}(\theta) - \Phi_{h_{k-1}}(\theta)| &\leq |\Phi(\theta) - \Phi_{h_k}(\theta)| + |\Phi(\theta) - \Phi_{h_{k-1}}(\theta)| \\ &\leq c_k(\theta)(1 + q^{-1})\, TolH_k. \end{aligned}$$

By integrating the square of the previous inequality over the sample space Θ, we can derive the following estimation from (4.32):

$$\mathbb{V}[\Phi_{h_k} - \Phi_{h_{k-1}}] \leq \mathbb{E}\Big[\big(\Phi_{h_k} - \Phi_{h_{k-1}}\big)^2\Big] = \int_\Theta \big(\Phi_{h_k}(\theta) - \Phi_{h_{k-1}}(\theta)\big)^2 \,\mathrm{d}\mathbb{P}(\theta) \leq \hat{C}_H \,(1+q^{-1})^2\, TolH_k^2$$

with $\hat{C}_H = \max_{k=0,\dots,K} \mathbb{E}[c_k^2] > 0$. □

4.4 Implementation

We describe the algorithm of our single-level and multi-level Monte Carlo method. Both algorithms are fully error-controlled such that the provided approximation achieves an accuracy close to the user-prescribed accuracy ϵ.

First, we draw M_{init} different samples $\widehat{I}_n := I(\theta_n)$ of the random input data field I according to its given distribution, for example the random initial value or random boundary data. For each sample $\widehat{I}_n$, we obtain a deterministic problem of the form (2.11)–(2.15). The resulting M deterministic problems are decoupled and different from each other because the MC estimate is defined with samples of i.i.d. random variables. Therefore, the computation of the corresponding numerical solutions can be parallelized in the implementation.

As explained in the previous sections, we assume to have an adaptive black box solver at hand which is called for each drawn sample of the random input field and for a given physical tolerance $TolH > 0$. Then, this algorithm numerically solves the deterministic network problem corresponding to the given sample of the random input field such that the numerical solution satisfies the accuracy requirement in Assumption 4.2. We denote the adaptive solver by $\mathrm{ADET}(\cdot,\cdot)$ and the physical approximation by

$$\widehat{u}_{h,n} := u_h(\theta_n) = \mathrm{ADET}(\widehat{I}_n, TolH)$$

which is a sample of the random numerical solution u_h.

Let ϵ be a user-prescribed accuracy for the error of the MC approximations. According to the error analysis in Section 4.1, we need to choose the tolerance $TolH$ and the number of samples M as in (4.5). The constant C_h can be estimated by a pre-study of a few samples for a small sequence of relatively coarse physical tolerances. For the computation of M, we use the initial set of M_{init} samples and estimate the variance-like term $V[u_h] = \|\mathbb{E}[u_h] - u_h\|^2_{L^2(\Theta,\mathbb{L}^2)}$ in (4.5) by the following estimator.

Definition 4.31 (Variance Estimator)
Let v be an element of the space $L^2(\Theta,\mathcal{W})$ with Banach space $\mathcal{W}$ and let $v_1,\dots,v_M \in L^2(\Theta,\mathcal{W})$ be i.i.d. random variables with the same probability distribution as v. The variance estimator is defined as

$$V_M[v] = \frac{1}{M-1}\sum_{n=1}^{M} \|v_n - E_M[v]\|^2_{\mathcal{W}} = \frac{1}{M-1}\sum_{n=1}^{M} \left\| v_n - \frac{1}{M}\sum_{j=1}^{M} v_j \right\|^2_{\mathcal{W}},$$

where $E_M[v]$ is the MC estimator defined in Definition 4.1.

Similar to (3.2), we obtain $V_M[v] \in L^2(\Theta)$ since the MC estimator $E_M[v]$ is also a mean-square integrable random variable. Using this estimator, we can compute the required number of samples M. If additional samples are now needed, we compute new samples and finally the MC estimate including all computed samples of u_h. The resulting SLMC algorithm is also described in Algorithm 4.1. In the implementation, the estimates E_h and V_h are realizations of the estimators $E_M[u_h]$ and $V_M[u_h]$, respectively. Note that, all samples $\widehat{I}_n$ are different from each other so that in general the samples $\widehat{u}_{h,n}$ differ as well. In Chapter 7, we set $M_{init} = 10$. For a control of the quality of the MC estimate, the term V_h and the number of samples M are usually recomputed in the end of the algorithm. If the recomputed number of samples is significantly higher than the number of the already drawn samples, it is reasonable to draw additional samples and to recompute the corresponding MC estimate.

Algorithm 4.1 Adaptive single-level Monte Carlo method for the approximation of the expected value $\mathbb{E}[u]$.

procedure ASINGLELEVELMC(ϵ, M_{init}, ADET)
  Estimate C_h
  $TolH = \epsilon/(2\,C_h)$
  Draw M_{init} samples $\widehat{I}_n$ of random input data
  Compute samples of u_h:
    $\widehat{u}_{h,n} = \text{ADET}(\widehat{I}_n, TolH), \quad \text{for } n = 1, \ldots, M_{init}$
  $E_h = 1/M_{init} \sum_{n=1,\ldots,M_{init}} \widehat{u}_{h,n}$
  $V_h = 1/(M_{init} - 1) \sum_{n=1,\ldots,M_{init}} \|\widehat{u}_{h,n} - E_h\|^2_{\mathbb{L}^1}$
  $M = \lceil 4\,\epsilon^{-2}\, C_{T,\Omega}\, V_h \rceil$.
  if $M > M_{init}$ **then**
    Draw new samples $\{\widehat{I}_n\}_{n=M_{init}+1,\ldots,M}$
    Compute new samples of u_h:
      $\widehat{u}_{h,n} = \text{ADET}(\widehat{I}_n, TolH)$, for $n = M_{init} + 1, \ldots, M$
    $E_h = 1/M \sum_{n=1,\ldots,M} \widehat{u}_{h,n}$
  end if
  return E_h
end procedure

The implementation of the fully adaptive MLMC method is described in Algorithm 4.2. For each level $k = 0, \ldots, K$, we start again with drawing M_{init} samples $\widehat{I}^k_n := I(\theta^k_n)$ of the random input data I for $n = 1, \ldots, M_{init}$. Note that all $(K+1)M_{init}$ samples are different from each other. The numerical solution computed by the adaptive physical solver $\text{ADET}(\cdot,\cdot)$ for the sample I^k_n and for a given physical tolerance $TolH_k$ is denoted by

$$\widehat{u}^k_{h_k,n} := u_{h_k}(\theta^k_n) = \text{ADET}(\widehat{I}^k_n, TolH_k).$$

For $TolH_{k-1}$, we define analogously

$$\widehat{u}^k_{h_{k-1},n} := u_{h_{k-1}}(\theta^k_n) = \text{ADET}(\widehat{I}^k_n, TolH_{k-1}).$$

Algorithm 4.2 Adaptive multi-level Monte Carlo method for the approximation of the expected value $\mathbb{E}[u]$.

procedure AMULTILEVELMC(ϵ, q, K, M_{init}, ADET)
  Estimate C_H and γ
  $TolH_K = \epsilon/(2\,C_H)$
  $TolH_k = q^{k-K} TolH_K, \quad$ for $k = 0, \dots, K-1$
  Draw M_{init} samples $\widehat{I}_n^k$ for each level $k = 0, \dots, K$
  $\widehat{u}_{h_0,n}^0 = \text{ADET}(\widehat{I}_n^0, TolH_0), \quad$ for $n = 1, \dots, M_{init}$
  $E_0 = 1/M_{init} \sum_{n=1,\dots,M_{init}} \widehat{u}_{h_0,n}^0$
  $V_0 = 1/(M_{init}-1) \sum_{n=1,\dots,M_{init}} \|\widehat{u}_{h_0,n}^0 - E_0\|_{\mathbb{L}^1}^2$
  for $k = 1, \dots, K$ **do**
    $\widehat{u}_{h_k,n}^k - \widehat{u}_{h_{k-1},n}^k = \text{ADET}(\widehat{I}_n^k, TolH_k) - \text{ADET}(\widehat{I}_n^k, TolH_{k-1})$,
    for $n = 1, \dots, M_{init}$
    $E_k = 1/M_{init} \sum_{n=1,\dots,M_{init}} (\widehat{u}_{h_k,n}^k - \widehat{u}_{h_{k-1},n}^k)$
    $V_k = 1/(M_{init}-1) \sum_{n=1,\dots,M_{init}} \|(\widehat{u}_{h_k,n}^k - \widehat{u}_{h_{k-1},n}^k) - E_k\|_{\mathbb{L}^1}^2$
  end for
  $M_k = \left\lceil 4\,\epsilon^{-2}\, C_{T,\Omega}\, V_k^{1/2} G_k(\gamma)^{-1/2} \sum_{j=0,\dots,K} G_j(\gamma)^{1/2}\, V_j^{1/2} \right\rceil$ with $G_k(\gamma)$ as in (4.13)
    for $k = 0 \dots, K$
  if $M_0 > M_{init}$ **then**
    Draw new samples $\{\widehat{I}_n^0\}_{n=M_{init}+1,\dots,M_0}$
    $\widehat{u}_{h_0,n}^0 = \text{ADET}(\widehat{I}_n^0, TolH_0), \quad$ for $n = M_{init}+1, \dots, M_0$
    $E_0 = 1/M_0 \sum_{n=1\dots,M_0} \widehat{u}_{h_0,n}^0$
  end if
  for $k = 1, \dots, K$ **do**
    if $M_k > M_{init}$ **then**
      Draw new samples $\{\widehat{I}_n^k\}_{n=M_{init}+1,\dots,M_k}$
      $\widehat{u}_{h_k,n}^k = \text{ADET}(\widehat{I}_n^k, TolH_k), \quad$ for $n = M_{init}+1, \dots, M_k$
      $\widehat{u}_{h_{k-1},n}^k = \text{ADET}(\widehat{I}_n^k, TolH_{k-1}), \quad$ for $n = M_{init}+1, \dots, M_k$
      Update $E_k = 1/M_k \sum_{n=1,\dots,M_k} (\widehat{u}_{h_k,n}^k - \widehat{u}_{h_{k-1},n}^k)$
    end if
  end for
  return $E^K = \sum_{k=0}^K E_k$
end procedure

Since K is fixed, we choose $TolH_K = \epsilon/(2\,C_H)$ and the samples M_k as in (4.14). As in the complexity theorem 4.18, we consider a sequence of physical tolerances characterized by a geometric design such that $TolH_k = q^k\, TolH_0$ with a positive reduction factor $q < 1$. The constant C_H and the rate γ can be estimated as before using a few samples for a small sequence of relatively coarse

physical tolerances. If the samples studied in the SL approach for C_h are available, then they can be reused to estimate γ and C_h can be used as an estimation for C_H as well. Additionally, we need to estimate the variance-like term of the difference $u_{h_k} - u_{h_{k-1}}$ on each level k by the variance estimator introduced in the previous definition. If necessary, we compute additional samples and update the corresponding MC estimates. Finally, we sum up the MC estimates on each level providing the MLMC estimate. The estimates E_k, V_k and E^K are again realizations of the estimators $E_{M_k}[u_{h_k} - u_{h_{k-1}}], V_{M_k}[u_{h_k} - u_{h_{k-1}}]$ and $E^K[u_{h_K}]$, respectively.

4.4.1 Real-Valued Quantity of Interest

The previous implementation of the SLMC and MLMC algorithm for the solution can be easily transferred to the functional Φ of the solution u. In the functional setting, the physical solver ADET numerically solves deterministic network problems such that the functional of the numerical solution now satisfies the accuracy requirement in Assumption 4.20 or Assumption 4.24. These approximations are samples of Φ_h and they are denoted by

$$\widehat{\Phi}_{h,n} := \Phi_h(\theta_n) = \text{ADET}(\widehat{I}_n, TolH) \quad \text{and} \quad \widehat{\Phi}^k_{h_j,n} := \Phi_{h_j}(\theta^k_n) = \text{ADET}(\widehat{I}^k_n, TolH_j)$$

with $j \in \{k, k-1\}$. According to the analysis in Section 4.3, the formula for the tolerances and for the number of samples changes slightly since the MSE and not its root is considered. The physical tolerances increase by a factor $\sqrt{2}$ and in the formula for the number of samples, the constant $C_{T,\Omega}$ vanishes and the factor 4 reduces to 2. Moreover, the variance-like term $V[\cdot]$ coincides with the variance $\mathbb{V}[\cdot]$ for Φ_h and for the ML difference $\Phi_{h_k} - \Phi_{h_{k-1}}$. The corresponding variance estimator is unbiased $\mathbb{E}[V_M[\cdot]] = \mathbb{V}[\cdot]$ since the random variables in Definition 4.31 are independent of each other.

Algorithm 4.3 Adaptive single-level Monte Carlo method for the approximation of the expected value $\mathbb{E}[\Phi]$.

procedure ASINGLELEVELMC(ϵ, M_{init}, ADET)
  Estimate C_h
  $TolH = \epsilon/(\sqrt{2}\, C_h)$
  Draw M_{init} samples $\widehat{I}_n$ of random input data
  Compute samples of Φ_h:
    $\widehat{\Phi}_{h,n} = \text{ADET}(\widehat{I}_n, TolH), \quad \text{for } n = 1, \ldots, M_{init}$
  $E_h = 1/M_{init} \sum_{n=1,\ldots,M_{init}} \widehat{\Phi}_{h,n}$
  $V_h = 1/(M_{init} - 1) \sum_{n=1,\ldots,M_{init}} (\widehat{\Phi}_{h,n} - E_h)^2$
  $M = \lceil 2\, \epsilon^{-2}\, V_h \rceil$.
  if $M > M_{init}$ **then**
    Draw new samples $\{\widehat{I}_n\}_{n=M_{init}+1,\ldots,M}$
    Compute new samples of Φ_h:
      $\widehat{\Phi}_{h,n} = \text{ADET}(\widehat{I}_n, TolH)$, for $n = M_{init} + 1, \ldots, M$
    $E_h = 1/M \sum_{n=1,\ldots,M} \widehat{\Phi}_{h,n}$
  end if
  return E_h
end procedure

Algorithm 4.4 Adaptive multi-level Monte Carlo method for the approximation of the expected value $\mathbb{E}[\Phi]$.

procedure AMULTILEVELMC(ϵ, q, K, M_{init}, ADET)
  Estimate C_H and γ
  $TolH_K = \epsilon/(\sqrt{2}\, C_H)$
  $TolH_k = q^{k-K} TolH_K, \quad$ for $k = 0, \dots, K-1$
  Draw M_{init} samples $\widehat{I}_n^k$ for each level $k = 0, \dots, K$
  $\widehat{\Phi}_{h_0,n}^0 = \text{ADET}(\widehat{I}_n^0, TolH_0), \quad$ for $n = 1, \dots, M_{init}$
  $E_0 = 1/M_{init} \sum_{n=1,\dots,M_{init}} \widehat{\Phi}_{h_0,n}^0$
  $V_0 = 1/(M_{init}-1) \sum_{n=1,\dots,M_{init}} (\widehat{\Phi}_{h_0,n}^0 - E_0)^2$
  for $k = 1, \dots, K$ **do**
    $\widehat{\Phi}_{h_k,n}^k - \widehat{\Phi}_{h_{k-1},n}^k = \text{ADET}(\widehat{I}_n^k, TolH_k) - \text{ADET}(\widehat{I}_n^k, TolH_{k-1})$,
     for $n = 1, \dots, M_{init}$
    $E_k = 1/M_{init} \sum_{n=1,\dots,M_{init}} (\widehat{\Phi}_{h_k,n}^k - \widehat{\Phi}_{h_{k-1},n}^k)$
    $V_k = 1/(M_{init}-1) \sum_{n=1,\dots,M_{init}} \left((\widehat{\Phi}_{h_k,n}^k - \widehat{\Phi}_{h_{k-1},n}^k) - E_k \right)^2$
  end for
  $M_k = \left\lceil 2\,\epsilon^{-2}\, V_k^{1/2} G_k(\gamma)^{-1/2} \sum_{j=0,\dots,K} G_j(\gamma)^{1/2}\, V_j^{1/2} \right\rceil$ with $G_k(\gamma)$ as in (4.13)
     for $k = 0 \dots, K$
  if $M_0 > M_{init}$ **then**
    Draw new samples $\{\widehat{I}_n^0\}_{n=M_{init}+1,\dots,M_0}$
    $\widehat{\Phi}_{h_0,n}^0 = \text{ADET}(\widehat{I}_n^0, TolH_0), \quad$ for $n = M_{init}+1, \dots, M_0$
    $E_0 = 1/M_0 \sum_{n=1\dots,M_0} \widehat{\Phi}_{h_0,n}^0$
  end if
  for $k = 1, \dots, K$ **do**
    if $M_k > M_{init}$ **then**
      Draw new samples $\{\widehat{I}_n^k\}_{n=M_{init}+1,\dots,M_k}$
      $\widehat{\Phi}_{h_k,n}^k - \widehat{\Phi}_{h_{k-1},n}^k = \text{ADET}(\widehat{I}_n^k, TolH_k) - \text{ADET}(\widehat{I}_n^k, TolH_{k-1})$,
        for $n = M_{init}+1, \dots, M_k$
      Update $E_k = 1/M_k \sum_{n=1,\dots,M_k} (\widehat{\Phi}_{h_k,n}^k - \widehat{\Phi}_{h_{k-1},n}^k)$
    end if
  end for
  return $E^K = \sum_{k=0}^K E_k$
end procedure

Chapter 5

Adaptive Multi-Level Stochastic Collocation Method

In the previous chapter, we analyzed the adaptive multi-level Monte Carlo method. Another sampling method is the stochastic collocation (SC) method which determines the samples in a structured way. In contrast to Monte Carlo (MC) methods, SC methods are able to exploit any smoothness or special structure in the stochastic dependence. Therefore, the stochastic collocation method is usually an effective alternative which reduces the number of samples and the total computational effort. The stochastic collocation method is based on an interpolation in the usually multi-dimensional stochastic space. Consequently, we obtain an approximation of the considered quantity of interest (QoI) and not only statistical quantities of the QoI like in the Monte Carlo methods. However, stochastic collocation methods suffer from the curse of dimensionality meaning that the number of samples increases rapidly with the dimension of the stochastic space [106].

First, we introduce the idea of the SC method and present the interpolation based on sparse grids in the stochastic space as well as the development of an adaptive strategy in detail. Next, the (single-level) SC method is combined with adaptive strategies in the stochastic and the physical space such that we get a fully error-controlled method. Finally, we extend this method by a multi-level structure and analyze its convergence and complexity following the same concept as in the analysis of the MC methods in Chapter 4.

The SC method in this work focuses on the parametrized network problem (3.8)–(3.12) which we obtained in Section 3.3 under Assumption 3.4:

$$\left.\begin{aligned} \partial_t u^{(j)}(x,t,\mathbf{y}) + \partial_x F_{m_j}\big(u^{(j)}(x,t,\mathbf{y})\big) &= G_{m_j}\big(u^{(j)}(x,t,\mathbf{y}),x,t\big), \quad && m_j \in \mathcal{M}_{\mathrm{PDE}},\\ \partial_x F_{m_j}\big(u^{(j)}(x,t,\mathbf{y})\big) &= G_{m_j}\big(u^{(j)}(x,t,\mathbf{y}),x,t\big), && m_j \in \mathcal{M}_{\mathrm{QS}},\\ u^{(j)}(x,0,\mathbf{y}) &= u_0^{(j)}(x,\mathbf{y}), && x \in \Omega_j, \end{aligned}\right\} x \in \Omega_j,$$

for $j = 1,\dots,P$, $t \in \mathbb{R}_+$ and $\mathbf{y} \in \Gamma$. The problem is complemented by boundary and coupling conditions

$$\begin{aligned} \mathcal{B}_b\big(u^{(j)}(v_b,t,\mathbf{y})\big) &= H_b(t,\mathbf{y}), && b \in \mathcal{I}_{BC},\ j \in \delta(v_b),\\ \mathcal{C}_c\big(u^{(j_1)}(v_c,t,\mathbf{y}),\dots,u^{(j_{N_c})}(v_c,t,\mathbf{y})\big) &= \Pi_c(t,\mathbf{y}), && c \in \mathcal{I}_0,\ j_k \in \delta(v_c),\\ & && k = 1,\dots,N_c, \end{aligned}$$

for $t \in \mathbb{R}_+$, $\mathbf{y} \in \Gamma$ and $N_c := |\delta(v_c)|$. As a reminder, we have parametrized the uncertainty of the input into N independent random variables $\xi_i : \Theta \to \Gamma_i$. We have assumed that each variable ξ_i has a probability density function (PDF) $\rho_i : \Gamma_i \to \mathbb{R}_+$ and that $\Gamma_i = [-1, 1]$, see Assumptions 3.5 and 3.6. The stochastic parameter space $\Gamma = \Gamma_1 \times \cdots \times \Gamma_N = [-1, 1]^N$ is the image space of the random vector $\xi = (\xi_1, \ldots, \xi_N)$ and thus the product of the image spaces Γ_i.

The SC method applied to our parametrized network problem is characterized by the following steps:

- Choose Q collocation points $\mathbf{y}_i \in \Gamma$ for a discretization of the stochastic parameter space Γ. These points are realizations of the random vector $\xi = (\xi_1, \ldots, \xi_N)$.
- Insert each point $\mathbf{y}_i$ into the network problem and solve these decoupled Q problems. Indeed, the exact solution $u^{(j)}(x, t, \mathbf{y}_i)$ is usually unknown and therefore the problems are solved by a numerical method in order to get approximations $u_h^{(j)}(x, t, \mathbf{y}_i)$.
- Interpolate the physical approximations $u_h^{(j)}(x, t, \mathbf{y}_i)$ to obtain a continuous representation of the approximate stochastic solution $u_h^{(j)}(x, t, \mathbf{y})$. The interpolation points are the collocation points $\mathbf{y}_i$ chosen in the first step.
- Compute statistical quantities using the obtained interpolant.

In practice, the QoI is often a functional of the solution and not the full solution itself. In this case, we need to modify the previous procedure: In the second step, the functional of each approximation $u_h^{(j)}(x, t, \mathbf{y}_i)$ is computed. The third and the fourth steps are applied to the functional analogously.

In this work, we use global polynomials for the interpolation and so we require a certain smoothness of the solution in the stochastic space. In contrast to hyperbolic PDEs, the theory of the elliptic PDEs is well studied (see for example [6]): The regularity in the physical space of the solution guarantees the smoothness of the solution in the stochastic space. In the case of hyperbolic PDEs, discontinuities in the spatial dimension could be transferred to the stochastic space. In this case, the solution is not smooth in the stochastic space and oscillations may appear because of using a global interpolation. Therefore, we assume appropriate regularity of the solution in the stochastic space.

The key part of the SC method is the computation of the QoI (the solution or a functional of the solution) for every collocation point which is realized by solving the corresponding deterministic network problem and, hence, the underlying PDEs numerically. For complex PDE models as well as for sufficiently fine spatial and time discretizations, the numerical simulations can be very expensive and therefore we want to reduce the computational cost by using as few collocation points as possible. In the next section, we present such an interpolation in an N-dimensional space using sparse grids of collocation points and an adaptive algorithm which iteratively adds points to the grid, namely in regions where the QoI changes the most.

5.1 Sparse Grid Interpolation

In this section, we first present the interpolation in the multi-dimensional parameter space $[-1, 1]^N$ based on tensor products. Then, the Smolyak sparse grid interpolation proposed by Smolyak [102] and the generalized formula from Gerstner and Griebel [39] are considered. Then, we state the main properties of the general sparse grid operator and discuss how we can use the interpolant to compute statistical quantities. Furthermore, we give an overview of common

collocation points and explain the idea, the implementation and our modifications of the adaptive sparse grid algorithm introduced by Gerstner and Griebel [38] and extended by Nobile et al. [81].

In order to cover the treatment of the solution u of problem (3.8)–(3.12) and of a functional of the solution simultaneously, we consider a general function $f := (f_1, \ldots, f_N) : \Gamma \to \mathcal{W}$ with Banach space $(\mathcal{W}, \|\cdot\|_{\mathcal{W}})$. We suppose that the space $\mathcal{W}$ is either $\mathbb{R}$ or a function space containing functions $w : G \to \mathbb{R}^d$ with domain G and $d \geq 1$. In the latter case, we will skip the dependence on the variable $z \in G$ during this section because we focus on the interpolation in the stochastic space Γ.

For each univariate function $f_n \in C(\Gamma_n, \mathcal{W})$ with $\Gamma_n = [-1, 1]$, we consider one-dimensional interpolation operators which are needed for the construction of an interpolation of the function f in the multi-dimensional space. We approximate the function f_n by a global interpolation using Lagrange polynomials. However, there is no restriction in the method on this type of interpolation polynomials, for example piecewise linear basis functions with local support can also be used [74]. The interpolation points are called *collocation points* and are given by the set

$$Z^{m(i_n)} := \left\{ y_j^{(i_n)},\ j = 1, \ldots, m(i_n) \right\} \subset [-1, 1],$$

where the points $y_j^{(i_n)}$ are distinct. The index $i_n \in \mathbb{N}_+$ determines the number of collocation points $m(i_n)$ via the function $m : \mathbb{N}_+ \to \mathbb{N}_+$ which satisfies

$$m(1) = 1, \quad m(k) < m(k+1) \quad \text{for } k \geq 1.$$

Let $\mathcal{P}_{m(i_n)-1}(\Gamma_n)$ be the space of all one-dimensional polynomials $p : \Gamma_n \to \mathbb{R}$ of degree at most $m(i_n) - 1$. The interpolation operator $\mathcal{U}^{(i_n)} : C(\Gamma_n, \mathcal{W}) \to \mathcal{P}_{m(i_n)-1}(\Gamma_n) \otimes \mathcal{W}$ has to satisfy the property that the function is exactly interpolated at the nodes $y_j^{(i_n)}$:

$$\mathcal{U}^{(i_n)}\left[f_n\right]\left(y_j^{(i_n)}\right) = f_n\left(y_j^{(i_n)}\right)$$

for all $j = 1, \ldots, m(i_n)$. Therefore, the operator $\mathcal{U}_{i_n}$ is defined by

$$\mathcal{U}^{(i_n)}[f_n](y) := \sum_{j=1}^{m(i_n)} f_n\left(y_j^{(i_n)}\right) L_j^{(i_n)}(y),$$

using function evaluations in the collocation points and the Lagrange polynomials $L_j^{(i_n)}$ as basis functions of $\mathcal{P}_{m(i_n)-1}(\Gamma_n)$. The Lagrange polynomial $L_j^{(i_n)} : \Gamma_n \to \mathbb{R}$ corresponding to node $y_j^{(i_n)}$ has degree $m(i_n) - 1$ and is constructed by

$$L_j^{(i_n)}(y) = \prod_{\substack{k=1 \\ k \neq j}}^{m(i_n)} \frac{y - y_k^{(i_n)}}{y_j^{(i_n)} - y_k^{(i_n)}}$$

such that $L_j^{(i_n)}\left(y_k^{(i_n)}\right) = \delta_{jk}$ for $k = 1, \ldots, m(i_n)$. Obviously, the interpolation operator $\mathcal{U}^{m(i_n)}$ reproduces all polynomials contained in $\mathcal{P}_{m(i_n)-1}(\Gamma_n)$.

Now, we consider the N-dimensional hypercube $\Gamma = [-1,1]^N$ with $N > 1$ and the multivariate function $f \in C(\Gamma, \mathcal{W})$. Let $\mathbf{i} = (i_1, \ldots, i_N) \in \mathbb{N}_+^N$ be a multi-index where the component i_k determines the number of collocation points in k-th direction, namely $m_k(i_k)$. By means of tensor products, we combine the one-dimensional interpolation operators $\mathcal{U}_k^{(i_k)}$ of each dimension $k \in \{1, \ldots, N\}$ and their corresponding grids $Z^{m_k(i_k)}$. In order to avoid a complicated notation, we consider the case $m = m_1 = m_2 = \cdots = m_N$ resulting in $\mathcal{U}_k^{(i_k)} = \mathcal{U}^{(i_k)}$ and $Z^{m_k(i_k)} = Z^{m(i_k)}$. The following presentation can be transferred to the general case straightforwardly.

The full tensor product is now defined by

$$\begin{aligned} \mathcal{T}_{\mathbf{i}}[f](\mathbf{y}) &:= \left(\mathcal{U}^{(i_1)} \otimes \cdots \otimes \mathcal{U}^{(i_N)}\right)[f](\mathbf{y}) \\ &= \sum_{j_1=1}^{m(i_1)} \cdots \sum_{j_N=1}^{m(i_N)} f\left(y_{j_1}^{(i_1)}, \ldots, y_{j_N}^{(i_N)}\right) L_{j_1}^{(i_1)}(y_1) \cdots L_{j_N}^{(i_N)}(y_N) \end{aligned} \tag{5.1}$$

with $\mathbf{y} = (y_1, \ldots, y_N) \in \Gamma$ and collocation points $y_{j_k}^{(i_k)} \in Z^{m(i_k)}$ in the k-th dimension for $k = 1, \ldots, N$. The full tensor product is called isotropic if the number of collocation points in each direction coincide, i.e. $i_1 = i_2 = \cdots = i_N$. In order to write the tensor product in a shorter form, let $\mathbf{y}_{\mathbf{j}}^{\mathbf{i}} = (y_{j_1}^{(i_1)}, \ldots, y_{j_N}^{(i_N)})$ be the multivariate collocation points and $\mathcal{L}_{\mathbf{j}}^{\mathbf{i}} = L_{j_1}^{(i_1)} \cdots L_{j_N}^{(i_N)}$ be the corresponding multi-dimensional Lagrange polynomials. Then, we can describe the full tensor product by

$$\mathcal{T}_{\mathbf{i}}[f](\mathbf{y}) = \sum_{\mathbf{j}=\mathbf{1}}^{\mathbf{m(i)}} f\left(\mathbf{y}_{\mathbf{j}}^{\mathbf{i}}\right) \mathcal{L}_{\mathbf{j}}^{\mathbf{i}}(\mathbf{y}), \tag{5.2}$$

where we set $\mathbf{m(i)} := (m(i_1), \ldots, m(i_N))$. The set of the collocation points $\mathbf{y}_{\mathbf{j}}^{\mathbf{i}}$ used for the full tensor product builds the tensor grid

$$\begin{aligned} \mathbb{T}_{\mathbf{i}} &:= Z^{m(i_1)} \times Z^{m(i_2)} \times \cdots \times Z^{m(i_N)} \\ &= \left\{y_{j_1}^{(i_1)}\right\}_{j_1=1}^{m(i_1)} \times \left\{y_{j_2}^{(i_2)}\right\}_{j_2=1}^{m(i_2)} \times \cdots \times \left\{y_{j_N}^{(i_N)}\right\}_{j_N=1}^{m(i_N)}, \end{aligned}$$

which contains $m(i_1) \cdot m(i_2) \cdot \ldots \cdot m(i_N)$ points. In the isotropic case, the number of collocation points is given by $m(i_1)^N$ and grows exponentially with the dimension N of the space Γ. This behavior is called the *curse of dimensionality*. Consequently, the needed function evaluations increase very rapidly with the stochastic dimension and will be too expensive for high dimensions. The sparse grid construction reduces this strong dependence on the dimension N by using tensor products with relatively small grids as building blocks of the interpolation operator.

5.1.1 Smolyak Sparse Grid Construction

For the Smolyak formula [39, 102], we define the difference of two consecutive one-dimensional interpolation formula by

$$\Delta^{(i_n)} := \mathcal{U}^{(i_n)} - \mathcal{U}^{(i_n - 1)}$$

for $i_n \in \mathbb{R}_+$ with the convention that

$$\mathcal{U}^{(0)} := 0.$$

This operator is often called hierarchical surplus (operator) and represents the benefit of using the next more accurate interpolation rule. Using the hierarchical surplus, the univariate interpolation operators can be written as a telescoping sum $\mathcal{U}^{(k_n)} = \sum_{j_n=1}^{k_n} \Delta^{(j_n)}$ and the full tensor product has the representation

$$\mathcal{T}_{\mathbf{k}} = \sum_{\mathbf{j}=\mathbf{1}}^{\mathbf{k}} \Delta^{(j_1)} \otimes \cdots \otimes \Delta^{(j_N)}$$

for $\mathbf{k} \in \mathbb{R}_+^N$. The idea is now to truncate this sum such that the underlying grid contains significantly fewer points as a full tensor product. This construction delays the curse of dimensionality and will be introduced next. For the tensor product of the one-dimensional hierarchical surpluses, also called *hierarchical surplus*, we often use the shortened notation

$$\Delta_{\mathbf{i}} := \Delta^{(i_1)} \otimes \cdots \otimes \Delta^{(i_N)} = \bigotimes_{n=1}^{N} \left(\mathcal{U}^{(i_n)} - \mathcal{U}^{(i_n-1)} \right).$$

Definition 5.1 (Isotropic Smolyak Interpolant)
The isotropic Smolyak formula is defined as

$$\mathcal{S}_w := \sum_{\substack{\mathbf{i} \in \mathbb{N}_+^N \\ |\mathbf{i}| \leq w+N}} \Delta_{\mathbf{i}},$$

where $|\boldsymbol{i}| = i_1 + \cdots + i_N$ gives the sum over all entries of the multi-index $\boldsymbol{i}$. The parameter $w \in \mathbb{N}$ is called the level of Smolyak formula and starts with $w = 0$.

For $w = 0$ the Smolyak formula includes only the multi-index $\mathbf{i} = (1, \ldots, 1)$ such that a single collocation point is used. For Smolyak levels $w > 0$, we can write

$$\mathcal{S}_w = \mathcal{S}_{w-1} + \sum_{\substack{\mathbf{i} \in \mathbb{N}_+^N \\ |\mathbf{i}| = w+N}} \Delta_{\mathbf{i}}.$$

The formula is called isotropic, because all directions are treated equally. Inserting the difference operators, the formula can be written as a linear combination of tensor products of one-dimensional interpolation operators [112]:

$$\mathcal{S}_w = \sum_{\substack{\mathbf{i} \in \mathbb{N}_+^N \\ w+1 \leq |\mathbf{i}| \leq w+N}} (-1)^{w+N-|\mathbf{i}|} \binom{N-1}{w+N-|\mathbf{i}|} \cdot \left(\mathcal{U}^{(i_1)} \otimes \cdots \otimes \mathcal{U}^{(i_N)} \right). \tag{5.3}$$

Using the short notation (5.2) introduced for the tensor products provides the representation

$$\mathcal{S}_w[f](\mathbf{y}) = \sum_{\substack{\mathbf{i} \in \mathbb{N}_+^N \\ w+1 \leq |\mathbf{i}| \leq w+N}} c_{\mathbf{i}} \cdot \left(\sum_{\mathbf{j}=\mathbf{1}}^{\mathbf{m(i)}} f\left(\mathbf{y}_{\mathbf{j}}^{\mathbf{i}}\right) \mathcal{L}_{\mathbf{j}}^{\mathbf{i}}(\mathbf{y}) \right)$$

with the constants $c_{\mathbf{i}} = (-1)^{w+N-|\mathbf{i}|} \binom{N-1}{w+N-|\mathbf{i}|} \in \mathbb{Z}$.

In Figure 5.1, we compare the grid of the isotropic Smolyak interpolant $\mathcal{S}_w$ with the one of the corresponding full tensor product $\mathcal{T}_{(w+1,w+1)}$. We consider the Clenshaw-Curtis nodes (see Subsection 5.1.5), a two-dimensional parameter space $\Gamma = [-1,1]^2$ and the level $w = 5$. The tensor grid contains $33^2 = 1089$ points and the grid of the Smolyak formula, also called sparse grid, contains only 145 points. Hence, a significant reduction of the number of collocation points is achieved. Note that the Smolyak interpolant has a lower polynomial exactness than the full tensor product, see Subsection 5.1.3 and Figure 5.2. However, this disadvantage is usually compensated by the reduction in computation time.

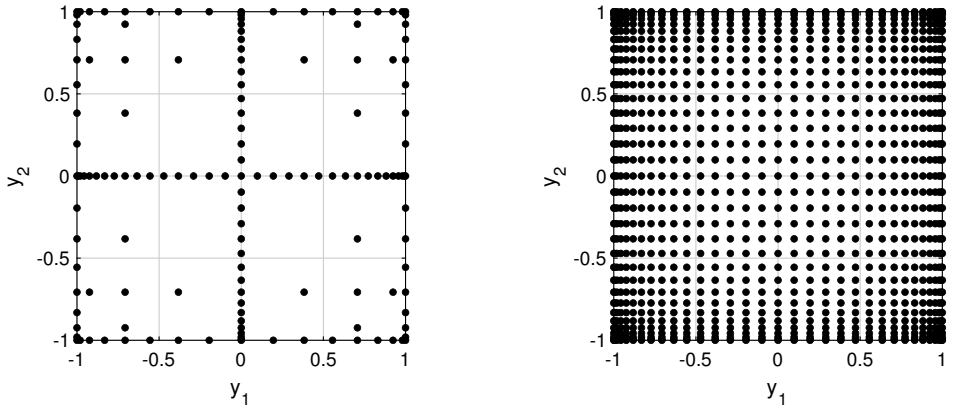


Figure 5.1: Collocation points used for the Smolyak formula of level $w = 5$ (left). Corresponding tensor grid $\mathcal{T}_{(6,6)}$ with 33 points in each direction (right). In both cases, Clenshaw-Curtis nodes are used.

5.1.2 General Sparse Grid Construction

One drawback of the isotropic Smolyak interpolant is that it handles all stochastic dimensions equally by using the same collocation points in each dimension. Therefore, the number of collocation points increases quickly with the Smolyak level w for higher dimensions N, but still significantly slower than in the case of the full tensor product. If coarser refinements are already sufficient for some dimensions, this approach is inefficient because more collocation points than required are used in these dimensions. Consequently, function evaluations and thus computation time could be saved. One approach to take a varying influence of the stochastic dimensions on the function f into account is the anisotropic Smolyak formula, see for example [80] for elliptic PDEs. Each dimension is separately weighted by multiplying each index i_n with a positive weight $a_n \in \mathbb{R}_+$ so that the multi-indices have to satisfy $|\mathbf{a} \cdot \mathbf{i}| = a_1 i_1 + \cdots + a_N i_N \leq w + N$. However, we need some a priori knowledge about the function f and the possible anisotropy in order to choose the weighting parameters a_n appropriately. In order to allow more general anisotropic structures, Gerstner and Griebel [38] presented a generalized sparse grid construction using a set of multi-indices $\mathbf{i} \in \Lambda$ such that the telescoping sum expansion of the general sparse grid formula remains valid and no interpolation rule is skipped in between.

Definition 5.2 (Downward Closed Index Set)
A multi-index set $\Lambda \subset \mathbb{N}_+^N$ is called downward closed or admissible, if

$$\forall \mathbf{i} \in \Lambda : \ \mathbf{i} - \mathbf{e}_j \in \Lambda \quad \textit{for all } j = 1, \ldots, N \textit{ with } i_j > 1.$$

Definition 5.3 (General Sparse Grid Interpolant)
Let Λ be a downward closed multi-index set. The generalized sparse grid interpolation operator is defined as

$$\mathcal{G}_\Lambda := \sum_{\mathbf{i}\in\Lambda} \Delta_\mathbf{i} = \sum_{\mathbf{i}\in\Lambda} \left(\Delta^{(i_1)} \otimes \cdots \otimes \Delta^{(i_N)}\right).$$

The general sparse grid interpolant can be written as a linear combination of tensor products (5.1) of one-dimensional interpolation operators:

$$\mathcal{G}_\Lambda[f] = \sum_{\mathbf{i}\in\Lambda} c_\mathbf{i} \mathcal{T}_\mathbf{i}[f] = \sum_{\mathbf{i}\in\Lambda} c_\mathbf{i} \left(\mathcal{U}^{(i_1)} \otimes \cdots \otimes \mathcal{U}^{(i_N)}\right)[f] \tag{5.4}$$

with

$$c_\mathbf{i} = \sum_{\mathbf{j}\in\{0,1\}^N} (-1)^{|\mathbf{j}|} \mathbb{1}_\Lambda(\mathbf{i}+\mathbf{j}) \ \in \mathbb{Z}, \tag{5.5}$$

see for example [39, 112]. This simplified representation is usually used in implementations. The coefficients $c_\mathbf{i}$ may also be equal to zero, for example if $\mathbf{i}+\mathbf{j} \in \Lambda$ for all $\mathbf{j} \in \{0,1\}^N$. Note that each hierarchical surplus $\Delta_\mathbf{i}$ is independent of the index set Λ. Therefore, the hierarchical surpluses will not change if Λ is enlarged. However, some coefficients $c_\mathbf{i}$ will obviously change. For example, if an index $\mathbf{k} \in \mathbb{N}_+^N$ is added to the set Λ, then the coefficient $c_\mathbf{i}$ only changes if there exists an integer $\mathbf{j} \in \{0,1\}^N$ such that it holds $\mathbf{k} = \mathbf{i}+\mathbf{j}$.

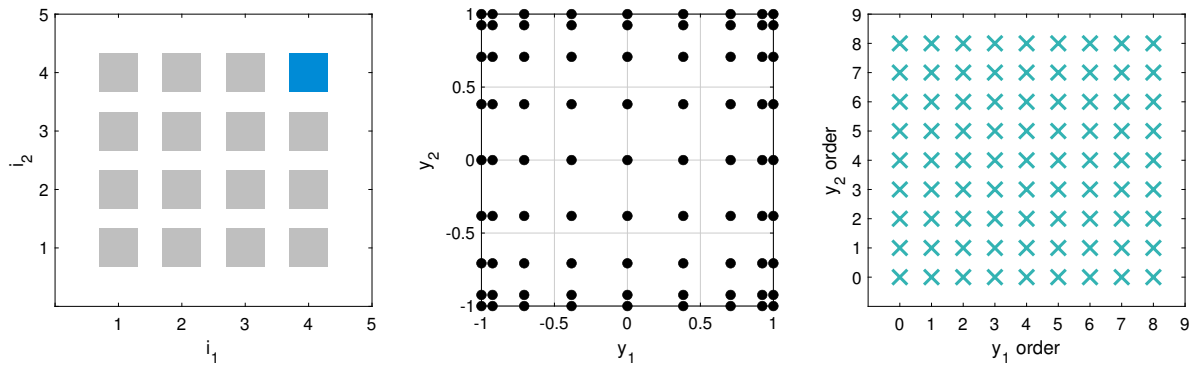
The computation of the previous formula (5.4) requires function evaluations of f on a finite set of collocations points, the so-called *sparse grid* $\mathbb{H}_\Lambda \subset \Gamma$. The sparse grid corresponding to $\mathcal{G}_\Lambda$ is built by the union of all tensor grids contained in (5.4) with a nonzero coefficient $c_\mathbf{i}$:

$$\mathbb{H}_\Lambda := \bigcup_{\substack{\mathbf{i}\in\Lambda \\ c_\mathbf{i}\neq 0}} Z^{m(i_1)} \times \cdots \times Z^{m(i_N)} \subset \Gamma.$$

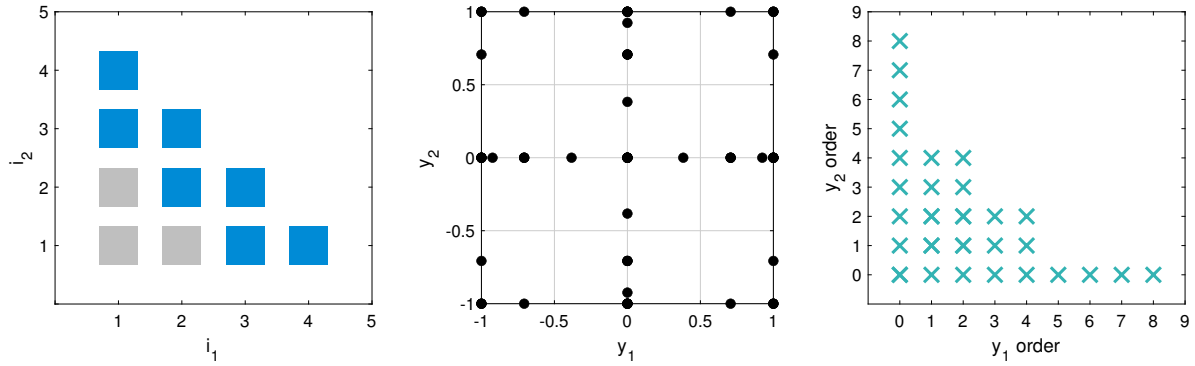
Obviously, the Smolyak sparse grid formula is a general sparse grid interpolant with the index set $\Lambda^{\mathrm{SMO}}(w) := \{\mathbf{i} \in \mathbb{N}_+^N : |\mathbf{i}| \leq w+N\}$. In this case, formula (5.4) reduces to (5.3) and the sparse grid is given by

$$\mathbb{H}_{\Lambda^{\mathrm{SMO}}(w)} = \bigcup_{\substack{\mathbf{i}\in\mathbb{N}_+^N \\ w+1\leq|\mathbf{i}|\leq w+N}} Z^{m(i_1)} \times \cdots \times Z^{m(i_N)} \tag{5.6}$$

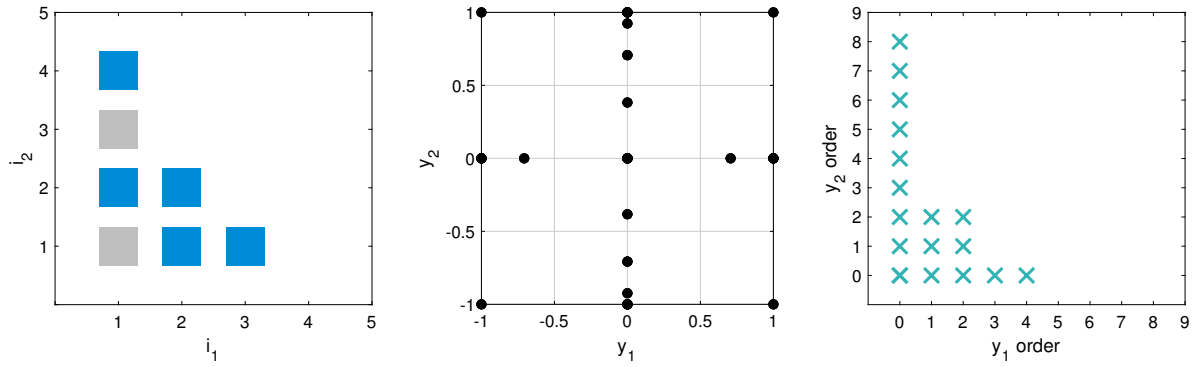
since $c_\mathbf{i} = 0$ if $|\mathbf{i}| \leq w$. The full tensor product which corresponds to the above Smolyak formula with level w is described by the downward closed index set $\{\mathbf{i} \in \mathbb{N}_+^N : \max_{n=1,\ldots,N}(i_n - 1) \leq w\}$. This tensor grid is the smallest one which contains all collocation points used for the Smolyak interpolant with level w. In Figure 5.2, we show three examples of a downward closed index set Λ and their corresponding sparse grid.



(a) Isotropic full tensor product $\mathcal{T}_{(4,4)}$.



(b) Isotropic Smolyak formula S_3.



(c) Arbitrary anisotropic formula.

Figure 5.2: Downward closed index sets (left) and the corresponding grids using Clenshaw-Curtis nodes which will be introduced in Subsection 5.1.5 (middle). The grey and blue squares represent the downward closed index set Λ. The blue squares correspond to the indices for which the coefficient $c_{\mathbf{i}}$ in (5.4) does not vanish. The right plots show the corresponding polynomial exactness discussed in the next Subsection.

Remark 5.4
Sparse Grids are also often used for numerical integration of multivariate integrals

$$\int_{[-1,1]^N} f(\mathbf{y}) w(\mathbf{y}) \, \mathrm{d}\mathbf{y}$$

with weight function $w(\mathbf{y}) = w_1(y_1) \cdots w_N(y_N)$ *[39]. In order to approximate this integral, we*

follow the procedure of the previous sections by replacing the interpolation operators $\mathcal{U}^{(i_n)}$ by univariate quadrature operators $Q^{(i_n)}$. The quadrature operators are given by

$$Q^{(i_n)}[f_n] := \sum_{j_n=1}^{m(i_n)} f_n\left(q_{j_n}^{(i_n)}\right) w_{n,j_n}^{(i_n)} \approx \int_{-1}^{1} f_n(y) w_n(y)\,\mathrm{d}y$$

with quadrature points $q_{j_n}^{(i_n)}$ and weights $w_{n,j_n}^{(i_n)}$. Analogously to Definition 5.3, we combine the quadrature operators using hierarchical surpluses and define the general sparse grid quadrature formula by

$$\mathcal{Q}_\Lambda = \sum_{\mathbf{i}\in\Lambda} \bigotimes_{n=1}^{N} \left(\mathcal{Q}^{(i_n)} - \mathcal{Q}^{(i_n-1)}\right).$$

Similar to (5.2) and (5.4), the operator can be written as

$$\mathcal{Q}_\Lambda[f] = \sum_{\mathbf{i}\in\Lambda} c_{\mathbf{i}} \left(\mathcal{Q}^{(i_1)} \otimes \cdots \otimes \mathcal{Q}^{(i_n)}\right)[f] = \sum_{\mathbf{i}\in\Lambda} c_{\mathbf{i}} \sum_{\mathbf{j}=\mathbf{1}}^{\mathbf{m(i)}} f\left(\mathbf{q}_{\mathbf{j}}^{\mathbf{i}}\right) w_{\mathbf{j}}^{\mathbf{i}} \tag{5.7}$$

with constants $c_{\mathbf{i}} = \sum_{\mathbf{k}\in\{0,1\}^N} (-1)^{|\mathbf{k}|} \mathbb{1}_\Lambda(\mathbf{i}+\mathbf{k})$, quadrature points $\mathbf{q}_{\mathbf{j}}^{\mathbf{i}} = \left(q_{j_1}^{(i_1)}, \ldots, q_{j_N}^{(i_N)}\right)$ and product weights $w_{\mathbf{j}}^{\mathbf{i}} = \prod_{n=1}^{N} w_{n,j_n}^{(i_n)}$. In order to get a very short representation, we enumerate all distinct quadrature points and denote them by $\{\mathbf{q}_k\}_{k=1}^{Q_\Lambda}$ such that we get the common quadrature form

$$\mathcal{Q}_\Lambda[f] = \sum_{k=1}^{Q_\Lambda} f(\mathbf{q}_k)\widehat{w}_k.$$

Note that the weights $\widehat{w}_k$ of the sparse grid quadrature formula can be negative even if the univariate operators $\mathcal{Q}^{(i_n)}$ have positive weights [39] and that they include the corresponding product weights $w_{\mathbf{j}}^{\mathbf{i}}$ and coefficients $c_{\mathbf{i}}$. Furthermore, enlarging the multi-index set Λ does not change the weights $\{w_{\mathbf{j}}^{\mathbf{i}}\}_{\mathbf{j}=\mathbf{1},\ldots,\mathbf{m(i)}}$ for $\mathbf{i}\in\Lambda$ and thus recomputations of these weights are not necessary in the implementation. However, the coefficients $c_{\mathbf{i}}$ can change and therefore also the weights $\widehat{w}_k$. Obviously, the polynomial exactness of the sparse grid quadrature depends on the chosen quadrature operators $\mathcal{Q}^{(i_n)}$ and the index set Λ. For example, if we use Clenshaw-Curtis quadrature and the index set $\Lambda^{\mathrm{SMO}}(w)$, the corresponding sparse grid quadrature has at least a degree $2w+1$ of exactness [83].

In the stochastic collocation context, the considered integral coincides with the expected value of the random field f if the weight function w_i is chosen as the PDF ρ_i of the underlying independent random variable ξ_i for all $i = 1, \ldots, N$.

5.1.3 Nested Nodes and Properties of the Interpolant

In order to increase the accuracy of the interpolant, the multi-index Λ has to be enhanced which implies a larger sparse grid. This requires new evaluations of the function f. The number of these evaluations can be highly reduced by using nested points. In the following, we address the advantage of nested points on the computational effort as well as the resulting property of the general sparse grid interpolant. Moreover, we investigate the polynomial exactness of the sparse grid interpolation.

The collocation points can be strictly divided into two categories: nested and non-nested points.

Definition 5.5
The univariate collocation points $y_j^{(i_n)} \in Z^{m(i_n)}$ with $i_n \in \mathbb{N}_+$ are called nested if it holds $y_j^{(i_n)} \in Z^{m(i_n+1)}$ for all $j = 1, \ldots, m(i_n)$, i.e. $Z^{m(i_n)} \subset Z^{m(i_n+1)}$. Otherwise, they are non-nested points.

If the univariate collocation points are nested, then the corresponding sparse grids are also nested such that each node used for a sparse grid interpolant with index set Λ belongs to the sparse grid of the interpolant with any larger index set $\Xi \supset \Lambda$, i.e. $\mathbb{H}_\Lambda \subset \mathbb{H}_\Xi$. This is a great advantage because we can reuse all previous evaluations computed for the smaller grid and thus we only need to evaluate the function f at the newly added point set instead of for the whole new and larger grid. In contrast to non-nested grids, the sparse grid interpolation operator constructed with nested points satisfies the interpolation property

$$\mathcal{G}_\Lambda[f]\left(\mathbf{y}_\mathbf{j}^\mathbf{i}\right) = f\left(\mathbf{y}_\mathbf{j}^\mathbf{i}\right)$$

for all $\mathbf{y}_\mathbf{j}^\mathbf{i} \in \mathbb{H}_\Lambda$, see [11, 65]. For nested points the sparse grid in (5.6) of the Smolyak formula simplifies to

$$\mathbb{H}_{\Lambda^{\mathrm{SMO}}(w)} = \bigcup_{\substack{\mathbf{i}\in\mathbb{N}_+^N \\ |\mathbf{i}|=w+N}} Z^{m(i_1)} \times \cdots \times Z^{m(i_N)}.$$

Focusing completely on nested points, it is useful to choose hierarchical Lagrange polynomials instead of nodal Lagrange polynomials which we consider in this work. If new collocation points are added, then the already computed hierarchical Lagrange polynomials do not have to be recomputed, see for example [95].

For considering the polynomial exactness of the general sparse grid formula, we define the polynomial space

$$\mathcal{P}_\Lambda(\Gamma) := \sum_{\mathbf{i}\in\Lambda} \left(\mathcal{P}_{m(i_1)-1}(\Gamma_1) \otimes \cdots \otimes \mathcal{P}_{m(i_N)-1}(\Gamma_N)\right).$$

The following properties are satisfied [20, 49, 65]:

Proposition 5.6

1. *For all $f \in C(\Gamma, \mathcal{W})$, it holds $\mathcal{G}_\Lambda[f] \in \mathcal{P}_\Lambda(\Gamma) \otimes \mathcal{W}$.*
2. *The general sparse grid operator $\mathcal{G}_\Lambda$ is exact on the space $\mathcal{P}_\Lambda(\Gamma) \otimes \mathcal{W}$, i.e. $\mathcal{G}_\Lambda[f] = f$ for all $f \in \mathcal{P}_\Lambda(\Gamma) \otimes \mathcal{W}$.*

For the Smolyak sparse grid construction, the polynomial space $\mathcal{P}_{\Lambda^{\mathrm{SMO}}(w)}(\Gamma)$ is simplified to

$$\mathcal{P}_{\Lambda^{\mathrm{SMO}}(w)}(\Gamma) = \sum_{\substack{\mathbf{i}\in\mathbb{N}_+^N \\ |\mathbf{i}|=w+N}} \left(\mathcal{P}_{m(i_1)-1}(\Gamma_1) \otimes \cdots \otimes \mathcal{P}_{m(i_N)-1}(\Gamma_N)\right),$$

see [11, 82]. Consequently, the Smolyak interpolation operator with a fixed level w has the same monomial exactness as the corresponding full tensor product, i.e. all monomials $y_n^{m(w+1)-1}$ for $n = 1 \ldots, N$ are reproduced exactly. However, the Smolyak formula has a reduced exactness for the mixed-terms polynomials due to its construction, for example $y_1^{m(w+1)-1} y_2^{m(w+1)-1}$ can be

interpolated exactly by the full tensor product, but not by the Smolyak formula (see Figure 5.2). If it holds $m(i) \geq i$ for $i \in \mathbb{N}_+$, the Smolyak interpolant is exact for all polynomials of total degree w, see [83] where a similar statement for Smolyak quadrature is proven. For example, if we use the Clenshaw-Curtis nodes introduced in Subsection 5.1.5, then the formula $\mathcal{S}_2$ reproduces the polynomials

$$y_j^4,\ y_j^3,\ y_j^2,\ y_j, 1, y_j^2 y_k^2,\ y_j^2 y_k,\ y_j y_k^2,\ y_j y_k$$

for $j, k \in \{1, \ldots, N\}$. The considered formula is exact for all polynomials of total degree 2.

5.1.4 Statistical Quantities

The sparse grid interpolant (Definition 5.3) of function f can be used further to approximate statistical quantities of f. We consider the function f as a random variable depending on the initially introduced random vector $\xi = (\xi_1, \ldots, \xi_N)$ with image space Γ. The components of ξ are independent real-valued random variables and each component ξ_i has a probability density function (PDF) ρ_i. Therefore, the random vector ξ has the joint PDF $\rho = \rho_1 \cdots \rho_N : \Gamma \to \mathbb{R}_+$. In the following, we present the computation of the expected value, higher moments as well as the variance of f using the interpolant $\mathcal{G}_\Lambda[f](\mathbf{y})$. If the function f depends on space and time, then these statistical quantities are also time- and space-dependent. In addition, the PDF of a real-valued function f can be approximated by a kernel density estimation which we study in Chapter 6.

Considering (5.4) and the notation introduced in (5.2), we obtain the detailed form

$$\mathcal{G}_\Lambda[f](\mathbf{y}) = \sum_{\mathbf{i} \in \Lambda} c_{\mathbf{i}} \mathcal{T}_{\mathbf{i}}[f](\mathbf{y}) = \sum_{\mathbf{i} \in \Lambda} c_{\mathbf{i}} \sum_{\mathbf{j}=\mathbf{1}}^{\mathbf{m(i)}} f\left(\mathbf{y}_{\mathbf{j}}^{\mathbf{i}}\right) \mathcal{L}_{\mathbf{j}}^{\mathbf{i}}(\mathbf{y}). \tag{5.8}$$

We integrate the sparse grid interpolant $\mathcal{G}_\Lambda$ multiplied by the joint PDF ρ over Γ in order to obtain an approximation of the mean $\mathbb{E}[f] \in \mathcal{W}$ of the function $f \in C(\Gamma, \mathcal{W})$ by

$$\mathbb{E}[f] \approx \mathbb{E}\big[\mathcal{G}_\Lambda[f]\big] = \sum_{\mathbf{i} \in \Lambda} c_{\mathbf{i}} \underbrace{\int_\Gamma \mathcal{T}_{\mathbf{i}}[f](\mathbf{y}) \rho(\mathbf{y}) \, \mathrm{d}\mathbf{y}}_{= \mathbb{E}[\mathcal{T}_{\mathbf{i}}]}.$$

As mentioned before, the joint PDF ρ is given by the product $\rho(\mathbf{y}) = \rho_1(y_1) \cdots \rho_N(y_N)$ and, hence, we get

$$\begin{aligned}
\mathbb{E}\big[\mathcal{T}_{\mathbf{i}}[f]\big] = \int_\Gamma \mathcal{T}_{\mathbf{i}}[f](\mathbf{y}) \rho(\mathbf{y}) \, \mathrm{d}\mathbf{y} &= \sum_{\mathbf{j}=\mathbf{1}}^{\mathbf{m(i)}} \int_\Gamma f\left(\mathbf{y}_{\mathbf{j}}^{\mathbf{i}}\right) \mathcal{L}_{\mathbf{j}}^{\mathbf{i}}(\mathbf{y}) \rho(\mathbf{y}) \, \mathrm{d}\mathbf{y} \\
&= \sum_{\mathbf{j}=\mathbf{1}}^{\mathbf{m(i)}} f\left(\mathbf{y}_{\mathbf{j}}^{\mathbf{i}}\right) \int_\Gamma L_{j_1}^{(i_1)}(y_1) \cdots L_{j_N}^{(i_N)}(y_N) \rho(\mathbf{y}) \, \mathrm{d}\mathbf{y} \\
&= \sum_{\mathbf{j}=\mathbf{1}}^{\mathbf{m(i)}} f\left(\mathbf{y}_{\mathbf{j}}^{\mathbf{i}}\right) \prod_{n=1}^{N} \int_{\Gamma_n} L_{j_n}^{(i_n)}(y) \rho_n(y) \, \mathrm{d}y.
\end{aligned}$$

Now, we need to approximate weighted integrals of one-dimensional Lagrange polynomials instead of multi-dimensional polynomials. These integrals can be approximated by appropriate

quadrature rules. The common choice is to use the interpolation points of the corresponding direction as quadrature points.

Therefore, the numerical integration is given by

$$\int_{\Gamma_n} L_{j_n}^{(i_n)}(y)\rho_n(y)\,\mathrm{d}y \approx \sum_{k=1}^{m(i_n)} L_{j_n}^{(i_n)}\left(y_k^{(i_n)}\right) w_{n,k}^{(i_n)} = w_{n,j_n}^{(i_n)}, \tag{5.9}$$

where $w_{n,k}^{(i_n)}$ are the weights of the univariate quadrature rule for an integral with the weight function ρ_n and quadrature points $y_k^{(i_n)}$. The quadrature weights depend in general on the PDF ρ_n, but not on the function f. If we consider the PDF of a uniform distribution and a quadrature rule which has at least a degree of $m(i_n)-1$, then (5.9) holds exactly. The expected value $\mathbb{E}[f]$ can be approximated by

$$\mathcal{E}_\Lambda[f] := \sum_{\mathbf{i}\in\Lambda} c_{\mathbf{i}} \sum_{\mathbf{j}=\mathbf{1}}^{\mathbf{m(i)}} f(\mathbf{y}_{\mathbf{j}}^{\mathbf{i}}) \prod_{n=1}^{N} w_{n,j_n}^{(i_n)} = \sum_{\mathbf{i}\in\Lambda} c_{\mathbf{i}} \sum_{\mathbf{j}=\mathbf{1}}^{\mathbf{m(i)}} f(\mathbf{y}_{\mathbf{j}}^{\mathbf{i}}) w_{\mathbf{j}}^{\mathbf{i}} \approx \mathbb{E}\big[\mathcal{G}_\Lambda[f]\big] \tag{5.10}$$

with $w_{\mathbf{j}}^{\mathbf{i}} = \prod_{n=1}^{N} w_{n,j_n}^{(i_n)}$. Therefore, we want to use univariate collocation points which provide a good quadrature rule and are suitable for Lagrange interpolation. Additionally, they have to be chosen according to the probability measure on Γ_n. In the following subsection, we present appropriate collocation points. Comparing (5.10) with the sparse grid quadrature formula (5.7) in Remark 5.4, we see that the formulas are equivalent if the collocation points coincide with the quadrature points and the PDFs ρ_i are chosen as the weight functions w_i since the interpolation is based on global Lagrange polynomials. Additionally, it holds $\mathcal{E}_\Lambda[f] = \mathcal{Q}_\Lambda\big[\mathcal{G}_\Lambda[f]\big]$. Of course, we can apply the sparse grid quadrature (5.7) directly to f to approximate its expected value. In this case, we would not obtain an approximation of f.

Often, higher moments of the function f are of interest as well. The approach is to apply the sparse grid interpolation operator (5.8) computed for f to f^r. We get the approximation

$$f^r(\mathbf{y}) \approx \mathcal{G}_\Lambda[f^r](\mathbf{y}) = \sum_{\mathbf{i}\in\Lambda} c_{\mathbf{i}} \sum_{\mathbf{j}=\mathbf{1}}^{\mathbf{m(i)}} f^r\left(\mathbf{y}_{\mathbf{j}}^{\mathbf{i}}\right) \mathcal{L}_{\mathbf{j}}^{\mathbf{i}}(\mathbf{y}).$$

Performing the same steps as before, the r-th moment $\mathbb{E}[f^r]$ can be approximated by

$$\mathcal{E}_\Lambda[f^r] = \sum_{\mathbf{i}\in\Lambda} c_{\mathbf{i}} \sum_{\mathbf{j}=\mathbf{1}}^{\mathbf{m(i)}} f^r\left(\mathbf{y}_{\mathbf{j}}^{\mathbf{i}}\right) w_{\mathbf{j}}^{\mathbf{i}}.$$

The coefficients $c_{\mathbf{i}}$ and the evaluations $f(\mathbf{y}_{\mathbf{j}}^{\mathbf{i}})$ are already calculated since we use the same collocation points and Lagrange polynomials as for the interpolation of f. The weights $w_{\mathbf{j}}^{\mathbf{i}}$ are the same for all moments and thus they have to be calculated only once.

Besides the moments of the function f, the variance is a typical statistical quantity. Proceeding analogously to the prior case, the variance can be estimated by

$$\mathbb{V}[f] = \mathbb{E}\left[(f-\mathbb{E}[f])^2\right] \approx \mathcal{E}_\Lambda\left[(f-\mathcal{E}_\Lambda[f])^2\right] = \sum_{\mathbf{i}\in\Lambda} c_{\mathbf{i}} \sum_{\mathbf{j}=\mathbf{1}}^{\mathbf{m(i)}} \left(f\left(\mathbf{y}_{\mathbf{j}}^{\mathbf{i}}\right) - \widehat{\mu}\right)^2 w_{\mathbf{j}}^{\mathbf{i}}$$

with $\widehat{\mu} := \mathcal{E}_\Lambda[f]$. For a bounded second moment $\mathbb{E}[f^2]$, we can use the expanded and simplified estimation

$$\mathbb{V}[f] = \mathbb{E}[f^2] - \mathbb{E}[f]^2 \approx \mathcal{E}_\Lambda[f^2] - \mathcal{E}_\Lambda[f]^2.$$

5.1.5 The Choice of Collocation Points

In general, arbitrary choices of the collocation points and of the function m are possible. However, we want to fulfill mainly three requirements. First, we want to obtain good interpolation properties. Using global Lagrange interpolation with equidistant points, the interpolant can diverge even if f is smooth, see for example [50, 108]. This is the so-called Runge's phenomen where oscillations near the boundary of the interval occur. Consequently, it is recommended to choose more collocation points close to the boundary. Secondly, we want to use good quadrature rules for the computation of the expected value and other statistical quantities as pointed out in the previous section. We need to compute one-dimensional, weighted integrals which have the general form

$$\int_{-1}^{1} L_p(y) w(y) \, \mathrm{d}y,$$

where the weight function w coincides with a PDF and the function L_p is a one-dimensional Lagrange polynomial, see (5.9). To this end, we want to apply quadrature rules using the interpolation points as quadrature points. If the random variable is uniformly distributed on $[-1, 1]$, then the weight function is constant ($w = 0.5$) and we directly consider the unweighted case ($w = 1$) since we only need to halve the weights of the unweighted case to get the weights for $w = 0.5$. For integration over the interval $[a, b]$ where the weight function w coincides with a PDF defined on the same interval, we only need to scale the collocation points which belong to the PDF of the same distribution on $[-1, 1]$. The corresponding weights can be taken without modifications. As the third condition, we prefer nested collocation points because all function evaluations computed for the collocation points of the current sparse grid can then be reused if new collocation points are added.

In the following, we give a short overview of common collocation points which are listed in Table 5.1 based on [81]. Basically, there are three classes: Clenshaw-Curtis, (extended) Gauss and Leja points.

Clenshaw-Curtis Nodes. The most common and well-established collocation points are the Clenshaw-Curtis points. For stochastic collocation methods using sparse grids, they have been utilized intensively in [49, 80, 106, 114]. We also select them in the numerical examples in Chapter 7. The one-dimensional Clenshaw-Curtis points are the extrema of Chebyshev polynomials [17]. They are given by

$$y_j^{(i)} = -\cos\left(\frac{\pi \cdot (j-1)}{m(i)-1}\right) \quad \text{for } j = 1, \cdots, m(i), \tag{5.11}$$

and with $m(i) > 0$. In order to generate nested point sets, we define

$$m(i) = 2^{i-1} + 1 \quad \text{for} \quad i > 1, \qquad m(1) = 1 \quad \text{with} \quad y_1^{(1)} := 0$$

so that all function evaluations computed for the collocation points of the current sparse grid can be reused if further collocation points are added. In addition, these points are located closer to the boundary than to the center of the interval. The Clenshaw-Curtis points are often used for

Points	Typical Measure	Domain	Nested?	$m(i)$
Clenshaw-Curtis	Uniform	$[-1,1]$	Yes	$2^{i-1}+1$
Gauss-Legendre	Uniform	$[-1,1]$	No	i
Gauss-Patterson	Uniform	$[-1,1]$	Yes	2^i-1
Leja	Uniform	$[-1,1]$	Yes	$m(i)=i$ or $m(i)=2i-1$
Gauss-Hermite	Gaussian	$(-\infty,\infty)$	No	i
Genz-Keister	Gaussian	$(-\infty,\infty)$	Yes	$m(i)=1,3,9,19,35$
Weighted Leja	Gaussian	$(-\infty,\infty)$	Yes	i

Table 5.1: Common choices for collocation points.

uniformly distributed random variables, but can also be applied for an arbitrary PDF ρ, see [95, 103] for a derivation of the formula of the weights. The collocation points remain the same and only the weights depend on the weight function. The numerical integration with Clenshaw-Curtis points has a polynomial exactness of $m(i)-1$ [103].

Gauss Formulas. One alternative is Gauss quadrature which provides the highest possible polynomial exactness of $2m(i)-1$. The Gauss-Legendre rule is used for uniform distributions and the Gauss-Hermite rule for Gaussian distributions. However, the corresponding quadrature points are non-nested. In [87], Patterson presented a recursion of the Gauss-Legendre formula in order to construct nested quadrature points, the Gauss-Patterson rule. This leads to the function $m(i)=2^i-1$ and a quadrature formula of degree $(3m(i)-1)/2$ [72]. For example, these points are applied in the context of sparse grids in [39]. The extension to arbitrary weight functions is in general complicated and not always possible [95]. Considering the PDF of Gaussian distributions as weight function, the Gauss-Hermite quadrature can be extended in a similar way to a rule with nested nodes, the Genz-Keister nodes [37].

Leja Sequences. Another recent choice for the uniform distribution are Leja sequences which are defined by a recursive formula [70]. In [79], Leja points are extended for arbitrary non-constant weight functions and additionally on unbounded domains, so for example for Gaussian distributed random variables. In contrast to Clenshaw-Curtis, the weighted Leja nodes depend on the weight function, i.e. on the PDF. In both cases, the points can be easily computed and are nested due to their recursive construction. In Table 5.1, we list the typical choice of the function m which, however, can be arbitrarily chosen.

Focusing on numerical integration, the Clenshaw-Curtis nodes could cause convergence problems for some high-dimensional functions while the Gauss-Patterson quadrature achieve a high accuracy [39, 72]. For a more detailed comparison of Gauss and Clenshaw-Curtis quadrature, we refer to [109]. In [79], it was observed that the Gauss-Patterson formula performs well for numerical integration, but not for interpolation since it was primarily developed for the approximation of integrals. Additionally, numerical examples have shown that, on the one hand, Leja sequences are better suited for interpolation than Clenshaw-Curtis nodes and, on the other hand, Clenshaw-Curtis nodes are superior to Leja sequences in the case of numerical integration. Finally, we can conclude, that using Clenshaw-Curtis nodes is an adequate compromise if both applications, interpolation and integration, will be considered.

5.1.6 Adaptive Sparse Grid Algorithm

In order to compute a sparse grid interpolation of f which achieves a desired accuracy $TolY > 0$, we need to choose the multi-index set Λ and thereby the number of collocation points appropriately. To this end, we need error estimates for the stochastic directions. The first approach is based on a priori error estimates, see for example [6, 8, 81]. Before computing sparse grid approximations, the multi-index set is determined by using analytical error estimates which usually require strong assumptions. In most application settings, these assumptions are not fulfilled. Therefore, we decide on a posteriori error estimations which are based on computations of sparse grid approximations. The common procedure is to start with an initial multi-index set and then to enrich this set during runtime deciding in an iterative and greedy-like strategy which region should be refined next. This adaptive algorithm is typically more flexible and robust [81].

Gerstner and Griebel [38] developed an adaptive algorithm for quadrature which automatically detects which stochastic parameter has the most influence in the QoI f and adds more collocation points in this direction. One extension of this algorithm is to control the anisotropy of the sparse grid by an additional parameter [60]. This modification is not considered in this work because some a priori knowledge about the QoI is needed to choose the parameter appropriately. Nobile et al. [81] considered sparse grid interpolation and extended the algorithm from Gerstner and Griebel [38] to non-nested points which we will present in the following. Additionally, they provide an open-source MATLAB® implementation *Sparse Grid Matlab Kit* of sparse grids including their adaptive algorithm [7, 104].

The idea of the a posteriori adaptive algorithm is to successively add multi-indices to the current index set used to build the sparse grid such that the sparse grid interpolant changes the most. As previously mentioned, we always want to construct downward closed index sets for the general sparse grid formula in Definition 5.3. If we add a multi-index $\mathbf{i} \in \mathbb{N}_+^N$ to a downward closed index set Λ such that the new index set $\Xi := \Lambda \dot\cup \{\mathbf{i}\}$ remains downward closed, then the new sparse grid operator satisfies

$$\mathcal{G}_\Xi[f] = \mathcal{G}_\Lambda[f] + \Delta_\mathbf{i}[f].$$

This formulation proposes that each hierarchical surplus $\Delta_\mathbf{i}$ can serve as a local error indicator for the multi-index $\mathbf{i}$. Now, we measure the sparse grid approximation error by a non-negative sublinear functional $\mathcal{J} : C(\Gamma, \mathcal{W}) \to \mathbb{R}_+$ which means that the conditions

$$\begin{aligned} \mathcal{J}[\alpha\, w] &= \alpha\, \mathcal{J}[w] \qquad \forall \alpha \in \mathbb{R}_+ \text{ and } w \in C(\Gamma, \mathcal{W}), \\ \mathcal{J}[v + w] &\le \mathcal{J}[v] + \mathcal{J}[w] \qquad \forall v, w \in C(\Gamma, \mathcal{W}) \end{aligned}$$

are fulfilled. One typical example for this functional is the $L^2_\rho(\Gamma, \mathcal{W})$-norm. We will show further examples and their computations later. Supposing that we have $f = \mathcal{G}_{\mathbb{N}_+^N}[f] = \sum_{\mathbf{i} \in \mathbb{N}_+^N} \Delta_\mathbf{i}[f]$, we obtain that the error depends on the hierarchical surpluses:

$$\mathcal{J}\big[f - \mathcal{G}_\Lambda[f]\big] = \mathcal{J}\Big[\sum_{\mathbf{i} \neq \Lambda} \Delta_\mathbf{i}[f]\Big] \le \sum_{\mathbf{i} \neq \Lambda} \underbrace{\mathcal{J}\big[\Delta_\mathbf{i}[f]\big]}_{\approx\, g(\mathbf{i})}.$$

In general, the error contributions $\mathcal{J}\big[\Delta_\mathbf{i}[f]\big] \in \mathbb{R}_+$ cannot be computed exactly and so we need to approximate these terms. We denote the non-negative approximation of $\mathcal{J}\big[\Delta_\mathbf{i}[f]\big]$ by $g(\mathbf{i})$, the so-called *profit* of multi-index $\mathbf{i}$. This profit can be used as a local error indicator since it represents the gain which we obtain if we add this multi-index to our existing index set.

Finally, the main step of the algorithm is to improve the sparse grid approximation by iteratively choosing the multi-indices with the highest profit. Another reasonable choice for the profit is to combine the error contributions with the number of evaluations required for adding the hierarchical surplus $\Delta_{\mathbf{i}}[f]$ to the sparse grid approximation, see [39, 60, 81]. In this case, the required work is also taken into account in the decision which index should be added next. In the following, we will focus on adaptive refinements in the stochastic space which are completely controlled by the error contributions and therefore we decide for the first choice $g(\mathbf{i}) \approx \mathcal{J}\big[\Delta_{\mathbf{i}}[f]\big]$. Note that the error indicators are only heuristic estimates and no classical error estimators.

Since we only allow downward closed index sets, we consider here a downward closed set $\mathcal{O}$ and define the *reduced margin* of $\mathcal{O}$ by

$$R_{\mathcal{O}} := \big\{\mathbf{i} \in \mathbb{N}_+^N \setminus \mathcal{O} : \mathbf{i} - \mathbf{e}_n \in \mathcal{O},\ \forall n = 1, \dots, N \text{ with } i_n > 1\big\}.$$

This index set contains all multi-indices $\mathbf{i}$ such that $\mathcal{O} \dot\cup \{\mathbf{i}\}$ is downward closed. Moreover, the *neighboring indices*, or short *neighbors*, of an index $\mathbf{i} \in \mathcal{O}$ (with respect to $\mathcal{O}$) are given by

$$neigh(\mathbf{i}, \mathcal{O}) := \big\{\mathbf{j} \in \mathbb{N}_+^N \setminus \mathcal{O} : \mathbf{j} = \mathbf{i} + \mathbf{e}_k \text{ for } k \in \{1, \dots, N\}\big\}.$$

We call a neighbor $\mathbf{j} \in neigh(\mathbf{i}, \mathcal{O})$ *admissible*, if $\mathcal{O} \dot\cup \{\mathbf{j}\}$ is again downward closed. The reduced margin of $\mathcal{O}$ contains all admissible neighbors of the indices $\mathbf{i} \in \mathcal{O}$. Figure 5.3 illustrates an example of a downward closed multi-index set and its reduced margin.

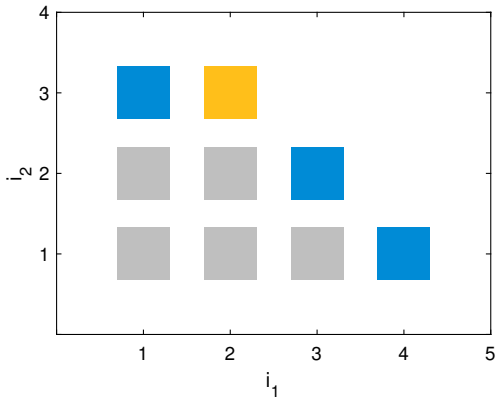


Figure 5.3: A downward-closed multi-index set $\mathcal{O}$ is given by the grey squares. The corresponding reduced margin $R_{\mathcal{O}}$ contains the multi-indices assigned to the blue squares. The index $(2, 3)$ represented by the orange square is a neighbor of index $(2, 2) \in \mathcal{O}$, but it is not admissible since index $(1, 3)$ is not contained in $\mathcal{O}$. Consequently, we have $(2, 3) \notin R_{\mathcal{O}}$.

As motivated above, we want to add the index to the current set $\mathcal{O}$ which has the largest profit and is contained in the set $R_{\mathcal{O}}$ as well. To this end, we have to compute hierarchical surpluses and profits for the whole set $R_{\mathcal{O}}$. For each index $\mathbf{i} \in R_{\mathcal{O}}$, the profit $g(\mathbf{i})$ can be computed using the two sparse grid approximations with the index sets $\mathcal{O} \cup \{\mathbf{i}\}$ and $\mathcal{O}$ since it is based on the hierarchical surplus $\Delta_{\mathbf{i}} = \mathcal{G}_{\mathcal{O}\cup\{\mathbf{i}\}} - \mathcal{G}_{\mathcal{O}}$. Choosing the index $\mathbf{i}_{max} \in R_{\mathcal{O}}$ with the highest profit, we obtain the sparse grid interpolation built for the index set $\mathcal{O} \cup \{\mathbf{i}_{max}\}$ as an approximation for f. In this case, we omit all functions evaluations computed anyway for the profits of the other indices $\mathbf{i} \in R_{\mathcal{O}} \setminus \{\mathbf{i}_{max}\}$ which we could have used to improve our sparse grid approximation.

Since this behavior is not efficient, we successively add the indices $\mathbf{i} \in R_{\mathcal{O}}$ to the currently used index set starting with the set $\mathcal{O}$ and compute the corresponding profit by using the sparse grid approximation of the new index set and of the previous one. Note that the hierarchical surplus $\Delta_{\mathbf{i}}$ and thereby profit $g(\mathbf{i})$ are independent of the index set to which $\mathbf{i}$ is added. Consequently, the hierarchical surplus and the profit do not change when further indices are added to the current index set and therefore the profits for $R_{\mathcal{O}}$ can be computed in any arbitrary order. Finally, we obtain the sparse grid approximation for the disjoint union $\Lambda := \mathcal{O} \cup R_{\mathcal{O}}$ instead of $\mathcal{O} \cup \{\mathbf{i}_{max}\}$, where $\mathbf{i}_{max} := \text{argmax}_{\mathbf{j} \in R_{\mathcal{O}}}\, g(\mathbf{j})$. In the next iteration step, we determine all admissible neighbors of the index $\mathbf{i}_a$ with respect to $\mathcal{O} \cup \{\mathbf{i}_{max}\}$, compute their profits as before and choose the index with the highest profit contained in $R_{\mathcal{O} \cup \{\mathbf{i}_{max}\}}$.

The whole adaptive procedure is described in detail in Algorithm 5.1. The algorithm starts with the initial multi-index $\mathbf{i}_{max} = \mathbf{1} = (1, \ldots, 1)$ which induces one collocation point and allows to reproduce constant functions exactly. We compute the sparse grid interpolant $\mathcal{G}_\Lambda$ and its expected value $\mathcal{E}_\Lambda$ for $\Lambda = \{\mathbf{i}\}$. Additionally, we set $\mathcal{O} = \{\mathbf{i}_{max}\}$ and $\mathcal{R} = \emptyset$. Next, we *explore* the neighbors of $\mathbf{i}_{max}$ which are given by $\mathbf{i}_k = \mathbf{i}_{max} + e_k$ for $k = 1 \ldots, N$: If index $\mathbf{i}_k$ is admissible, then we add $\mathbf{i}_k$ to Λ and compute the corresponding sparse grid quantities (interpolant, expected value and evaluations) in order to get the profit $g(\mathbf{i}_k)$. Additionally, we add the index $\mathbf{i}_k$ to the set $\mathcal{R}$. After having explored all indices $\mathbf{i}_k$, we obtain $\Lambda = \mathcal{R} \dot{\cup} \mathcal{O} = \{\mathbf{j} : \mathbf{j} = \mathbf{1} + e_k \text{ for } k = 1, \ldots, N\}$. At this point, the set $\mathcal{R}$ is equal to the reduced margin $R_{\mathcal{O}}$ of the set $\mathcal{O}$. Next, we update the global error indicator η which serves as an approximation of the total error and naturally depends on the values of the profits. We choose the maximum profit as global error indicator

$$\eta := \max_{\mathbf{j} \in \mathcal{R}} g(\mathbf{j}).$$

Another typical indicator η is the sum of all profits [38]. For this indicator, we observe for higher dimensional functions that the indicator is often more pessimistic meaning that the indicator differs from the error by at least one order of magnitude more than for our choice. In order to prepare the next iteration step, we already select the index $\mathbf{i}_{max}$ out of the active set $\mathcal{R}$ which has the highest profit. In the beginning of the next step, the stopping criterion is checked: The global error indicator has to be smaller than a user-prescribed tolerance $TolY > 0$. The stopping criterion often includes the condition that the number of collocation points are smaller than a user-prescribed upper bound [81]. If the global error indicator η is still larger than the prescribed tolerance $TolY$, then index $\mathbf{i}_{max}$ is added to $\mathcal{O}$ and removed from $\mathcal{R}$ such that we can start to explore the neighboring indices of $\mathbf{i}_{max}$ with respect to $\mathcal{O}$. Otherwise, the algorithm stops and returns the current as well as the previous sparse grid interpolant and their approximate expected values.

The index set $\mathcal{O}$ is called the *inactive* (or *old*) set and contains all indices whose neighbors have already been explored. The remaining indices $\mathbf{i} \in \mathcal{R}$ used for the sparse grid approximation are called *active* because these indices have already been explored, but their neighbors have remained unexplored so far. Of course, this set contains the index with the highest profit which will be considered next by exploring its neighboring indices. In each iteration step, the updated index set Λ contains all indices used to built the sparse grid and it holds $\Lambda = \mathcal{O} \cup \mathcal{R}$. Note that $\mathcal{O}$ and Λ are always downward closed.

Algorithm 5.1 Adaptive sparse grid algorithm.

procedure ASPAGRID($f, TolY$, profitName)
  $\mathbf{i}_{max} = (1, \ldots, 1)$
  $\mathcal{O} = \emptyset$
  $\Lambda = \{\mathbf{i}_{max}\}$
  $\mathcal{R} = \{\mathbf{i}_{max}\}$
  $\mathcal{G}_{old} = \mathcal{G}_\Lambda[f]$
  $\mathcal{E}_{old} = \mathcal{E}_\Lambda[f]$
  $\eta = \infty$
  while $\eta > TolY$ **do**
    $\mathcal{R} = \mathcal{R} \setminus \{\mathbf{i}_{max}\}$
    $\mathcal{O} = \mathcal{O} \cup \{\mathbf{i}_{max}\}$
    $a\mathcal{N}b = \{\mathbf{j} : \mathbf{j} \in neigh(\mathbf{i}_{max}, \mathcal{O})$ and $\mathcal{O} \cup \{\mathbf{j}\}$ is downward closed$\}$
    for $\mathbf{i} \in a\mathcal{N}b$ **do**
      $\Lambda = \Lambda \cup \{\mathbf{i}\}$
      $\mathcal{G} = \mathcal{G}_\Lambda[f]$
      $\mathcal{E} = \mathcal{E}_\Lambda[f]$
      if using nested points **then**
        $\mathcal{N}ew = \mathbb{H}_\Lambda \setminus \mathbb{H}_{\Lambda \setminus \{\mathbf{i}\}}$
        $e_{\mathcal{N}ew}$ = evaluations of f on each $y \in \mathcal{N}ew$
      else
        $\mathcal{N}ew = \mathbb{T}_\mathbf{i}$
        $e_{\mathcal{N}ew}$ = evaluations of $\mathcal{G}$ on each $y \in \mathcal{N}ew$
      end if
      e_{old} = evaluations of $\mathcal{G}_{old}$ on each $y \in \mathcal{N}ew$
      $g(\mathbf{i})$ = COMPUTEPROFITS(profitName, $e_{\mathcal{N}ew}, e_{old}, \mathcal{G}, \mathcal{G}_{old}, \mathcal{E}, \mathcal{E}_{old}$)
      $\mathcal{R} = \mathcal{R} \cup \{\mathbf{i}\}$
      $\mathcal{G}_{old} = \mathcal{G}$
      $\mathcal{E}_{old} = \mathcal{E}$
    end for ▷ $\mathcal{R} = R_\mathcal{O}$
    if $a\mathcal{N}b \neq \emptyset$ **then**
      Update η based on the profits: $\eta = \max_{\mathbf{j} \in \mathcal{R}} g(\mathbf{j})$
    end if
    Select index $\mathbf{i}_{max}$ from $\mathcal{R}$ with highest profit: $\mathbf{i}_{max} := \text{argmax}_{\mathbf{j} \in \mathcal{R}} g(\mathbf{j})$
  end while
  return $\mathcal{G}_{old}, \mathcal{E}_{old}, \eta, \mathcal{G}, \mathcal{E}$
end procedure

Choices of $\mathcal{J}$ for Computing the Profits

In this part, we consider the most common cases for the non-negative sublinear functional $\mathcal{J}$ and discuss how to compute the profits $g(\mathbf{i})$ defined as approximations of the error contributions $\mathcal{J}\big[\Delta_{\mathbf{i}}[f]\big]$. The computation of the profits is outlined as pseudocode in Algorithm 5.2. The two cases **ExVal** and **Ltwo** are particularly relevant in our adaptive SL and ML stochastic collocation approach, presented in the following sections. Let $\mathbf{I}$ be an arbitrary index set to which we add the index $\mathbf{i}$ such that we denote the new set by $\mathbf{J} := \mathbf{I} \dot{\cup} \{\mathbf{i}\}$. As discussed before, we consider only downward closed sets $\mathbf{I}, \mathbf{J}$. We denote the error contribution $\mathcal{J}\big[\Delta_{\mathbf{i}}[f]\big]$ by $e(\mathbf{i})$.

- $\mathcal{J}[\cdot] = \|\mathbb{E}[\cdot]\|_{\mathcal{W}} \rightarrow$ **ExVal**
  In this case, we are interested in the error of the expected value of the sparse grid approximation. We approximate the quantity $e(\mathbf{i})$ by approximating the expected values of the sparse grid interpolants as shown in Subsection 5.1.4. This provides the profit

$$g(\mathbf{i}) := \|\mathcal{E}_{\mathbf{J}}[f] - \mathcal{E}_{\mathbf{I}}[f]\|_{\mathcal{W}} \approx \big\|\mathbb{E}\big[\mathcal{G}_{\mathbf{J}}[f] - \mathcal{G}_{\mathbf{I}}[f]\big]\big\|_{\mathcal{W}} = \big\|\mathbb{E}\big[\Delta_{\mathbf{i}}[f]\big]\big\|_{\mathcal{W}} = e(\mathbf{i}).$$

- $\mathcal{J}[\cdot] = \|\cdot\|_{C(\Gamma,\mathcal{W})} \rightarrow$ **Max**
  This choice measures the point-wise accuracy of the approximation. We approximate the norm by evaluating the hierarchical surplus for a finite point set $\widehat{Z} \subset \Gamma$ such that the profit is given by

$$g(\mathbf{i}) := \max_{\hat{\mathbf{y}} \in \widehat{Z}} \|\Delta_{\mathbf{i}}[f](\hat{\mathbf{y}})\|_{\mathcal{W}} \approx e(\mathbf{i}).$$

  As in [81], we distinguish between nested and non-nested collocation points. For nested points, we have $\mathbb{H}_{\mathbf{I}} \subset \mathbb{H}_{\mathbf{J}}$ and the sparse grid interpolant is interpolatory. Therefore, it holds $\Delta_{\mathbf{i}}[f](\mathbf{y}) = \mathcal{G}_{\mathbf{J}}[f](\mathbf{y}) - \mathcal{G}_{\mathbf{I}}[f](\mathbf{y}) = 0$ for any $\mathbf{y} \in \mathbb{H}_{\mathbf{I}}$. However, for any newly added collocation points $\mathbf{y} \in \mathbb{H}_{\mathbf{J}} \setminus \mathbb{H}_{\mathbf{I}}$ the hierarchical surplus does not vanish which leads us to $\widehat{Z} := \mathbb{H}_{\mathbf{J}} \setminus \mathbb{H}_{\mathbf{I}}$. In addition, the point set $\widehat{Z}$ is independent of the set $\mathbf{I}$ because of the downward closed property of the index sets. Using non-nested points, we choose as point set $\widehat{Z}$ the full tensor grid $\mathbb{T}_{\mathbf{i}}$ associated to $\mathbf{i}$ since the interpolation property is not satisfied anymore and it holds in general $\mathbb{H}_{\mathbf{I}} \not\subset \mathbb{H}_{\mathbf{J}}$. When adding the index $\mathbf{i}$ to $\mathbf{I}$, the new collocation points $\mathbf{y} \in \mathbb{H}_{\mathbf{J}} \setminus \mathbb{H}_{\mathbf{I}}$ at which we have not yet evaluated the function f depend in general on the current set $\mathbf{I}$.

  An alternative procedure to construct a finite set $\widehat{Z}$ is to randomly sample a finite number of points in the stochastic space Γ according to the PDF ρ, as proposed in [49].

- $\mathcal{J}[\cdot] = \|\cdot\|_{L^2_\rho(\Gamma,\mathcal{W})} \rightarrow$ **Ltwo**
  Finally, we consider the hierarchical surplus in the typical $L^2_\rho(\Gamma, \mathcal{W})$-norm:

$$e(\mathbf{i}) = \mathbb{E}\left[\|\mathcal{G}_{\mathbf{J}}[f] - \mathcal{G}_{\mathbf{I}}[f]\|^2_{\mathcal{W}}\right]^{1/2}.$$

  As in [34], a classical Monte Carlo integration where we fix a finite point set $\widehat{Z} \subset \Gamma$ leads to the profit

$$g(\mathbf{i})^2 := \frac{1}{|\widehat{Z}|} \sum_{\hat{\mathbf{y}} \in \widehat{Z}} \|\mathcal{G}_{\mathbf{J}}[f](\hat{\mathbf{y}}) - \mathcal{G}_{\mathbf{I}}[f](\hat{\mathbf{y}})\|^2_{\mathcal{W}} \approx e(\mathbf{i})^2.$$

  Analogously to the previous **Max** case, we suggest to generate the point set $\widehat{Z}$ by randomly sampling points in the stochastic space Γ according to the given PDF ρ.

Another possibility to approximate $e(\mathbf{i})$ may be to use an appropriate quadrature rule using evaluations of the sparse grid approximations, for example the rule induced by the collocation points.

Algorithm 5.2 Algorithm to compute the profits.

1: **procedure** COMPUTEPROFITS(profitName, $e_{\mathcal{N}ew}, e_{old}, \mathcal{G}, \mathcal{G}_{old}, \mathcal{E}, \mathcal{E}_{old}$)
2: **switch** profitName **do**
3: **case ExVal**
4: $g = \|\mathcal{E} - \mathcal{E}_{old}\|_{\mathcal{W}}$
5: **case Max**
6: $g = \max \|e_{\mathcal{N}ew} - e_{old}\|_{\mathcal{W}}$
7: **case Ltwo**
8: Generate a finite sampling set $\widehat{Z} \subset \Gamma$
9: $\hat{e}$ = evaluations of $\mathcal{G}$ on each $\mathbf{y} \in \widehat{Z}$
10: $\hat{e}_{old}$ = evaluations of $\mathcal{G}_{old}$ on each $\mathbf{y} \in \widehat{Z}$
11: $g = \left(\frac{1}{|\widehat{Z}|} \sum \|\hat{e} - \hat{e}_{old}\|^2_{\mathcal{W}}\right)^{1/2}$
return g
12: **end procedure**

Modifications

We modify the original adaptive algorithm included in the MATLAB® package *Sparse Grid Kit* [7, 104] regarding two aspects.

First Modification. The first modification is realized in lines 30 - 32 in Algorithm 5.1. If the current considered index $\mathbf{i}$ with the highest profit has no admissible neighbors ($a\mathcal{N}b = \emptyset$), then the index set Λ and therefore the sparse grid and the corresponding interpolant do not change. Therefore, we want to ensure that the error indicator also remains the same and is not updated. In the original algorithm, the error indicator is updated even if there is no change in the sparse grid.

In order to illustrate this behavior, we consider the two-dimensional function

$$f(y_1, y_2) = \frac{10}{y_1^2 + 15y_2^2 + 0.3}$$

and the case that the underlying random variables $\xi_1, \xi_2 : \Theta \to \mathbb{R}$ are independent and uniformly distributed on $[-1, 1]$. We recall that $\mathbf{y} = (y_1, y_2)$ with $y_i = \xi_i(\theta) \in [-1, 1]$. We use the Clenshaw-Curtis nodes (5.11) as collocation points and **ExVal** for computing the profits since we want to approximate the expected value $\mathbb{E}[f] = \int_\Gamma f(\mathbf{y})\rho(\mathbf{y})\,\mathrm{d}\mathbf{y}$ where $\rho : \Gamma \to \mathbb{R}$ is the joint PDF of the random vector $\xi = (\xi_1, \xi_2)$.

In Figure 5.4(a), we show the error estimates of the algorithm for tolerances $TolY = 10^{-i}$ for $i = 2, 3, \ldots, 6$. For the two different tolerances 10^{-4} and 10^{-5}, the original algorithm return the same sparse grid interpolant and the same number of collocation points. However, the error indicators differ from each other. In order to understand this behavior, we take a closer look inside the algorithm executed for tolerance 10^{-5}. We start with the iteration after which the algorithm would stop for the smaller tolerance 10^{-4} which means that the error indicator η becomes smaller than 10^{-4}, but is still larger than 10^{-5}. This is equivalent to the final iteration

of the algorithm with $TolY = 10^{-4}$. The corresponding index sets $\Lambda, \mathcal{O}, \mathcal{R}$ as well as the current index with the highest profit, $(5, 7)$, are shown in Figure 5.4(c). In the next iteration step, the selected index $(5, 7)$ has no admissible neighbors and so the approximations $\mathcal{G}, \mathcal{E}$ and the index set Λ remain the same. However, the error indicator η is updated and is now smaller than 10^{-5} such that the algorithm stops. Note that the index sets $\mathcal{R}$ and $\mathcal{O}$ change since the index with the next highest profit is selected, see Figure 5.4(e).

Having implemented the modification described above, the algorithm behaves differently which is shown in Figure 5.4(b): For the tolerance 10^{-5}, we obtain a different sparse grid than for 10^{-4}. In the first iteration step in which the error indicator is smaller than 10^{-4} and larger than 10^{-5}, the considered index $(5, 7)$ has no admissible neighbors. Up to this point, both versions of the algorithm provide the same results such that the index sets shown in Figure 5.4(d) are the same as for the original version in Figure 5.4(c). Due to lines 30 - 32 in Algorithm 5.1 the error indicator does not change this time. Since the error indicator is still larger than 10^{-5}, we choose the next index with the highest profit which is $(6, 1)$. This index has admissible neighbors, $(6, 2)$ and $(7, 1)$, which we explore such that the index sets Λ and $\mathcal{O}$ enlarge, see Figure 5.4(f). Additionally, we get a new sparse grid and new approximations $\mathcal{G}, \mathcal{E}$. The error indicator is updated and becomes smaller than 10^{-5}. Consequently, the algorithm stops and returns a different sparse grid and error estimate than for $TolY = 10^{-4}$. In summary, our first modification could provide a slightly larger sparse grid for a given tolerance, but we can now guarantee that each resulting sparse grid is assigned to exactly one error indicator.

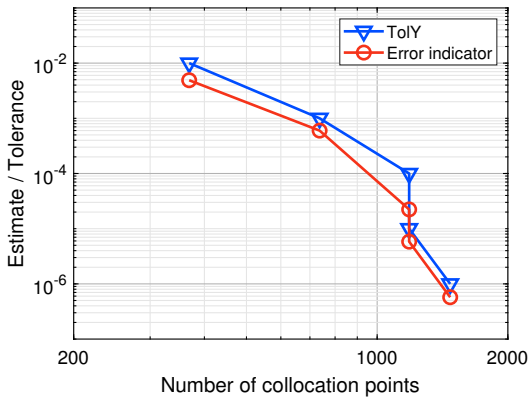
Second Modification. The second modification which we adopt from [67] is to return the sparse grid interpolant and its expected value which are computed without the hierarchical surpluses added in the last iteration step, see line 35 in Algorithm 5.1. Note that we do not adapt the profits and the error indicator. Computing the error based on these values, the error indicator η which is the highest profit, gets more practical and reasonable. Therefore, the convergence behavior in the single-level and multi-level approach can be better investigated. However, the final values would be used in most realistic applications, because they are computed with more collocation points and the required computations are already at hand.

As an example, we consider the function

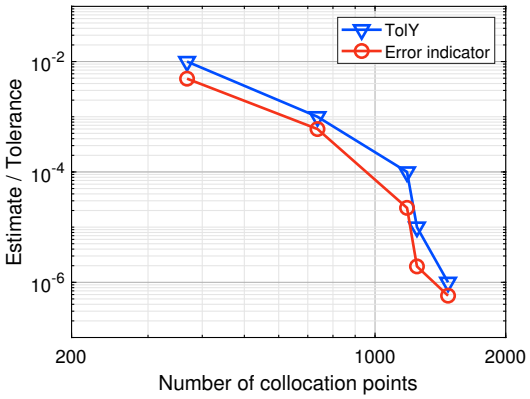
$$f(y_1, y_2) = \exp\left(-\sum_{i=1}^{2} a_i^2 (y_i - u_i)^2\right) \tag{5.12}$$

with $a = (0.3, 0.5)$ and $u = (0.2, 0.6)$. Similar to the previous example, we are interested in the expected value of f where the underlying random variables ξ_1, ξ_2 are independent of each other and uniformly distributed on $[-1, 1]$. We use the Clenshaw-Curtis nodes (5.11) as collocation points and **ExVal** for computing the profits.

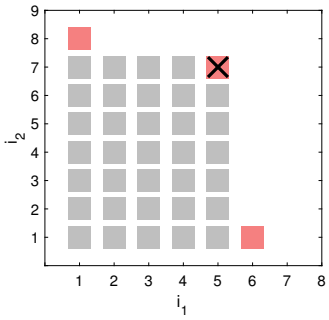
In Figure 5.5, the error estimates and errors of the expected value $\mathbb{E}[f]$ are shown for tolerances $TolY = 10^{-i}, i = 2, 3, \ldots, 7$ computed with the original algorithm as well as for the version including the second modification. In both cases, the error is always smaller than the error estimate. We compute a reference solution with an isotropic Smolyak formula of level $w = 15$. Using the original algorithm, the errors are up to two orders of magnitude smaller than the error estimates. In contrast, our second modification causes that the error and error estimate are very close to each other. They differ for each $TolY$ only by a factor smaller than 10. Therefore, the error indicator provided by our modified algorithm serves as a more accurate approximation of the error than before.



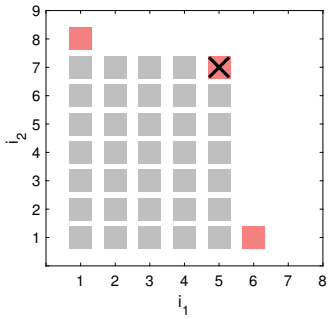
(a) Original adaptive algorithm.



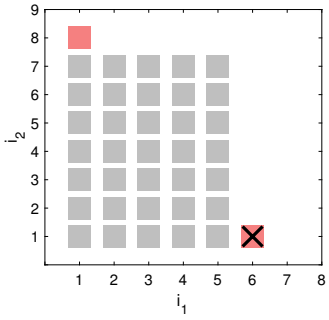
(b) Modified adaptive algorithm.



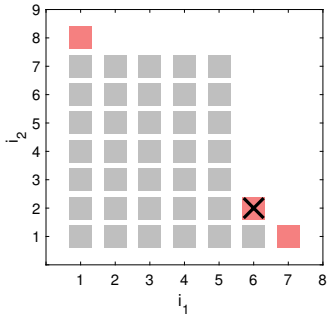
(c) Multi-index set for $TolY = 10^{-4}$.



(d) Multi-index set for $TolY = 10^{-4}$.



(e) Multi-index set for $TolY = 10^{-5}$.



(f) Multi-index set for $TolY = 10^{-5}$.

Figure 5.4: Adaptive sparse grid algorithm applied to $f(y_1, y_2) = 10/(y_1^2 + 15y_2^2 + 0.3)$ for tolerances $TolY = 10^{-i}, i = 2, 3, \ldots, 6$ using Clenshaw-Curtis nodes and **ExVal** for computing the profits, see (a) and (b). The resulting multi-index sets are shown in the subfigures (c)–(f). The grey squares are elements of the inactive set $\mathcal{O}$. The set $\mathcal{R}$ is assigned to the red squares and the cross indicates the index which has the current highest profit and will be explored in the next iteration. All squares (red and grey) represent the set Λ which is used to build the current sparse grid for the approximation. Subfigures (a), (c), (e) correspond to the original adaptive algorithm and (b), (d), (f) to the first modified version.

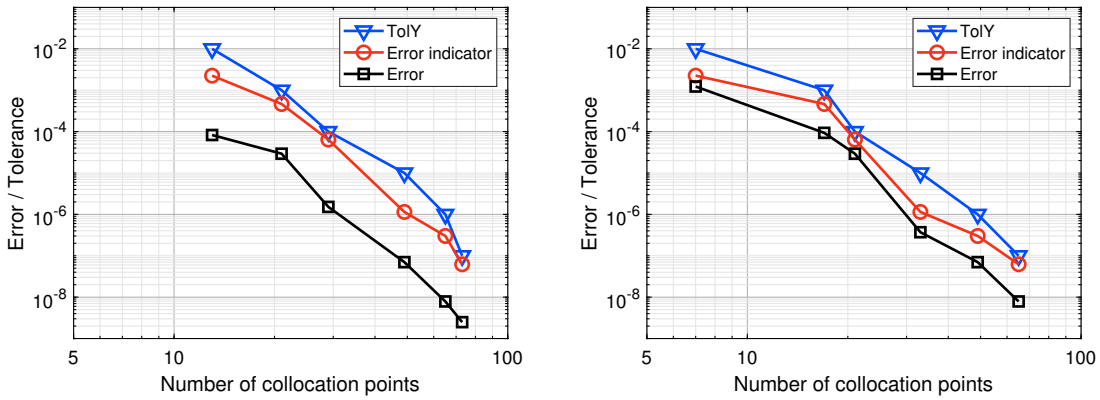


Figure 5.5: Adaptive sparse grid algorithm applied to function (5.12) for $TolY = 10^{-i}$, $i = 2, 3, \ldots, 7$ using Clenshaw-Curtis and **ExVal** for computing the profits. The left figure corresponds to the original algorithm and the right figure to the version incorporating the second modification.

5.2 Single-Level Structure

In this section, we consider the solution u of problem (3.8)–(3.12) as our function-valued QoI and present a fully adaptive single-level stochastic collocation (SLSC) method. Additionally, we analyze the error of the resulting approximation and the complexity of the method. We follow the approach given in [67, 68].

In the SC setting (3.8)–(3.12) derived in Section 3.3, the stochastic space $\Gamma = [-1, 1]^N$ is the image space of the underlying random vector $\xi : \Theta \to \Gamma$ which has a PDF $\rho : \Gamma \to \mathbb{R}_+$. The stochastic parameter $\mathbf{y}$ denotes the elements of Γ which are realizations of ξ. Note that we assumed $\rho \in L^\infty(\Gamma)$ in Assumption 3.4 which provides $C(\Gamma, \mathbb{L}^1) \subset L^2_\rho(\Gamma, \mathbb{L}^1)$, see Subsection 3.1.1.

We recall that the SC method approximates the QoI, here the full solution u, by an interpolation in the stochastic parameter space Γ based on deterministic sample points $\{\mathbf{y}_i\}_{i=1}^Q \in \Gamma$. Inserting the points into the SC problem (3.8)–(3.12) provides Q decoupled deterministic network problems of the form (2.11)–(2.15) which differ in the input data and depend only on space and time. Since the solutions $u(\mathbf{y}_i)$ of the deterministic network problems are in general unknown, we compute finite-dimensional space-time approximations $u_h(\mathbf{y}_i)$ of the solution using numerical methods (e.g. a finite volume method or an implicit box scheme as described in Section 2.4). A very common approach is to predefine a uniform discretization, see for example [80]. Similar to the Monte Carlo approach, we now assume that we have access to an adaptive black box solver which computes the numerical approximations $u_h(\mathbf{y}_i)$ by adaptively refining the spatial, temporal and model discretizations until the error estimate is less than a prescribed physical tolerance $TolH > 0$. Consequently, each sample $u_h(\mathbf{y}_i)$ has a sample-adaptive resolution in the physical space, i.e. in space, time and model hierarchy. Therefore, the samples could have different resolutions in time and space. In this case, we need to interpolate all samples onto a common fine spatial-temporal grid. The approximate solution u_h refers to the tolerance $TolH$ and still depends on the stochastic variable $\mathbf{y}$. The general aspects of adaptive strategies were described in Subsection 2.4.2 and an adaptive algorithm developed for gas transport was presented in Subsection 2.5.4. Next, we formulate the described error control of u_h in the following assumption.

Assumption 5.7 (Adaptive Physical Approximation)

(i) Let u be the stochastic entropy solution of problem (3.8)–(3.12). For $\mathbf{y} \in \Gamma$, let $u_h(\mathbf{y})$ be a physical approximation of $u(\mathbf{y})$ with a space-time-model discretization computed for a given tolerance $TolH > 0$ by an adaptive solver. Then, there exists a physical constant $c_h : \Gamma \to \mathbb{R}_+$ with $\mathbb{E}[c_h^2] < \infty$ such that the approximation $u_h(\mathbf{y})$ satisfies

$$\|u(\mathbf{y}) - u_h(\mathbf{y})\|_{\mathbb{L}^1} \leq c_h(\mathbf{y})\, TolH \tag{5.13}$$

for all $\mathbf{y} \in \Gamma$.

(ii) It holds $u_h \in C(\Gamma, \mathbb{L}^1) \subset L^2_\rho(\Gamma, \mathbb{L}^1)$.

Based on the numerical approximations $u_h(\mathbf{y}_i)$, the so-called samples, we approximate u_h by an interpolation in the stochastic space Γ. The resulting interpolant serves as an approximation to the solution u of problem (3.8)–(3.12). Let $\mathcal{A}_Q : C(\Gamma, \mathbb{L}^1) \to L^2_\rho(\Gamma, \mathbb{L}^1)$ be an interpolation operator using Q points in the stochastic space Γ which fulfills the following assumption.

Assumption 5.8 (Adaptive Stochastic Approximation)

For each stochastic tolerance $TolY > 0$, there exists a number of collocation points $Q > 0$ and a positive constant C_y such that the approximation $\mathcal{A}_Q[u_h]$ satisfies the error estimate

$$\|u_h - \mathcal{A}_Q[u_h]\|_{L^2_\rho(\Gamma,\mathbb{L}^1)} \leq C_y\, TolY. \tag{5.14}$$

The constant C_y does not depend on h meaning that it is independent of the physical tolerance $TolH$ and thus on the physical resolution of every $u_h(\mathbf{y})$.

In order to construct this operator, we apply the adaptive sparse grid algorithm introduced in the previous section and described in detail in Algorithm 5.1. We use heuristic error indicators, also called profits, which are computed for the case **Ltwo** in Algorithm 5.2 since the $L^2_\rho(\Gamma, \mathbb{L}^1)$-norm is considered in (5.14). For a given stochastic tolerance $TolY > 0$, the adaptive algorithm returns a sparse grid operator (Definition 5.3)

$$\mathcal{A}_Q := \mathcal{G}_\Lambda : C(\Gamma, \mathbb{L}^1) \to L^2_\rho(\Gamma, \mathbb{L}^1)$$

as soon as the heuristic error estimate is smaller than $TolY$. The index set Λ determines the number of collocation points Q. We denote the corresponding sparse grid by $\mathbb{H}_\mathcal{A} := \mathbb{H}_\Lambda \subset \Gamma$ with $|\mathbb{H}_\mathcal{A}| = Q$. Moreover, the algorithm provides an approximation of $\mathbb{E}[\mathcal{A}_Q[u_h]]$. In the error analysis, the overall SLSC error $(u - \mathcal{A}_Q[u_h])$ is usually investigated in the $L^2_\rho(\Gamma, \mathbb{L}^1)$-norm. Using the triangle inequality, we split the error into the sum of a physical error and a stochastic interpolation error:

$$\|u - \mathcal{A}_Q[u_h]\|_{L^2_\rho(\Gamma,\mathbb{L}^1)} \leq \underbrace{\|u - u_h\|_{L^2_\rho(\Gamma,\mathbb{L}^1)}}_{I)\text{ physical error}} + \underbrace{\|u_h - \mathcal{A}_Q[u_h]\|_{L^2_\rho(\Gamma,\mathbb{L}^1)}}_{II)\text{ stochastic error}}. \tag{5.15}$$

The physical error arises from the resolution in space, time and model hierarchy which is used for the computation of the approximations $u_h(\mathbf{y})$. The SLSC error can be bounded as follows:

Theorem 5.9

Suppose Assumptions 3.8, 5.7 and 5.8 hold for given tolerances $TolH > 0$ and $TolY > 0$. Then, the SLSC error satisfies

$$\|u - \mathcal{A}_Q[u_h]\|_{L^2_\rho(\Gamma,\mathbb{L}^1)} \leq C_h\, TolH + C_y\, TolY, \tag{5.16}$$

where the constant $C_h > 0$ is independent of $\mathbf{y}$ and the constant $C_y > 0$ independent of h.

Proof. Starting with the physical error, we integrate the square of the assumed inequality (5.13) multiplied by the PDF ρ over the stochastic space Γ and get

$$\|u - u_h\|_{L^2_\rho(\Gamma,\mathbb{L}^1)} = \mathbb{E}\left[\|u - u_h\|^2_{\mathbb{L}^1}\right]^{\frac{1}{2}} \leq C_h \, TolH$$

with $C_h := \|c_h^2\|_{L^2_\rho(\Gamma)}$ independent of the stochastic parameter $\mathbf{y} \in \Gamma$. Using inequality (5.15) and Assumption 5.8 for the stochastic error yield the stated estimate. □

Let $\epsilon > 0$ be a user-prescribed accuracy for the overall error $\|u - \mathcal{A}_Q[u_h]\|_{L^2_\rho(\Gamma,\mathbb{L}^1)}$. The common approach is to bound the physical error and the stochastic error in the splitting (5.15) equally by $\epsilon/2$. Due to Theorem 5.9, we need to bound the two terms on the right-hand side of the error estimate (5.16) by $\epsilon/2$. Therefore, the desired accuracy ϵ is achieved by the choices

$$TolH = \frac{\epsilon}{2\,C_h} \quad \text{and} \quad TolY = \frac{\epsilon}{2\,C_y}. \tag{5.17}$$

5.2.1 Complexity Analysis

The total computational cost C^{SC} of the single-level SC method is the sum of the single costs to compute the samples $u_h(\mathbf{y}_i)$ of u_h for tolerance $TolH$. Let W_h be an upper bound for the cost of each sample of u_h. Then, we get

$$C^{\mathrm{SC}} := \sum_{i=1}^{Q} cost\big(u_h(\mathbf{y}_i)\big) \leq Q \, W_h,$$

where $cost\big(u_h(\mathbf{y}_i)\big)$ denotes the cost to compute the sample $u_h(\mathbf{y}_i)$ for each collocation point $\mathbf{y}_i \in \mathbb{H}_{\mathcal{A}} \subset \Gamma$. Since the collocation points are determined using a fixed structure and not randomly, the cost C^{SC} does not depend on $\mathbf{y}$.

In order to compute the SC samples, we use the same deterministic solver as for the MC methods in Chapter 4. Therefore, we have the same upper bound W_h and suppose the same cost rate γ as in Assumption 4.7. Moreover, we require that the stochastic tolerance decrease with the number of collocation points.

Assumption 5.10

(i) Let W_h be an upper bound for the cost to compute a sample of the approximate solution u_h for tolerance $TolH > 0$. Then, there exists a constant $\gamma > 0$ such that

$$W_h \leq C_W \, TolH^{-\gamma}$$

with constant $C_W > 0$ independent of $\mathbf{y}, TolH$ and γ.

(ii) There exists a constant $\mu > 0$ such that

$$C_y \, TolY = C_I(N) \, Q^{-\mu} \, \|u_h\|_{L^2_\rho(\Gamma,\mathbb{L}^1)}.$$

The constant $C_I(N) > 0$ is independent of $\mathbf{y}, Q$ and μ.

The second assumption ensures that the stochastic approximation in (5.14) converges with respect to the number of collocation points. For elliptic PDEs, the convergence of the sparse grid approximation and the deterioration of the rate μ for increasing N was shown in [6, 81].

Now, we can estimate the computational cost C_ϵ^{SC} of the SL method required to achieve an accuracy ϵ for the SL interpolant $\mathcal{A}_Q[u_h]$. For the complexity analysis, we introduce the notation $a \lesssim b$ to denote the relation $a \leq Cb$ with a constant C independent of the physical tolerance $TolH$, the stochastic tolerance $TolY$ and the accuracy ϵ. In the case that $a \lesssim b$ and $a \gtrsim b$, we write $a \eqsim b$.

Lemma 5.11
Let Assumptions 3.8, 5.7, 5.8 and 5.10 be fulfilled. Then, for any $\epsilon < 1$, there exists a physical tolerance $TolH > 0$ and a stochastic tolerance $TolY > 0$ such that

$$\|u - \mathcal{A}_Q[u_h]\|_{L^2_\rho(\Gamma,\mathbb{L}^1)} \leq \epsilon$$

and

$$C_\epsilon^{\mathrm{SC}} \lesssim \epsilon^{-\gamma-\frac{1}{\mu}}.$$

Proof. Theorem 5.9 and Assumption 5.10 (ii) provide the error estimate

$$\|u - \mathcal{A}_Q[u_h]\|_{L^2_\rho(\Gamma,\mathbb{L}^1)} \leq C_h\, TolH + C_I\, Q^{-\mu}\, \|u_h\|_{L^2_\rho(\Gamma,\mathbb{L}^1)}.$$

In order to achieve an accuracy ϵ for the SLSC error, we bound the two terms on the right-hand side of the previous inequality by $\epsilon/2$ such that we have to choose

$$TolH = \frac{\epsilon}{2\,C_h} \quad \text{and} \quad Q = \left\lceil \left(\frac{2\,C_I\, \|u_h\|_{L^2_\rho(\Gamma,\mathbb{L}^1)}}{\epsilon} \right)^{\frac{1}{\mu}} \right\rceil.$$

Using this choice and Assumption 5.10 (i), we obtain

$$C_\epsilon^{\mathrm{SC}} \leq Q\, W_h \leq \left(\left(C_I\, \|u_h\|_{L^2_\rho(\Gamma,\mathbb{L}^1)} \right)^{\frac{1}{\mu}} \epsilon^{-\frac{1}{\mu}} + 1 \right) C_W (2\,C_h)^\gamma \epsilon^{-\gamma} \lesssim \epsilon^{-\gamma-\frac{1}{\mu}}.$$

Obviously, combining the obtained number of samples with Assumption 5.10 (ii) results in the same stochastic tolerance as stated in (5.17):

$$TolY = C_I\, C_y^{-1}\, Q^{-\mu}\, \|u_h\|_{L^2_\rho(\Gamma,\mathbb{L}^1)} = \frac{\epsilon}{2\,C_y}. \qquad \square$$

Remark 5.12
As discussed for the Monte Carlo approach in Remark 4.9, we can also choose a non-scalar function of the solution as our QoI instead of the full solution u. Some possible options are a single component of the solution, the solution integrated over the time or the spatial space and the solution evaluated at a fixed point in time $\bar{t} \in [0,T]$ or at a fixed spatial point $\bar{x} \in \bar{\Omega}_j$. We now transfer Remark 4.9 to the SC setting.

For the error of the i-th components u_i, all previous results are valid without additional assumptions because it holds

$$\|u_i(\mathbf{y}) - \mathcal{A}_Q[(u_h)_i](\mathbf{y})\|_{L^1((0,T)\times\Omega_1)\times\cdots\times L^1((0,T)\times\Omega_P)} \leq \|u(\mathbf{y}) - \mathcal{A}_Q[u_h](\mathbf{y})\|_{\mathbb{L}^1}$$

for $\mathbf{y} \in \Gamma$. The same argument applies to the quantities $\sum_{j=1}^P \int_{[0,T]} u^{(j)}(\cdot,t,\cdot)\,\mathrm{d}t$ and $\int_{\Omega_j} u^{(j)}(x,\cdot,\cdot)\,\mathrm{d}x$ using the linearity of the interpolation operator $\mathcal{A}_Q$. Analyzing the stochastic collocation error of $u^{(j)}(x,\bar{t},\mathbf{y})$ for a fixed $\bar{t} \in [0,T]$ requires that the pointwise evaluation in

time of the approximate solution is defined for all $\mathbf{y} \in \Gamma$. *Therefore, we assume*

$$u_h \in C\big(\Gamma, H^1((0,T), L^1(\Omega_1, \mathbb{R}^d)) \times \cdots \times H^1((0,T), L^1(\Omega_P, \mathbb{R}^d))\big).$$

Assumptions 5.7, 5.8 and 5.10 have to hold for the considered quantity $u_h^{(j)}(x, \bar{t}, \mathbf{y})$. *Then, the convergence and complexity analysis of the error*

$$\|u^{(j)}(\cdot, \bar{t}, \cdot) - \mathcal{A}_Q[u_h^{(j)}(\cdot, \bar{t}, \cdot)]\|_{L^2_\rho(\Gamma, L^1(\Omega_j, \mathbb{R}^d))}$$

can be analogously performed as before. In the case of the approximate solution $u_h^{(j)}(\bar{x}, t, \mathbf{y})$ *evaluated at a fixed spatial point, we need to assume that* u *and* u_h *belong to the space*

$$C\big(\Gamma, L^1((0,T), H^1(\Omega_1, \mathbb{R}^d)) \times \cdots \times L^1((0,T), H^1(\Omega_P, \mathbb{R}^d))\big).$$

If Assumptions 5.7, 5.8 and 5.10 are fulfilled for $u_h^{(j)}(\bar{x}, t, \mathbf{y})$, *we can analyze the error*

$$\|u^{(j)}(\bar{x}, \cdot, \cdot) - \mathcal{A}_Q[u_h^{(j)}(\bar{x}, \cdot, \cdot)]\|_{L^2_\rho(\Gamma, L^1((0,T), \mathbb{R}^d))}$$

as shown in this section.

5.3 Multi-Level Structure

The SC method is now extended to a multi-level structure in order to further reduce the computational complexity. First of all, the multi-level strategy was developed independently in the context of MC methods in [42, 53]. Afterwards, the ML extension was transferred to stochastic collocation methods and applied to elliptic PDEs with random data using a hierarchy based on the spatial uniform mesh and the number of collocation points [106]. Therefore, the curse of dimensionality could be further delayed. Recently, Lang et al. [67] developed a novel multi-level method by combining adaptive sparse grids in the stochastic space and adaptive mesh refinements for the spatial approximation with the multi-level structure. In a collaboration between the author, Lang and Domschke in [68], the fully adaptive MLSC approach has already been extended to uncertain gas transport problems considering functionals of the solution. The hierarchy of sample-dependent spatial discretizations in the elliptic case in [67] is replaced by a more complex hierarchy of discretizations in space, time and model hierarchy. Proceeding similar to the extension in [68], we extend the fully adaptive MLSC approach to the solution of general hyperbolic PDEs with random data on networks.

The main idea of the multi-level structure is to couple physical approximate solutions of different accuracies with different stochastic interpolation operators such that less accurate interpolation operators are used on more accurate physical approximate solutions. This means that we need to compute less samples on fine meshes and with high fidelity models. Accordingly, the most accurate interpolation operator would be used on the least accurate physical approximation. Since the computational cost for computing one sample increases with the required accuracy of the sample, we expect to obtain a reduction of the computational cost.

Let $\{TolH_k\}_{k=0}^K$ be a sequence of physical tolerances with

$$1 \geq TolH_0 > TolH_1 > \cdots > TolH_K > 0$$

and $K \in \mathbb{N}_+$. Each index $k \in \{0, \ldots, K\}$ refers to a numerical approximation u_{h_k} of the solution u which is constructed as in the previous section on the single-level structure: The

physical approximation $u_{h_k}(\mathbf{y})$ is computed with a sample-dependent resolution in space, time and model hierarchy controlled by the tolerance $TolH_k$. This error-controlled behavior is realized by an adaptive black box solver which uses numerical methods to compute each sample $u_{h_k}(\mathbf{y})$ and which adaptively refines the spatial, temporal and model discretizations until the error estimate is less than the prescribed tolerance $TolH_k$. Since the physical tolerances are constructed as a decreasing sequence, the space-time-model resolution of a fixed sample $u_{h_k}(\mathbf{y})$ gets finer with increasing k. Now, we assume the following property of the error control of u_{h_k}.

Assumption 5.13 (Sequence of Adaptive Physical Approximations)

(i) Let u be the stochastic entropy solution of problem (3.8)–(3.12). For $\mathbf{y} \in \Gamma$, let $u_{h_k}(\mathbf{y})$ be a physical approximation computed by an adaptive solver for a given tolerance $TolH_k > 0$. Then, for all $k = 0, \dots, K$ there exists a physical constant $c_k : \Gamma \to \mathbb{R}_+$ with $\mathbb{E}[c_k^2] < \infty$ such that the approximate solution $u_{h_k}(\mathbf{y})$ satisfies

$$\|u(\mathbf{y}) - u_{h_k}(\mathbf{y})\|_{\mathbb{L}^1} \leq c_k(\mathbf{y})\, TolH_k \tag{5.18}$$

for all $\mathbf{y} \in \Gamma$.

(ii) It holds $u_{h_k} \in C(\Gamma, \mathbb{L}^1) \subset L^2_\rho(\Gamma, \mathbb{L}^1)$ for $k = 0, \dots, K$.

We rewrite the approximate solution u_{h_K} with the highest physical accuracy as the telescoping sum of approximations with lower physical accuracies

$$u_{h_K} = \sum_{k=0}^{K} u_{h_k} - u_{h_{k-1}},$$

where we set $u_{h_{-1}} = 0$. Instead of applying a stochastic interpolation operator only to the most accurate approximation u_{h_K}, we interpolate each term $(u_{h_k} - u_{h_{k-1}})$ of the telescoping sum separately. To this end, we introduce a second sequence of (stochastic) tolerances $\{TolY_k\}_{k=0}^K$ where each $TolY_k$ relates to an interpolation operator $\mathcal{A}_{Q_k} : C(\Gamma, \mathbb{L}^1) \to L^2_\rho(\Gamma, \mathbb{L}^1)$ which uses Q_k points and fulfills the following assumption.

Assumption 5.14 (Sequence of Adaptive Stochastic Approximations)

There exists a sequence of number of collocation points $\{Q_k\}_{k=0,\dots,K}$ and a positive constant C_Y which does not depend on k such that

$$\|(u_{h_k} - u_{h_{k-1}}) - \mathcal{A}_{Q_k}[u_{h_k} - u_{h_{k-1}}]\|_{L^2_\rho(\Gamma,\mathbb{L}^1)} \leq C_Y\, TolY_k \tag{5.19}$$

for $k = 0, \dots, K$.

As in the single-level approach, we apply the adaptive sparse grid algorithm presented in Subsection 5.1.6 in order to construct these interpolation operators. Since the algorithm computes the approximation $\mathcal{A}_{Q_k}[u_{h_k} - u_{h_{k-1}}]$ by successively adding collocation points until a heuristic error estimate is lower than a given stochastic tolerance $TolY_k$, the previous assumption is justified. Considering these interpolation operators together with the telescoping sum, we can define the multi-level interpolant $u_K^{(\mathrm{ML})}$.

Definition 5.15 (Multi-Level Interpolant)

The multi-level interpolant of u is defined as

$$u_K^{(\mathrm{ML})} := \sum_{k=0}^{K} \mathcal{A}_{Q_k}[u_{h_k} - u_{h_{k-1}}]$$

with $u_{h_{-1}} := 0$.

Under Assumption 5.13, we obtain $u_K^{(\mathrm{ML})} \in L^2_\rho(\Gamma, \mathbb{L}^1)$. Integrating the square of inequality (5.18) with respect to the PDF ρ, we get the error estimate

$$\|u - u_{h_k}\|_{L^2_\rho(\Gamma,\mathbb{L}^1)} \leq C_H \, TolH_k, \qquad k = 0, \ldots, K, \tag{5.20}$$

with the constant

$$C_H = \max_{k=0,\ldots,K} \mathbb{E}\left[c_k^2\right]^{1/2} = \max_{k=0,\ldots,K} \|c_k\|_{L^2_\rho(\Gamma)}$$

which is independent of $\mathbf{y}$ and k. This estimate provides $\|u_{h_k} - u_{h_{k-1}}\|_{L^2_\rho(\Gamma,\mathbb{L}^1)} \leq 2C_H TolH_{k-1}$ which decreases for $k \to \infty$. Therefore, we expect to need less accurate interpolation operators for more accurate physical approximations which means that we need to compute less samples on fine meshes and with high fidelity models. This is the main reason to use the multi-level structure for stochastic collocation methods. In order to analyze the convergence of the ML interpolant $u_K^{(\mathrm{ML})}$ to the solution u, we split the multi-level stochastic collocation (MLSC) error into a deterministic physical and a stochastic interpolation error by using the triangle inequality:

$$\|u - u_K^{(\mathrm{ML})}\|_{L^2_\rho(\Gamma,\mathbb{L}^1)} \leq \underbrace{\|u - u_{h_K}\|_{L^2_\rho(\Gamma,\mathbb{L}^1)}}_{I) \text{ physical error}} + \underbrace{\|u_{h_K} - u_K^{(\mathrm{ML})}\|_{L^2_\rho(\Gamma,\mathbb{L}^1)}}_{II) \text{ stochastic error}}. \tag{5.21}$$

Theorem 5.16
Suppose Assumptions 3.8, 5.13 and 5.14 are fulfilled. Then, it holds

$$\|u - u_K^{(\mathrm{ML})}\|_{L^2_\rho(\Gamma,\mathbb{L}^1)} \leq C_H \, TolH_K + C_Y \sum_{k=0}^{K} TolY_k, \tag{5.22}$$

where the constants C_H, C_Y are positive and independent of k and $\mathbf{y}$.

Proof. Due to the error splitting in (5.21), it is sufficient to bound the physical and the stochastic error term separately. According to the inequality (5.20), the physical error is bounded by

$$\|u - u_{h_K}\|_{L^2_\rho(\Gamma,\mathbb{L}^1)} \leq C_H \, TolH_K$$

with $C_H = \max_{k=0,\ldots,K} \|c_k\|_{L^2_\rho(\Gamma)}$. The stochastic interpolation error which is the second term on the right-hand side of (5.21) can be rewritten by inserting Definition 5.15 of the ML interpolant and the identity $u_{h_K} = \sum_{k=0}^{K}(u_{h_k} - u_{h_{k-1}})$. Using additionally the triangle inequality and Assumption 5.14, we obtain

$$\begin{aligned}
\|u_{h_K} - u_K^{(\mathrm{ML})}\|_{L^2_\rho(\Gamma,\mathbb{L}^1)} &= \left\| u_{h_K} - \sum_{k=0}^{K} \mathcal{A}_{Q_k}[u_{h_k} - u_{h_{k-1}}] \right\|_{L^2_\rho(\Gamma,\mathbb{L}^1)} \\
&= \left\| \sum_{k=0}^{K} (u_{h_k} - u_{h_{k-1}}) - \mathcal{A}_{Q_k}[u_{h_k} - u_{h_{k-1}}] \right\|_{L^2_\rho(\Gamma,\mathbb{L}^1)} \\
&\leq \sum_{k=0}^{K} \|u_{h_k} - u_{h_{k-1}} - \mathcal{A}_{Q_k}[u_{h_k} - u_{h_{k-1}}]\|_{L^2_\rho(\Gamma,\mathbb{L}^1)} \leq C_Y \sum_{k=0}^{K} TolY_k
\end{aligned}$$

with constant $C_Y > 0$ independent of k and $\mathbf{y}$. □

In order to achieve that the stochastic error has the same size as the physical error, we choose the stochastic tolerances $\{TolY_k\}$ depending on the physical tolerances $\{TolH_k\}$. One straightforward approach is to balance both error terms, see [67, 106]. Due to the error estimate (5.22) in the previous theorem, we choose

$$TolY_k = C_H\, TolH_K\, (C_Y(K+1))^{-1} \qquad \text{for } k = 0, \dots, K.$$

From this, it follows $\|u - u_K^{(\mathrm{ML})}\|_{L^2_\rho(\Gamma,\mathbb{L}^1)} \leq 2\, C_H\, TolH_K$ which ensures that the MLSC error converges for $TolH_K \to 0$. If we now want to achieve a user-prescribed accuracy $\epsilon > 0$ for the MLSC error, both terms on the right-hand side of estimate (5.22) have to be bounded by $\epsilon/2$. For the considered approach, we obtain the physical tolerance $TolH_K = \epsilon/(2\, C_H)$ with the tolerances $TolY_k$ defined as above.

Another strategy is to choose the tolerances $TolY_k$ on each level so that the computational cost of the method is minimized and the MLSC error satisfies the desired accuracy ϵ [18, 67, 106]. In the following, we concentrate on this cost minimization approach.

5.3.1 Complexity Analysis

In the following, we investigate in detail the complexity of the MLSC method and we mainly follow the analysis given in [67]. The complexity analysis of the adaptive MLMC method in Section 4.2 was performed in a similar way.

Let W_k be an upper bound for the cost of a single sample $u_{h_k}(\mathbf{y}_i)$ for $k = 0, \dots, K$. We can bound the total computational cost of the multi-level approximation by

$$C^{\mathrm{MLSC}} = \sum_{k=0}^{K} \sum_{i=1}^{Q_k} cost\big(u_{h_k}(\mathbf{y}_i)\big) + cost\big(u_{h_{k-1}}(\mathbf{y}_i)\big) \leq \sum_{k=0}^{K} Q_k (W_k + W_{k-1}) \tag{5.23}$$

with $W_{-1} := 0$.

Similar to Assumption 5.10, we need to link the stochastic tolerances with the number of collocation points. Thus, we suppose the following assumption such that the stochastic approximations in (5.19) converge with respect to the number of collocation points.

Assumption 5.17
There exist positive constants $C_I(N)$ and μ such that for all $k = 0, \dots, K$ it holds

$$C_Y\, TolY_k = C_I(N)\, Q_k^{-\mu}\, TolH_{k-1}$$

with $TolH_{-1} := \|u_{h_0}\|_{L^2_\rho(\Gamma,\mathbb{L}^1)}$. The constant $C_I(N)$ is independent of $\mathbf{y}, k$ and μ.

Now, we can estimate the computational cost $C_\epsilon^{\mathrm{MLSC}}$ of the MLSC method required to achieve an accuracy ϵ for the multi-level approximation $u_K^{(\mathrm{ML})}$. Note that the physical tolerance $TolH_k$ needs to converge to zero for $k \to \infty$.

Theorem 5.18
Let $\{TolH_k\}_{k=0,1,\dots}$ be a strictly decreasing null sequence of physical tolerances $TolH_k \in (0,1]$. Further, suppose Assumptions 3.8, 5.13, 5.14 and 5.17 hold. Then, for any $\epsilon > 0$, there exists a number $K = K(\epsilon) \in \mathbb{N}_+$ and a sequence of tolerances $\{TolY_k\}_{k=0,\dots,K}$ in (5.19) such that

$$\|u - u_K^{(\mathrm{ML})}\|_{L^2_\rho(\Gamma,\mathbb{L}^1)} \leq \epsilon$$

and

$$C_\epsilon^{\mathrm{MLSC}} \leq \hat{C}\,\epsilon^{-\frac{1}{\mu}} F(\mu)^{\frac{\mu}{\mu+1}} + \sum_{k=0}^{K} G_k$$

with $\hat{C} = (2\,C_I)^{\frac{1}{\mu}}$ and

$$F(\mu) = \sum_{k=0}^{K} G_k^{\frac{\mu}{\mu+1}} \left(TolH_{k-1}\right)^{\frac{1}{\mu+1}}, \qquad G_k = W_k + W_{k-1}.$$

The optimal stochastic tolerances $TolY_k$ are given by

$$TolY_k = \epsilon \left(2\,C_Y F(\mu)\right)^{-1} G_k^{\frac{\mu}{\mu+1}} \left(TolH_{k-1}\right)^{\frac{1}{\mu+1}}.$$

Proof. This proof follows the same approach as the proof of Theorem 4.14. Considering the MLSC error, the error estimate (5.22) and Assumption 5.17 provide

$$\|u - u_K^{(\mathrm{ML})}\|_{L^2_\rho(\Gamma,\mathbb{L}^1)} \leq C_H\,TolH_K + C_I \sum_{k=0}^{K} Q_k^{-\mu} TolH_{k-1}.$$

In order to achieve a total accuracy $\epsilon > 0$, we ensure that the physical discretization and the stochastic interpolation error contribution are bounded by $\epsilon/2$. First, we choose a suitable integer $K \in \mathbb{N}_+$ such that $C_H\,TolH_K \leq \epsilon/2$ is satisfied. Note that this choice defines the number $K = K(\epsilon)$ as a function of ϵ and can be always made since $\{TolH_k\}_{k=0,1,\dots}$ is a null sequence. Concerning the stochastic error, we require

$$C_I \sum_{k=0}^{K} Q_k^{-\mu}\,TolH_{k-1} = \frac{\epsilon}{2}$$

and want to determine the number of samples in such a way that the upper bound (5.23) of the total cost of the MLSC approximation is minimized. Consequently, we consider the following minimization problem

$$\begin{aligned} \min_{Q_0,\dots,Q_K} \quad & \sum_{k=0}^{K} Q_k\,(W_k + W_{k-1}) \\ \text{s.t.} \quad & C_I \sum_{k=0}^{K} Q_k^{-\mu}\,TolH_{k-1} = \frac{\epsilon}{2}. \end{aligned}$$

We solve this problem by the Lagrangian multiplier method considering Q_k as a continuous variable. The Lagrangian function is given by

$$\mathcal{L}(Q_0,\dots,Q_K,\alpha) = \sum_{k=0}^{K} Q_k\,(W_k + W_{k-1}) + \alpha\left(C_I \sum_{k=0}^{K} Q_k^{-\mu}\,TolH_{k-1} - \frac{\epsilon}{2}\right)$$

with the Lagrangian multiplier $\alpha \in \mathbb{R}$. The next step is to find an extremum of the function by imposing the condition $\nabla \mathcal{L}(Q_0, \dots, Q_K, \alpha) \stackrel{!}{=} 0$. This provides the equations

$$\frac{\partial \mathcal{L}}{\partial Q_k} = W_k + W_{k-1} - \alpha\, \mu\, C_I\, Q_k^{-\mu-1}\, TolH_{k-1} = 0 \qquad k = 0, \dots, K, \tag{5.24}$$

$$\frac{\partial \mathcal{L}}{\partial \alpha} = C_I \sum_{k=0}^{K} Q_k^{-\mu}\, TolH_{k-1} - \frac{\epsilon}{2} = 0. \tag{5.25}$$

Now, we solve the equation (5.24) for Q_k:

$$Q_k = \left(\frac{\alpha\, \mu\, C_I\, TolH_{k-1}}{W_k + W_{k-1}} \right)^{\frac{1}{\mu+1}}. \tag{5.26}$$

Inserting this formula into (5.25) and solving the resulting equation for α, we get

$$\alpha = \mu^{-1}\, C_I^{-\frac{1}{\mu}} \left(2\, \epsilon^{-1} F(\mu)\right)^{\frac{\mu+1}{\mu}}$$

with

$$F(\mu) := \sum_{k=0}^{K} G_k^{\frac{\mu}{\mu+1}} \left(TolH_{k-1}\right)^{\frac{1}{\mu+1}} \quad \text{and} \quad G_k := W_k + W_{k-1}.$$

Substituting the formula for the parameter α into equation (5.26) provides the optimal number of samples

$$Q_k = C_I^{\frac{1}{\mu}} \left(2\, \epsilon^{-1} F(\mu)\right)^{\frac{1}{\mu}} \left(TolH_{k-1}\, G_k^{-1}\right)^{\frac{1}{\mu+1}}$$

which we round up to the next integer Q_k^*. Using the chosen number of samples which satisfies $Q_k^* \leq Q_k + 1$, we obtain for the ϵ-cost the estimation

$$\begin{aligned} C_\epsilon^{\text{MLSC}} &\leq \sum_{k=0}^{K} (Q_k + 1)(W_k + W_{k-1}) \\ &= \sum_{k=0}^{K} \left(C_I^{\frac{1}{\mu}} \left(2\, \epsilon^{-1}\, F(\mu)\right)^{\frac{1}{\mu}} \left(TolH_{k-1}\, G_k^{-1}\right)^{\frac{1}{\mu+1}} + 1 \right) G_k \\ &= \left(2\, C_I\, \epsilon^{-1}\right)^{\frac{1}{\mu}} F(\mu)^{\frac{\mu}{\mu+1}} + \sum_{k=0}^{K} G_k. \end{aligned}$$

Combining the obtained optimal number of samples with Assumption 5.17, the optimal choice of tolerances $TolY_k$ is determined by

$$TolY_k = C_I\, C_Y^{-1}\, Q_k^{-\mu}\, TolH_{k-1} = \epsilon \big(2\, C_Y\, F(\mu)\big)^{-1} G_k(\mu)^{\frac{\mu}{\mu+1}} \left(TolH_{k-1}\right)^{\frac{1}{\mu+1}} > 0$$

which guarantees the accuracy $\epsilon/2$ of the stochastic error since

$$\sum_{k=0}^{K} C_Y\, TolY_k = \frac{\epsilon}{2} F(\mu)^{-1} \underbrace{\sum_{k=0}^{K} G_k(\mu)^{\frac{\mu}{\mu+1}} \left(TolH_{k-1}\right)^{\frac{1}{\mu+1}}}_{=F(\mu)} = \frac{\epsilon}{2}. \qquad \square$$

We assume that the upper bound W_k is bounded in the same way as in Assumption 5.10 in the single-level approach such that we obtain a slightly modified formula for the stochastic tolerances $TolY$ on each level which we use in the implementation of the MLSC method.

Assumption 5.19
Let W_k be an upper bound for the cost to compute a sample of the approximate solution u_{h_k} for tolerance $TolH_k > 0$. There exists a constant $\gamma > 0$ with

$$W_k \leq C_W \, TolH_k^{-\gamma},$$

where C_W is a positive constant independent of $\mathbf{y}, k$ and γ.

Corollary 5.20
Let $\{TolH_k\}_{k=0,1,\dots}$ be a strictly decreasing null sequence of physical tolerances $TolH_k \in (0,1]$. Further, let Assumptions 3.8, 5.13, 5.14, 5.17 and 5.19 be fulfilled. Then, for any $\epsilon > 0$, there exists a number $K = K(\epsilon) \in \mathbb{N}_+$ and a sequence $\{TolY_k\}_{k=0,\dots,K}$ in (5.19) such that

$$\|u - u_K^{(\mathrm{ML})}\|_{L^2_\rho(\Gamma,\mathbb{L}^1)} \leq \epsilon$$

and

$$C_\epsilon^{\mathrm{MLSC}} \leq \hat{C}\, \epsilon^{-\frac{1}{\mu}} \, F(\mu,\gamma)^{\frac{\mu+1}{\mu}} + C_W \sum_{k=0}^{K} G_k(\gamma)$$

with $\hat{C} := C_W (2\,C_I)^{\frac{1}{\mu}}$ and

$$F(\mu,\gamma) = \sum_{k=0}^{K} G_k(\gamma)^{\frac{\mu}{\mu+1}} \big(TolH_{k-1}\big)^{\frac{1}{\mu+1}} \quad \text{with} \quad G_k(\gamma) = TolH_k^{-\gamma} + (1-\delta_{k0})\, TolH_{k-1}^{-\gamma}. \tag{5.27}$$

As usual, the Kronecker delta δ_{k0} is defined as $\delta_{k0} = 1$ for $k = 0$ and $\delta_{k0} = 0$ otherwise. The optimal stochastic tolerances $TolY_k$ are given by

$$TolY_k = \epsilon \big(2\, C_Y \, F(\mu,\gamma)\big)^{-1} G_k(\gamma)^{\frac{\mu}{\mu+1}} \big(TolH_{k-1}\big)^{\frac{1}{\mu+1}}. \tag{5.28}$$

Proof. We prove the stated results analogously to the proof of Theorem 5.18. According to Assumption 5.19, the considered optimization problem modifies to

$$\begin{aligned} \min_{Q_0,\dots,Q_K} \quad & C_W \sum_{k=0}^{K} Q_k \left(TolH_k^{-\gamma} + (1-\delta_{k0})\, TolH_{k-1}^{-\gamma} \right) \\ \text{s.t.} \quad & C_I \sum_{k=0}^{K} Q_k^{-\mu}\, TolH_{k-1} = \frac{\epsilon}{2}. \end{aligned}$$

□

Here, the use of the Kronecker delta is essential because on the coarsest level $k = 0$ there are only samples of u_{h_0} and the overall cost on this level is bounded by $C_W\, Q_0\, TolH_0^{-\gamma}$. Note that we define $TolH_{-1} = \|u_{h_0}\|_{L^2_\rho(\Gamma,\mathbb{L}^1)}$ in Assumption 5.17 and therefore we cannot handle this issue analogously to the previous case where we set $W_{-1} = 0$.

Next, we focus on physical tolerances generated via a geometric design $TolH_k = q^k\, TolH_0$ using a positive reduction factor $q < 1$. With the previous results, we can now analyze the complexity of the MLSC method which is smaller than the complexity of the SLSC method.

Theorem 5.21
Let a null sequence of physical tolerances $\{TolH_k\}_{k=0,1,\dots}$ *be given by* $TolH_k = q^k\, TolH_0$ *with* $TolH_0 \in (0,1]$ *and a reduction factor* $q \in (0,1)$. *Further, let Assumptions 3.8, 5.13, 5.14, 5.17 and 5.19 be fulfilled. Then, for any* $\epsilon \in (0, 1/e]$, *there exists a number* $K = K(\epsilon) \in \mathbb{N}_+$ *and a sequence* $\{TolY_k\}_{k=0}^K$ *such that*

$$\|u - u_K^{(\mathrm{ML})}\|_{L^2_\rho(\Gamma,\mathbb{L}^1)} \leq \epsilon$$

and

$$C_\epsilon^{\mathrm{MLSC}} \lesssim \begin{cases} \epsilon^{-\frac{1}{\mu}} & if\ \gamma\mu < 1, \\ \epsilon^{-\frac{1}{\mu}} |\log \epsilon|^{1+\frac{1}{\mu}} & if\ \gamma\mu = 1, \\ \epsilon^{-\gamma} & if\ \gamma\mu > 1. \end{cases}$$

Proof. We transferred the proof of [106, Theorem 4.2] to our fully adaptive setting. This proof is very similar to the proof of Theorem 4.18 for adaptive MLMC methods. Corollary 5.20 implies the first statement of this theorem including the condition $C_H\, TolH_K = C_H\, q^K\, TolH_0 \leq \epsilon/2$ for K. We choose

$$K := \left\lceil \log_q \left(\frac{\epsilon}{2\, C_H\, TolH_0} \right) \right\rceil + K_0 \tag{5.29}$$

with the smallest possible constant $K_0 \in \mathbb{N}$ such that $K \in \mathbb{N}_+$. It follows

$$C_H\, TolH_K \leq q^{K_0+1}\epsilon/2 \leq \epsilon/2$$

since $q^{K_0+1} < 1$ and $K \leq \log_q\big(\epsilon/(2\, C_H\, TolH_0)\big) + 1 + K_0$. In addition, Corollary 5.20 provides the estimate

$$C_\epsilon^{\mathrm{MLSC}} \leq \underbrace{C_W (2\, C_I)^{\frac{1}{\mu}}\, \epsilon^{-\frac{1}{\mu}} \Big(F(\mu,\gamma) \Big)^{\frac{\mu+1}{\mu}}}_{=:\ (I)} + \underbrace{C_W \sum_{k=0}^{K} G_k(\gamma)}_{=:\ (II)}, \tag{5.30}$$

where we set

$$F(\mu,\gamma) = \sum_{k=0}^{K} G_k(\gamma)^{\frac{\mu}{\mu+1}} \big(TolH_{k-1} \big)^{\frac{1}{\mu+1}}$$

with

$$G_k(\gamma) = TolH_k^{-\gamma} + (1 - \delta_{k0})\, TolH_{k-1}^{-\gamma}.$$

In order to estimate the complexity $C_\epsilon^{\mathrm{MLSC}}$ further, we need the following property of a geometric sum with $q \in (0,1)$ and $\alpha > 0$:

$$\sum_{k=0}^{K} q^{\alpha k} = \frac{1 - q^{\alpha(K+1)}}{1 - q^\alpha} < \frac{1}{1 - q^\alpha} \tag{5.31}$$

since $q^\alpha < 1$. Together with the formula (5.29) for K, it follows

$$\begin{aligned} \sum_{k=0}^{K} q^{-\alpha k} = \sum_{k=0}^{K} \big(q^\alpha\big)^{k-K} = q^{-\alpha K} \sum_{k=0}^{K} \big(q^\alpha\big)^k < \frac{q^{-\alpha K}}{1 - q^\alpha} \\ \lesssim q^{-\alpha\,(\log_q(\epsilon/(2 C_H TolH_0)) + 1 + K_0)} \eqsim \epsilon^{-\alpha}. \end{aligned} \tag{5.32}$$

First, we consider the second term (II) on the right-hand side of (5.30). The assumed geometric design of $TolH_k = q^k\, TolH_0$ and inequality (5.32) yield

$$(II) = C_W \sum_{k=0}^{K} TolH_k^{-\gamma} + (1-\delta_{k0})\, TolH_{k-1}^{-\gamma} = C_W\, TolH_0^{-\gamma} \sum_{k=0}^{K} q^{-\gamma k} + (1-\delta_{k0})\, q^{-\gamma(k-1)}$$
$$\leq C_W\, TolH_0^{-\gamma}\big(1 + (1-\delta_{k0})\, q^{\gamma}\big) \sum_{k=0}^{K} q^{-\gamma k} \lesssim \frac{q^{-\gamma K}}{1-q^{\gamma}} \eqsim \epsilon^{-\gamma}.$$

For the sum $F(\mu,\gamma)$ in the estimation (5.30) of the total cost, the relation $TolH_k = q^k\, TolH_0$ leads to

$$\begin{aligned} F(\mu,\gamma) &= \sum_{k=0}^{K} \Big(TolH_k^{-\gamma} + (1-\delta_{k0})\, TolH_{k-1}^{-\gamma} \Big)^{\frac{\mu}{\mu+1}} \left(TolH_{k-1}\right)^{\frac{1}{\mu+1}} \\ &= TolH_0^{-\frac{\gamma\mu}{\mu+1}} TolH_{-1}^{\frac{1}{\mu+1}} + TolH_0^{\frac{1-\gamma\mu}{\mu+1}} \sum_{k=1}^{K} \Big(q^{-\gamma k} + q^{-\gamma(k-1)} \Big)^{\frac{\mu}{\mu+1}} q^{\frac{k-1}{\mu+1}} \\ &= TolH_0^{-\frac{\gamma\mu}{\mu+1}} TolH_{-1}^{\frac{1}{\mu+1}} + TolH_0^{\frac{1-\gamma\mu}{\mu+1}} \big(1+q^{\gamma}\big)^{\frac{\mu}{\mu+1}} q^{-\frac{1}{\mu}} \sum_{k=1}^{K} q^{-k\frac{\gamma\mu}{\mu+1}} q^{\frac{k}{\mu+1}} \\ &\lesssim 1 + \sum_{k=1}^{K} q^{k\frac{1-\gamma\mu}{\mu+1}} \eqsim \sum_{k=0}^{K} q^{k\frac{1-\gamma\mu}{\mu+1}} \end{aligned}$$

with $TolH_{-1}$ being constant. Therefore, we get for the term (I) in (5.30) the estimate:

$$(I) = C_W (2\, C_I)^{\frac{1}{\mu}}\, \epsilon^{-\frac{1}{\mu}} F(\mu,\gamma)^{\frac{\mu+1}{\mu}} \lesssim \epsilon^{-\frac{1}{\mu}} \left(\sum_{k=0}^{K} q^{k\frac{1-\gamma\mu}{\mu+1}} \right)^{\frac{\mu+1}{\mu}}. \tag{5.33}$$

Now, we consider the geometric sum $(G) := \sum_{k=0}^{K} \left(q^{\frac{1-\gamma\mu}{\mu+1}} \right)^k$ for the following three different cases:

(i) $1-\gamma\mu > 0$:
For this case, inequality (5.31) provides that the geometric sum (G) is bounded by a constant independent of K. Therefore, we have $(I) \lesssim \epsilon^{-\frac{1}{\mu}}$ and it follows

$$C_{\epsilon}^{\mathrm{MLSC}} \lesssim \epsilon^{-\frac{1}{\mu}} + \epsilon^{-\gamma} \lesssim \epsilon^{-\frac{1}{\mu}}$$

since $\epsilon < 1$ and $1/\mu > \gamma$.

(ii) $1-\gamma\mu = 0$:
Inserting the formula (5.29) for K gives

$$(G) = \sum_{k=0}^{K} \left(q^{\frac{1-\gamma\mu}{\mu+1}} \right)^k = K + 1 \leq \log_q \left(\frac{\epsilon}{2\, C_H\, TolH_0} \right) + 2 + K_0 \lesssim |\log(\epsilon)|$$

because it holds $|\log(\epsilon)| \geq 1$ due to $\epsilon \leq e^{-1}$. From this, it follows

$$(I) \lesssim \epsilon^{-\frac{1}{\mu}} |\log(\epsilon)|^{\frac{\mu+1}{\mu}}$$

and finally

$$C_\epsilon^{\mathrm{MLSC}} \lesssim \epsilon^{-\frac{1}{\mu}} |\log(\epsilon)|^{\frac{\mu+1}{\mu}} + \epsilon^{-\gamma} \lesssim \epsilon^{-\frac{1}{\mu}} |\log(\epsilon)|^{1+\frac{1}{\mu}}$$

since $\gamma = 1/\mu$ and $|\log(\epsilon)| > 1$.

(iii) $1 - \gamma\mu < 0$:
Using inequality (5.32), we obtain $(G) \lesssim \epsilon^{\frac{1-\gamma\mu}{\mu+1}}$. Therefore, we have

$$(I) \lesssim \epsilon^{-\frac{1}{\mu}} \left(\epsilon^{\frac{1-\gamma\mu}{\mu+1}}\right)^{\frac{\mu+1}{\mu}} \eqsim \epsilon^{-\gamma}.$$

Combining the estimations for the terms (I) and (II) yields

$$C_\epsilon^{\mathrm{MLSC}} \lesssim \epsilon^{-\gamma} + \epsilon^{-\gamma} \eqsim \epsilon^{-\gamma}.$$ □

Similar to [106], we interpret the complexity result of the previous theorem by considering (5.33). Moreover, we show the benefit of the ML approach by comparing the results with the complexity of the single-level SC approach proved in Lemma 5.11. In the first case $\gamma\mu < 1$, the term $q^{k(1-\gamma\mu)/(\mu+1)}$ in (5.33) decreases with increasing k since $q \in (0,1)$. Therefore, the most of the computational effort will be on the coarsest level $k = 0$ computing $\mathcal{A}_{Q_0}[u_{h_0}]$. Comparing with the single-level SC approach, the cost savings are $C_\epsilon^{\mathrm{SC}}/C_\epsilon^{\mathrm{MLSC}} \eqsim \epsilon^{-\gamma}$. If $\gamma\mu > 1$, the most of the computational effort is required on the finest level $k = K$ and the cost savings are $C_\epsilon^{\mathrm{SC}}/C_\epsilon^{\mathrm{MLSC}} \eqsim \epsilon^{-1/\mu}$. In the case $\gamma\mu = 1$, the term $q^{k(1-\gamma\mu)/(\mu+1)}$ in (5.33) is equal to one and, hence, the computational effort is distributed equally across the levels. The cost savings are up to a log factor of the order $\epsilon^{-\gamma}$.

The previous complexity theorem is formulated for a given sequence of physical tolerances and determines the number $K \in \mathbb{N}_+$. In our case, we have access to an adaptive physical solver which works with an arbitrary physical tolerance as input parameter. Therefore, we can first choose a fixed $K \in \mathbb{N}_+$ and then the physical tolerance $TolH_K$ depending on C_H and the desired total accuracy ϵ:

$$TolH_K = \frac{\epsilon}{2\,C_H}. \tag{5.34}$$

The remaining tolerances are defined by $TolH_k = q^{k-K} TolH_K$. The choice of the stochastic tolerances and the complexity analysis are performed as before.

5.4 Real-Valued Quantity of Interest

In many applications, a typical QoI is a functional of the solution which can be linear or nonlinear. We define the considered QoI $\Phi : \Gamma \to \mathbb{R}$ by

$$\Phi : \mathbf{y} \mapsto \mathcal{F}[u(\mathbf{y})],$$

where $\mathcal{F} : \mathbb{L}^1 \to \mathbb{R}$ is a functional and u is the stochastic entropy solution of problem (3.8)–(3.12).

The natural error norm is the $L^2_\rho(\Gamma)$-norm equivalent to the $L^2_\rho(\Gamma, \mathbb{L}^1)$-norm considering the solution. For this norm, the SC theory for the full solution u presented in the two previous sections can be completely transferred to the functional case. However, one is usually interested in the expected value of the functional of the solution such that we consider the error

$|\mathbb{E}[\Phi - \widetilde{\Phi}]|$, where $\widetilde{\Phi}$ denotes the SLSC or MLSC approximation of Φ. Note that Jensen's inequality provides

$$|\mathbb{E}[\Phi - \widetilde{\Phi}]| \leq \mathbb{E}\big[|\Phi - \widetilde{\Phi}|\big] \leq \mathbb{E}\big[|\Phi - \widetilde{\Phi}|^2\big]^{1/2} = \|\Phi - \widetilde{\Phi}\|_{L^2_\rho(\Gamma)}.$$

Therefore, the results of the error analysis and complexity analysis performed for the $L^2_\rho(\Gamma)$-norm are also valid for $|\mathbb{E}[\Phi - \widetilde{\Phi}]|$. Typically, the analysis is adapted directly to the error of the expected value of the QoI Φ by formulating Assumptions 5.8, 5.10 and 5.14, concerning the stochastic approximations, for the expected value.

In this section, we only present the main aspects and results of the adaptive SLSC and the adaptive MLSC method applied to a functional of the solution. In the context of gas networks, a detailed presentation of the adaptive MLSC method applied directly to functionals of the solution have already been published in [68].

From the previous sections, we recall that a physical tolerance $TolH > 0$ is given. We compute numerical approximations $\Phi_h(\mathbf{y}) := \mathcal{F}[u_h(\mathbf{y})]$ of $\Phi(\mathbf{y})$ by using an adaptive physical solver which computes the approximate solution $u_h(\mathbf{y})$ with a sample-dependent resolution in space, time and model hierarchy. This resolution is refined until the error estimate for Φ_h is smaller than the tolerance $TolH$. For the multi-level structure, we consider a sequence of physical tolerances $\{TolH_k\}_{k=0}^K$ with

$$1 \geq TolH_0 > TolH_1 > \cdots > TolH_K > 0$$

and $K \in \mathbb{N}_+$. We set $\Phi_{h_{-1}} := 0$ and compute the approximations $\Phi_{h_k}(\mathbf{y}) := \mathcal{F}[u_{h_k}(\mathbf{y})]$ by the adaptive physical solver for tolerance $TolH_k$ for $k = 0, \ldots, K$. The single-level and the multi-level interpolation operator are defined as

$$\Phi_h^{(\mathrm{SL})} := \mathcal{A}_Q[\Phi_h],$$
$$\Phi_K^{(\mathrm{ML})} := \sum_{k=0}^{K} \mathcal{A}_{Q_k}[\Phi_{h_k} - \Phi_{h_{k-1}}],$$

where $\mathcal{A}_Q : C(\Gamma) \to L^2_\rho(\Gamma)$ is an interpolation operator using Q points in the stochastic space Γ. If the functional $\mathcal{F}$ and the interpolation operators are linear, then we have

$$\Phi_h^{(\mathrm{SL})}(\mathbf{y}) = \mathcal{F}\big[\mathcal{A}_Q[u_h(\mathbf{y})]\big] \quad \text{and} \quad \Phi_K^{(\mathrm{ML})}(\mathbf{y}) = \mathcal{F}\big[u_K^{(\mathrm{ML})}(\mathbf{y})\big]$$

for $\mathbf{y} \in \Gamma$ [106].

Starting with the single-level approach, we suppose that Assumption 5.7 holds for the approximated QoI Φ_h and Assumption 5.8 for the expected value of this approximation. If we choose $TolH = \epsilon/(2\,C_H)$ and $TolY = \epsilon/(2\,C_Y)$ as in (5.17), the SLSC error achieves the prescribed accuracy $\epsilon > 0$. For the complexity of the SLSC method, we obtain the estimate $C_\epsilon^{\mathrm{SC}} \lesssim \epsilon^{-\gamma-1/\mu}$ with convergences rates γ and μ supposed in Assumption 5.10 replacing the norm of the approximate solution by the absolute value of the expected value of the approximated QoI Φ_h.

Before we state the multi-level complexity theorem, we formulate the required assumptions which are similar to Assumptions 5.13, 5.14, 5.17 and 5.19, but now adapted for the expected value of Φ_{h_k}.

Assumption 5.22
There exist a physical constant $c_k : \Gamma \to \mathbb{R}_+$ *with* $\mathbb{E}[c_k] < \infty$ *and further positive constants* $\gamma, \mu, C_Y, C_W, C_I$ *such that for all* $k \in \{0, \dots, K\}$ *it holds*

(i) $|\Phi(\mathbf{y}) - \Phi_{h_k}(\mathbf{y})| \leq c_k(\mathbf{y})\, TolH_k$ *for all* $\mathbf{y} \in \Gamma$,

(ii) $\Phi_{h_k} \in C(\Gamma) \subset L^2_\rho(\Gamma)$,

(iii) $\left|\mathbb{E}\left[(\Phi_{h_k} - \Phi_{h_{k-1}}) - \mathcal{A}_{Q_k}[\Phi_{h_k} - \Phi_{h_{k-1}}]\right]\right| \leq C_Y\, TolY_k$, *where the constant* C_Y *does not depend on* k *and* $\mathbf{y} \in \Gamma$,

(iv) $C_Y\, TolY_k = C_I(N)\, Q_k^{-\mu}\, TolH_{k-1}$ *with* $TolH_{-1} := |\mathbb{E}[\Phi_{h_0}]|$ *and* $C_I(N)$ *is independent of* k *and* μ,

(v) $W_k \leq C_W\, TolH_k^{-\gamma}$, *where* W_k *is an upper bound for the cost of each sample of* Φ_{h_k} *and* C_W *is independent of* k *and* γ.

Similar to the first part of the proof of Theorem 5.16, Assumption 5.22 (i) provides

$$|\mathbb{E}[\Phi - \Phi_{h_k}]| \leq C_H\, TolH_k \tag{5.35}$$

with the constant $C_H = \max_{k=0,\dots,K} \mathbb{E}[c_k]$. In order to realize Assumption 5.22 (iii), we apply the adaptive sparse grid algorithm from Subsection 5.1.6 which uses heuristic error indicators. These indicators are computed for the case **ExVal** in Algorithm 5.2 so that they are consistent with the considered error. If we choose the stochastic tolerance $TolY_k$ analogously to (5.28), then we obtain the desired accuracy ϵ for the MLSC error and the following complexity estimate.

Theorem 5.23
Let a null sequence of physical tolerances $\{TolH_k\}_{k=0,1,\dots}$ *be given by* $TolH_k = q^k\, TolH_0$ *with* $TolH_0 \in (0, 1]$ *and a reduction factor* $q \in (0, 1)$. *Further, let Assumptions 3.8 and 5.22 be fulfilled. Then, for any* $\epsilon \in (0, 1/e]$, *there exists a number* $K = K(\epsilon) \in \mathbb{N}_+$ *and a sequence* $\{TolY_k\}_{k=0}^K$ *such that*

$$\left|\mathbb{E}\left[\Phi - \Phi_K^{(\mathrm{ML})}\right]\right| \leq \epsilon$$

and

$$C_\epsilon^{\mathrm{MLSC}} \lesssim \begin{cases} \epsilon^{-\frac{1}{\mu}} & if\ \gamma\mu < 1, \\ \epsilon^{-\frac{1}{\mu}} |\log \epsilon|^{1+\frac{1}{\mu}} & if\ \gamma\mu = 1, \\ \epsilon^{-\gamma} & if\ \gamma\mu > 1. \end{cases}$$

Proof. The proof is analogous to the proof of Theorem 5.21. □

For details about the interpretation of the previous complexity result and the cost savings compared to the single-level approach, we refer to the discussion in Subsection 5.3.1.

5.5 Implementation

In this section, we describe the algorithm of our single-level and multi-level stochastic collocation methods. Both algorithms are self-adaptive and fully error-controlled such that the provided approximation achieves an accuracy close to the user-prescribed accuracy ϵ. We mainly follow the algorithmic concepts described for functionals of the solution in [67, 68].

As mentioned in the previous sections, we have an adaptive black box solver at our disposal which is called for a fixed collocation point $\mathbf{y}_i \in \Gamma$ and physical tolerance $TolH > 0$. This algorithm numerically solves the deterministic network problem resulting from inserting $\mathbf{y}_i$ into the stochastic network problem (3.8)–(3.12) such that the resulting numerical approximation satisfies the accuracy requirement in Assumptions 5.7 and 5.13. We denote the adaptive solver by $\text{ADET}(\cdot,\cdot)$ and the provided physical approximations of $u(\mathbf{y}_i)$ by

$$u_h(\mathbf{y}_i) = \text{ADET}(\mathbf{y}_i, TolH), \qquad u_{h_k}(\mathbf{y}_i) = \text{ADET}(\mathbf{y}_i, TolH_k).$$

Regarding the stochastic approximations in Assumptions 5.8 and 5.14, we use the adaptive sparse grid algorithm ASPAGRID described in Algorithm 5.1 (Subsection 5.1.6) with tolerance $TolY > 0$. In order to compute the samples, we run the physical black box solver ADET with a given physical tolerance for each point $\mathbf{y}_i \in \Gamma$. The algorithm returns the corresponding sparse grid approximation as well as an approximation of its expected value. Since the error of the solution is considered in the $L^2_\rho(\Gamma, \mathbb{L}^1)$-norm, we compute the heuristic error indicators of the sparse grids by choosing **Ltwo** as 'profitName' in Algorithm 5.2.

Let $\epsilon > 0$ be a user-prescribed accuracy for the error of the SC approximation. Now, choosing the tolerances $TolH, TolY$ as in (5.17) depending on ϵ delivers the SLSC method illustrated in Algorithm 5.3. The multi-level algorithm is described in Algorithm 5.4 and determines the stochastic tolerances as in (5.28). The constant $TolH_{-1}$ in line 6 in Algorithm 5.4 can be computed by the adaptive sparse grid algorithm ASPAGRID with **Ltwo** as 'profitName'. In order to obtain a sufficiently accurate approximation with acceptable cost, we choose $TolH_0$ as the user-prescribed stochastic tolerance. As in the complexity theorem 5.21, we focus on physical tolerances $TolH_k = q^k\, TolH_0$ with a positive reduction factor $q < 1$ where we choose $TolH_K = \epsilon/(2\, C_H)$ as in (5.34). The samples $u_{h_0}(\mathbf{y}_i)$ computed in line 5 can be reused for the computations on the coarsest level $k = 0$ in line 8. Furthermore, the samples $u_{h_k}(\mathbf{y}_i)$ computed on level k can be reused on level $k + 1$.

Algorithm 5.3 Adaptive single-level stochastic collocation method for the approximation of the solution u.

1: **procedure** ASINGLELEVELSC(ϵ, ADET)
2: Estimate C_h and C_y
3: $TolH = \epsilon/(2\, C_h)$
4: $TolY = \epsilon/(2\, C_y)$
5: $\mathcal{A}_Q[u_h] = \text{ASPAGRID}(\text{ADET}(\cdot, TolH), TolY, \textbf{Ltwo})$
6: **return** $\mathcal{A}_Q[u_h]$
7: **end procedure**

In order to estimate the parameters C_H, C_Y, μ and γ, we study a few samples with relatively coarse resolutions in a pre-processing step. For the physical constant C_H and rate γ, we compute the QoI for a decreasing sequence of physical tolerances using a fixed isotropic Smolyak grid of a small level w. The parameters C_Y and μ which characterize the convergence rate and the quality of the error estimates of the algorithm ASPAGRID are estimated by choosing a decreasing sequence of stochastic tolerances and a fixed coarse physical tolerance. The rates γ and μ are estimated via a least square fit applied to Assumption 5.17 andAssumption 5.19, respectively. The estimations C_H and C_Y are also estimates for C_h and C_y in the single-level approach, respectively. Note that the sparse grid returned for the stochastic tolerance $TolY_k$ does not use

exactly the rounded value of the optimal number of samples, Q_k, since an adaptive sparse grid corresponding to this value does not necessarily exist.

Algorithm 5.4 Adaptive multi-level stochastic collocation method for the approximation of the solution u.

procedure AMULTILEVELSC(ϵ, q, K, ADET)
  Estimate C_H, C_Y, γ and μ
  $TolH_K = \epsilon/(2\,C_H)$
  $TolH_k = q^{k-K} TolH_K, \quad$ for $k = 0, \dots, K-1$
  $\mathcal{A}_{Q_{-1}}[u_{h_0}] = \text{ASPAGRID}(\text{ADET}(\cdot, TolH_0), TolH_0, \textbf{Ltwo})$
  $TolH_{-1} = \|\mathcal{A}_{Q_{-1}}[u_{h_0}]\|_{L^2_\rho(\Gamma,\mathbb{L}^1)}$
  $TolY_k = \epsilon\,(2\,C_Y\,F(\mu,\gamma))^{-1}\,G_k(\gamma)^{\frac{\mu}{\mu+1}}\,TolH_{k-1}^{\frac{1}{\mu+1}}$ with $F(\mu,\gamma)$ and $G_k(\gamma)$ as in (5.27), for $k = 0, \dots, K$
  $\mathcal{A}_{Q_0}[u_{h_0}] = \text{ASPAGRID}(\text{ADET}(\cdot, TolH_0), TolY_0, \textbf{Ltwo})$
  $\mathcal{A}_{Q_k}[u_{h_k} - u_{h_{k-1}}] = \text{ASPAGRID}(\text{ADET}(\cdot, TolH_k) - \text{ADET}(\cdot, TolH_{k-1}), TolY_k, \textbf{Ltwo})$, for $k = 1, \dots, K$
  $u_K^{(\text{ML})} = \mathcal{A}_{Q_0}[u_{h_0}] + \sum_{k=1}^K \mathcal{A}_{Q_k}[u_{h_k} - u_{h_{k-1}}]$
  return $u_K^{(\text{ML})}$
end procedure

5.5.1 Real-Valued Quantity of Interest

The SLSC and the MLSC algorithm for the approximation of functionals of the solution analyzed in Section 5.4 differ slightly from the previous algorithms. First, the adaptive physical solver ADET provides a numerical approximation of $\Phi(\mathbf{y}_i) := \mathcal{F}[u(\mathbf{y}_i)]$ which satisfies the accuracy property in Assumption 5.22 and is denoted by

$$\Phi_h(\mathbf{y}_i) = \text{ADET}(\mathbf{y}_i, TolH) \quad \text{or} \quad \Phi_{h_k}(\mathbf{y}_i) = \text{ADET}(\mathbf{y}_i, TolH_k).$$

As in the previous algorithms, we use the adaptive sparse grid algorithm ASPAGRID from Subsection 5.1.6 which calls, for each collocation point, the physical solver with a given physical tolerance. Since we consider the error of the expected value of the functional of the solution, the heuristic error indicators are computed by choosing **ExVal** in Algorithm 5.2. Additionally, the algorithm yields the approximate expected value of the functional, see Subsection 5.1.4. We approximate the expected value of the SLSC and the MLSC approximation of the QoI by

$$\mathbb{E}[\Phi_h^{(\text{SL})}] \approx \mathcal{E}_Q[\Phi_h],$$
$$\mathbb{E}[\Phi_K^{(\text{ML})}] = \sum_{k=0}^K \mathbb{E}\big[\mathcal{A}_{Q_k}[\Phi_{h_k} - \Phi_{h_{k-1}}]\big] \approx \sum_{k=0}^K \mathcal{E}_{Q_k}[\Phi_{h_k} - \Phi_{h_{k-1}}].$$

The previous remarks about the reuse of samples and the estimation of the constants C_H, C_Y and the rates μ, γ remain valid.

Algorithm 5.5 Adaptive single-level stochastic collocation method for the approximation of the expected value $\mathbb{E}[\Phi]$.

procedure ASINGLELEVELSC(ϵ, ADET)
  Estimate C_h and C_y
  $TolH = \epsilon/(2\,C_h)$
  $TolY = \epsilon/(2\,C_y)$
  $\mathcal{E}_Q[\Phi_h] = \text{ASPAGRID}(\text{ADET}(\cdot, TolH), TolY, \textbf{ExVal})$
  return $\mathcal{E}_Q[\Phi_h]$
end procedure

Algorithm 5.6 Adaptive multi-level stochastic collocation method for the approximation of the expected value $\mathbb{E}[\Phi]$.

procedure AMULTILEVELSC(ϵ, q, K, ADET)
  Estimate C_H, C_Y, γ and μ
  $TolH_K = \epsilon/(2\,C_H)$
  $TolH_k = q^{k-K} TolH_K, \quad \text{for } k = 0, \ldots, K-1$
  $TolH_{-1} = |\mathcal{E}_{Q_{-1}}[\Phi_{h_0}]| = |\text{ASPAGRID}(\text{ADET}(\cdot, TolH_0), TolH_0, \textbf{ExVal})|$
  $TolY_k = \epsilon\,(2\,C_Y\,F(\mu,\gamma))^{-1} G_k(\gamma)^{\frac{\mu}{\mu+1}}\, TolH_{k-1}^{\frac{1}{\mu+1}}$ with $F(\mu,\gamma)$ and $G_k(\gamma)$ as in (5.27), for $k = 0, \ldots, K$
  $\mathcal{E}_{Q_0}[\Phi_{h_0}] = \text{ASPAGRID}(\text{ADET}(\cdot, TolH_0), TolY_0, \textbf{ExVal})$
  $\mathcal{E}_{Q_k}[\Phi_{h_k} - \Phi_{h_{k-1}}] = \text{ASPAGRID}(\text{ADET}(\cdot, TolH_k) - \text{ADET}(\cdot, TolH_{k-1}), TolY_k, \textbf{ExVal})$, for $k = 1, \ldots, K$
  $\mathcal{E}_K^{(\mathrm{ML})}[\Phi_{h_K}] = \sum_{k=0}^{K} \mathcal{E}_{Q_k}[\Phi_{h_k} - \Phi_{h_{k-1}}] \approx \mathbb{E}[\Phi_K^{(\mathrm{ML})}]$
  return $\mathcal{E}_K^{(\mathrm{ML})}[\Phi_{h_K}]$
end procedure

Chapter 6

Kernel Density Estimator

In this chapter, we introduce the kernel density estimator (KDE) which is a sample-based approach to estimate the (usually unknown) probability density function (PDF) of a random variable (RV). Furthermore, we analyze its convergence to the exact PDF and describe how to approximate probabilities of the random variables. In addition, we show the convergence also for the approximated probabilities and consider computational aspects for a common choice of the KDE. In the case of bounded random variables, we consider a boundary correction method which provides an approximated PDF with the same support as the exact PDF. Finally, we apply the presented methods to an SC approximation such that we obtain a cost-efficient approach to validate the feasibility of the approximation in a post-processing step of an SC method.

The first two sections and Section 6.5 are mainly based on [46]. Some parts of Section 6.4 are already published by the author in collaboration with Schuster, Gugat and Lang in [98] where the KDE was applied to probabilistic constrained optimization problems.

Let $(\Theta, \Sigma, \mathbb{P})$ be a probability space with sample space Θ, sigma algebra $\Sigma \subset \mathcal{P}(\Theta)$, where $\mathcal{P}(\Theta)$ is the power set of Θ, and probability measure $\mathbb{P} : \Sigma \to [0,1]$. We consider an n-dimensional $\mathbb{R}^n$-valued random variable (RV) $X : \Theta \to \mathbb{R}^n$ which we assume to have an absolutely continuous distribution. Therefore, the RV has a PDF $\rho_X : \mathbb{R}^n \to \mathbb{R}_+$ which however can be unknown. Now, we want to approximate this function by using a kernel density estimator which is based on a sampling set of the RV. First, we define the KDE in a general form including the univariate and multivariate case [46, 99].

Definition 6.1 (Kernel Function)
We define a kernel (function) as an integrable and Borel measurable function $\mathcal{K}_n : \mathbb{R}^n \to \mathbb{R}_+$ which satisfies

$$\int_{\mathbb{R}^n} \mathcal{K}_n(z)\,\mathrm{d}z = 1. \tag{6.1}$$

Definition 6.2 (Kernel Density Estimator)
Let X be an $\mathbb{R}^n$-valued random variable with an absolutely continuous distribution and PDF $\rho_X : \mathbb{R}^n \to \mathbb{R}_+$. Moreover, let $\mathcal{X} = \{X^{(1)}, \dots, X^{(S)}\} \subset \mathbb{R}^n$ be a finite, independent and identically distributed sampling of the random variable X where each sample $X^{(i)}$ is a realization of X. Then, the kernel density estimator $k_S : \mathbb{R}^n \to \mathbb{R}_+$ is defined as

$$k_S(z) = \frac{1}{S\ \det(H)^{1/2}} \sum_{i=1}^{S} \mathcal{K}_n\left(H^{-1/2}\big(z - X^{(i)}\big)\right) \tag{6.2}$$

with a symmetric positive definite bandwidth matrix $H = H(S, \mathcal{X}) \in \mathbb{R}^{n \times n}$ and a kernel function $\mathcal{K}_n : \mathbb{R}^n \to \mathbb{R}_+$.

The KDE can be interpreted as a weighted sum of single kernels $\mathcal{K}_n$ centered at samples $X^{(i)}$. In general, the matrix H depends on the number of samples and on the data $X^{(i)}$, but not on z and not directly on the RV X.

The KDE k_S is a PDF on $\mathbb{R}^n$ since it is a non-negative and integrable function which integrates to one. The latter property follows from the definition of kernel functions and the change of variables $z_i = H^{1/2}t_i + X^{(i)}$:

$$\int_{\mathbb{R}^n} k_S(z)\,\mathrm{d}z = \frac{1}{S \cdot \det(H)^{1/2}} \sum_{i=1}^{S} \int_{\mathbb{R}^n} \mathcal{K}_n\left(H^{-1/2}\big(z_i - X^{(i)}\big)\right)\,\mathrm{d}z_i \tag{6.3}$$

$$= \frac{1}{S \cdot \det(H)^{1/2}} \det(H^{1/2}) \sum_{i=1}^{S} \underbrace{\int_{\mathbb{R}^n} \mathcal{K}_n\left(t_i\right)\,\mathrm{d}t_i}_{=1} = 1. \tag{6.4}$$

Note that we obtain $\det(H^{1/2}) = \det(H)^{1/2}$, since the matrix H is positive definite.

6.1 Univariate Case

For the univariate case $n = 1$, it is common to set $H(S, \mathcal{X}) = h^2$ with $h > 0$. This leads to the KDE

$$k_S(z) = \frac{1}{Sh} \sum_{i=1}^{S} \mathcal{K}_1\left(\frac{z - X^{(i)}}{h}\right), \tag{6.5}$$

where $\mathcal{K}_1 : \mathbb{R} \to \mathbb{R}_+$ a univariate kernel and h is called the *bandwidth* or *smoothing parameter*. The construction of the kernel density estimation for three different bandwidths is illustrated in Figure 6.1. For the analysis of the KDE and the choice of the bandwidth, further conditions are often imposed on the kernel function. In [46, 99], the kernel function has to be symmetric and to satisfy

$$\int z\mathcal{K}_1(z)\,\mathrm{d}z = 0 \quad \text{(mean zero)}$$
$$\int z^2\mathcal{K}_1(z)\,\mathrm{d}z < \infty \quad \text{(bounded variance)}.$$

The most common kernel functions are listed in Table 6.1 where $\mathbb{1}_A$ is the indicator function. These functions fulfill the previous conditions and have bounded support, except the Gauss kernel. However, the Gauss kernel is frequently used because the resulting KDE is continuous and differentiable.

Kernel Name	$\mathcal{K}_1(z)$
Gauss (Normal)	$(2\pi)^{-1/2}\exp(-z^2/2)$
Uniform (Box)	$1/2\ \mathbb{1}_{[0,1)}(\lvert z\rvert)$
Triangular	$(1-\lvert z\rvert)\ \mathbb{1}_{[0,1)}(\lvert z\rvert)$
Epanechnikov	$2/4\ (1-z^2)\ \mathbb{1}_{[0,1)}(\lvert z\rvert)$

Table 6.1: Typical univariate kernel functions.

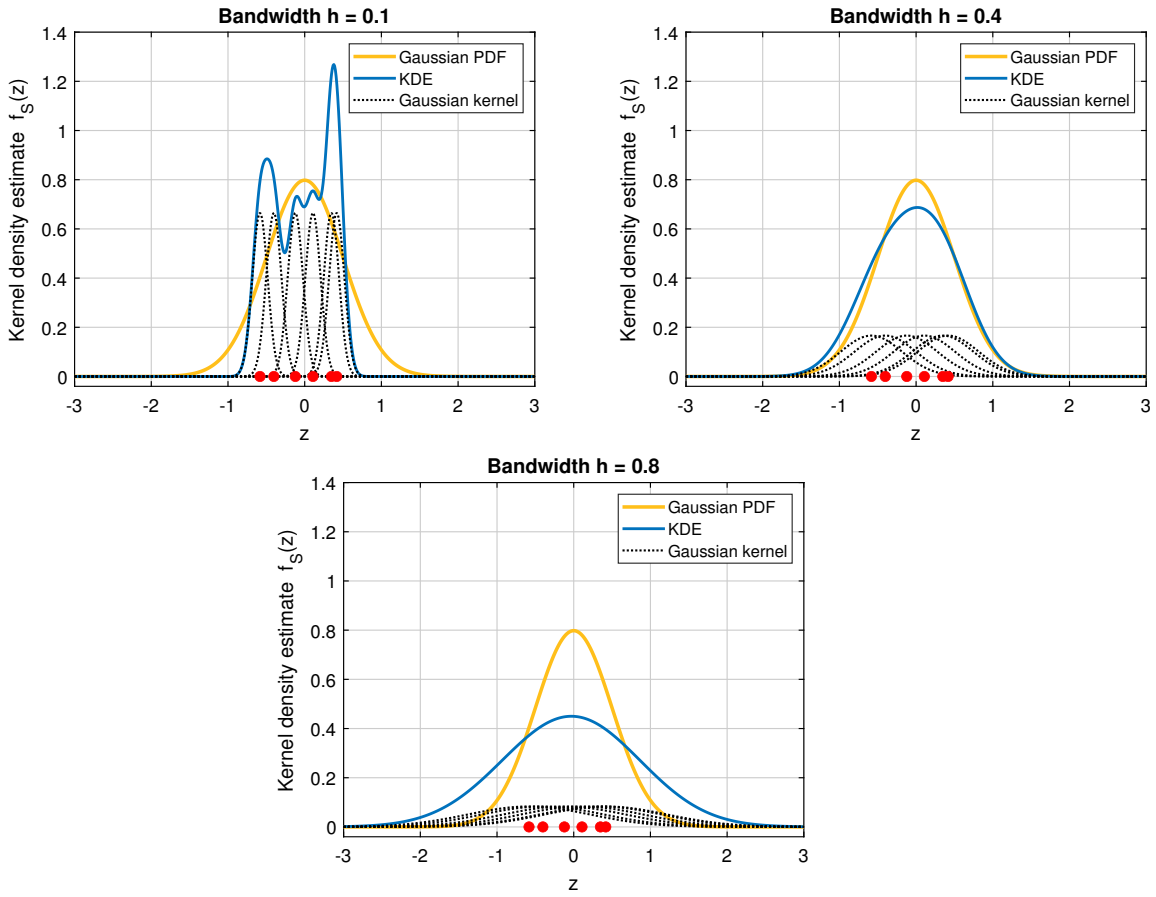


Figure 6.1: Kernel density estimations using the Gaussian kernel are plotted in blue for $S = 6$ and three different bandwidths $h = 0.1, 0.4, 0.8$. The samples $X^{(i)}$ (red points) are drawn from the Gaussian distribution $\mathcal{N}(0, 0.5)$. Each dotted, black line is a Gaussian kernel belonging to a single sample $X^{(i)}$ and weighted by S^{-1}. The exact Gaussian PDF is plotted in orange.

Beside the kernel function, we have to specify the bandwidth h in the formulation (6.5) of the KDE. The choice of the bandwidth has a much higher influence on the quality of the KDE than the choice of the kernel [46, 99]. In Figure 6.1, we illustrate the influence of the bandwidth h on the kernel and on the overall density estimation. We use the Gaussian kernel and 6 samples drawn from a Gaussian distribution to compute the kernel density estimation for three different bandwidths. For the bandwidth $h = 0.1$, the KDE has several peaks and is thus undersmoothed. In contrast, the bandwidth $h = 0.8$ has a oversmoothing behavior such that the width of the resulting KDE gets too large. Consequently, the bandwidth should neither be too small nor too large, because then the KDE is not a good approximation of the PDF ρ_X. The computation of the optimal bandwidth is a challenging and complex task which is not a focus of this work. An overview of the different methods can be found in [51, 52, 110]. For the Gaussian kernel, we use a heuristic formula, the so-called rule-of-thumb [51, 99]

$$h = h(S, \mathcal{X}) = \left(\frac{4}{3S}\right)^{\frac{1}{5}} \sigma_{\mathcal{X}} \approx 1.06 \, \sigma_{\mathcal{X}} \, S^{-\frac{1}{5}} \tag{6.6}$$

with the standard deviation $\sigma_{\mathcal{X}} \in \mathbb{R}_+$ of the sampling set $\mathcal{X}$ which is an estimate of the standard deviation of the RV X. For the derivation of this bandwidth selection, it is assumed that the unknown PDF ρ_X is a normal density. Härdle et al. [51] states that this rule also provides

acceptable results if the distribution of the RV X has no fat tail and is almost symmetric as well as unimodal, i.e. the PDF has only one peak/maximum. In Section 6.3, we show that the KDE converges $\mathbb{P}$-almost surely for the chosen bandwidth (6.6). So we can conclude the following relation: The larger the sampling set the lower the influence of the bandwidth choice on the accuracy of the KDE. Summing up, it is reasonable to use the rule-of-thumb for a PDF which looks roughly like a normal PDF.

6.2 Multivariate Case

For the multivariate case $n > 1$, we need to choose a multivariate kernel function $\mathcal{K}_n : \mathbb{R}^n \to \mathbb{R}_+$ and a symmetric positive definite bandwidth matrix H. There are two common approaches how to construct the multivariate kernels from univariate kernels [46, 51]. The first technique is to define the multivariate kernel $\mathcal{K}_n$ as product kernel

$$\mathcal{K}_n(z) = \prod_{j=1}^{n} \mathcal{K}_1(z_j)$$

with the univariate kernel $\mathcal{K}_1 : \mathbb{R} \to \mathbb{R}_+$. Hence, the multivariate KDE in (6.2) has the form

$$k_S(z) = \frac{1}{S \ \det(H)^{1/2}} \sum_{i=1}^{S} \prod_{j=1}^{n} \mathcal{K}_1 \left(\left(H^{-1/2}(z - X^{(i)}) \right)_j \right). \tag{6.7}$$

The other approach is to use radially symmetric kernel functions which are generated by $\mathcal{K}_n(z) = c(\mathcal{K}_1, n)\mathcal{K}_1(\|z\|)$. The constant $c(\mathcal{K}_1, n) \in \mathbb{R}$ is determined by condition (6.1) for kernel functions. Due to their construction, these kernels have the same value for all points on a n-dimensional sphere around zero. In Table 6.2, we extend the univariate kernels listed in Table 6.1. Note that the multivariate Gaussian kernel coincides with the product kernel composed of the univariate Gaussian kernel. As mentioned in the univariate case, it is common to choose the standard multivariate normal density as kernel $\mathcal{K}_n$ since the resulting KDE is continuous and differentiable.

Kernel Name	$\mathcal{K}_n(z)$
Gauss (Normal)	$(2\pi)^{-d/2} \exp(-\|z\|^2/2)$
Uniform (Box)	$\pi^{-d/2} \ \Gamma(1 + d/2) \ \mathbb{1}_{[0,1)}(\|z\|)$
Triangular	$(d+1)\pi^{-d/2} \ \Gamma(1 + d/2)(1 - \|z\|) \ \mathbb{1}_{[0,1)}(\|z\|)$
Epanechnikov	$\pi^{-d/2} \ \Gamma(2 + d/2)(1 - \|z\|^2) \ \mathbb{1}_{[0,1)}(\|z\|)$

Table 6.2: Typical radially symmetric multivariate kernel functions. Here, $\Gamma(\cdot)$ denotes the Gamma function and $\|\cdot\|$ the euclidean norm.

Similar to the univariate KDE, the choice of the bandwidth matrix H is an important task since it influences the orientation and shape of the kernels [46]. The bandwidth matrices H can be divided into three classes which are presented below. In [111], the classes are compared for several two-dimensional PDFs using the same kernel.

The simplest choice is $H = h_s^2 I_{n \times n}$ with $h_s > 0$. Since we use the same smoothing parameter h_s in each dimension, this approach is only adequate if the spread of the data is approximately the

same in each dimension. But this condition is in many datasets not fulfilled [46].

An extension of the previous class is to use different parameter h_i in each dimension such that we have a diagonal bandwidth matrix $H = \operatorname{diag}(h_1^2, \cdots, h_n^2)$ with $h_i = h_i(S, \mathcal{X}) > 0$ for $i = 1, \ldots, n$. For this case, the KDE has the form

$$k_S(z) = \frac{1}{S \prod_{j=1}^{n} h_j} \sum_{i=1}^{S} \mathcal{K}_n \left(\frac{z_1 - X_1^{(i)}}{h_1}, \cdots, \frac{z_n - X_n^{(i)}}{h_n} \right).$$

These bandwidth matrices are often suitable and should be the minimum requirement for the KDE since they allow a flexible smoothing by different parameters [111]. The heuristic rule (6.6) from the univariate case can be extended to the diagonal bandwidth matrix $H = \operatorname{diag}(h_1^2, \ldots, h_n^2)$ with

$$h_i = \left(\frac{4}{(n+2)S} \right)^{1/(n+4)} \sqrt{(\Sigma_{\mathcal{X}})_{ii}} \tag{6.8}$$

for $i = 1, \ldots, n$ where $\Sigma_{\mathcal{X}}$ denotes the covariance matrix of the sampling set $\mathcal{X}$ [46, 99]. Since the RV X has an absolutely continuous distribution, it holds $(\Sigma_X)_{ii} > 0$ which ensures that the matrix H is positive definite and $H^{-1/2}$ exists. In practice, the product kernel combined with a diagonal bandwidth matrix is actually recommended and adequate [99]. For this case, the estimator is given by

$$k_S(z) = \frac{1}{S \prod_{j=1}^{n} h_j} \sum_{i=1}^{S} \prod_{j=1}^{n} \mathcal{K}_1 \left(\frac{z_j - X_j^{(i)}}{h_j} \right). \tag{6.9}$$

If we choose the multivariate Gaussian kernel with a diagonal bandwidth matrix, then we get immediately the product kernel with the standard univariate normal density $\mathcal{K}_1$:

$$k_S(z) = \frac{1}{(\sqrt{2\pi})^n S \prod_{j=1}^{n} h_j} \sum_{i=1}^{S} \prod_{j=1}^{n} \exp \left(-\frac{\left(z_j - X_j^{(i)}\right)^2}{2h_j} \right). \tag{6.10}$$

In some situations, the use of bandwidth matrices of the previous two classes could be insufficient, for example, for asymmetric data with a north-west orientation. Therefore, a dense bandwidth matrix has to be chosen because the corresponding KDE should provide the best estimation of the exact PDF of the RV X [46]. One reasonable choice is a matrix which depends on the sample covariance matrix. Assuming that the sample covariance matrix is positive definite, the previous heuristic formula (6.8) can be generalized to a bandwidth matrix given by

$$H = \left(\frac{4}{(n+2)S} \right)^{2/(n+4)} \Sigma_{\mathcal{X}}, \tag{6.11}$$

see [46]. If we use full bandwidth matrices, then the KDEs have a complex structure, a high computational effort and a difficult analysis. In order to simplify the estimation, we can pre-scale the data using the whitening transformation $Y^{(i)} := \Sigma_{\mathcal{X}}^{-1/2} X^{(i)}$ such that the covariance matrix of the transformed data is equal to the unit matrix [46, 101]. Then, we can use a KDE with only one smoothing parameter $h_s > 0$ for the transformed data (i.e. $H = h_s^2 I_{n \times n}$). However, this procedure is not practical for every dataset and thus it should not used thoughtless as an automatic tool [46].

6.3 Convergence of the Kernel Density Estimator

In this section, we show under which conditions the kernel density estimator converges to the exact PDF for $n \geq 1$. First, we consider the KDE (6.2) with bandwidth matrix $H = h_s^2 I_{n\times n}$ and $h_s = h_s(S, \mathcal{X}) > 0$. Devroye and Györfi proved the L^1-convergence of this estimator in [23, Chapter 6, Theorem 1].

Theorem 6.3
Let k_S be a KDE with $H = h_s^2 I_{n\times n}$. Additionally, let $h_s : (S, \mathcal{X}) \to \mathbb{R}_+$ be a Borel measurable function of S and the samples. If $\left(h_s + (Sh_s^n)^{-1}\right) \to 0$ $\mathbb{P}$-almost surely as $S \to \infty$, then

$$\|k_S - \rho_X\|_{L_1} = \int_{\mathbb{R}^n} |k_S(z) - \rho_X(z)| \,\mathrm{d}z \xrightarrow{S\to\infty} 0 \qquad \mathbb{P}\text{-almost surely.}$$

The previous theorem is only formulated for kernel density estimators with a single smoothing parameter h_s. Therefore, we generalize the result to estimators using a more general bandwidth matrix $H = h_s^2 A_{spd}$ with parameter $h_s = h_s(S, \mathcal{X}) > 0$ and symmetric positive definite matrix A_{spd}.

Theorem 6.4
Let k_S be a KDE with bandwidth matrix $H = h_s^2 A_{spd}$ where A_{spd} is a symmetric positive definite matrix and $h_s : (S, \mathcal{X}) \to \mathbb{R}_+$ is a Borel measurable function depending on S and the data $\mathcal{X}$. If $\left(h_s + (Sh_s^n)^{-1}\right) \to 0$ $\mathbb{P}$-almost surely as $S \to \infty$, then

$$\|k_S - \rho_X\|_{L_1} \xrightarrow{S\to\infty} 0 \qquad \mathbb{P}\text{-almost surely.}$$

Proof. The KDE k_S is based on the sampling $\mathcal{X} = \{X^{(1)}, \ldots, X^{(S)}\}$ of the RV X. First, we transform the samples $X^{(i)}$ to $Y^{(i)} := {A_{spd}}^{-1/2} X^{(i)}$ which are now samples of the transformed RV $Y := {A_{spd}}^{-1/2} X$. Due to this bijective transformation, the KDE can be written as

$$\begin{aligned} k_S(z) &= \frac{1}{S \,\det(h_s^2 A_{spd})^{1/2}} \sum_{i=1}^{S} \mathcal{K}_n\left((h_s^2 A_{spd})^{-1/2}\big(z - X^{(i)}\big)\right) \\ &= \det(A_{spd})^{-1/2} \frac{1}{S \cdot h_s^n} \sum_{i=1}^{S} \mathcal{K}_n\left(\big({A_{spd}}^{-1/2} z - Y^{(i)}\big)/h_s\right) \\ &= \det(A_{spd})^{-1/2}\, k_S^t({A_{spd}}^{-1/2} z), \end{aligned}$$

where k_S^t is the KDE for the transformed RV Y with bandwidth matrix $H = h_s^2 I_{n\times n}$. Furthermore, the transformation theorem [59, Theorem 1.101] shows that the exact PDF ρ_X of the random variable X and ρ_X^t of the transformed random variable Y are related by

$$\rho_X(z) = \det(A_{spd})^{-1/2}\, \rho_X^t({A_{spd}}^{-1/2} z).$$

Using the two previous relations and integration by substitution, we get

$$\begin{aligned} \|k_S - \rho_X\|_{L^1} &= \int_{\mathbb{R}^n} |k_S(z) - \rho_X(z)| \,\mathrm{d}z = \int_{\mathbb{R}^n} |k_S^t({A_{spd}}^{-1/2} z) - \rho_X^t({A_{spd}}^{-1/2} z)| \,\det(A_{spd})^{-1/2} \,\mathrm{d}z \\ &= \int_{\mathbb{R}^n} |k_S^t(y) - \rho_X^t(y)| \,\det(A_{spd})^{-1/2} \cdot \det(A_{spd})^{1/2} \,\mathrm{d}y = \|k_S^t - \rho_X^t\|_{L_1}. \end{aligned}$$

The convergence of the estimator k_S follows from Theorem 6.3 which states that the L_1 error $\|k_S^t - \rho_X^t\|_{L_1}$ converges $\mathbb{P}$-almost surely if the parameter h_s satisfies the condition that $\left(h_s + (Sh_s^n)^{-1}\right) \to 0$ $\mathbb{P}$-almost surely as $S \to \infty$. □

Considering the heuristic bandwidth rules (6.6), (6.8) and (6.11), we notice that the rules have the required form with

$$h_s = \left(\frac{4}{(n+2)S}\right)^{\frac{1}{n+4}}.$$

Applying now the previous theorem, we get convergence of the KDE since the parameter h_s converges:

$$h_s + (Sh_s^n)^{-1} = \left(\frac{4}{n+2}\right)^{\frac{1}{n+4}} S^{-\frac{1}{n+4}} + \left(\frac{n+2}{4}\right)^{\frac{n+1}{n}} S^{-\frac{4}{n+4}} \xrightarrow{S\to\infty} 0.$$

6.4 Approximation of Probabilities

The kernel density estimator is an approximation of the PDF of X and converges for a suitable bandwidth. Therefore, it is reasonable to use this estimator in order to compute probabilities for the RV X. The convergence can be ensured under the same conditions as in Section 6.3. Finally, we consider how to compute the probability using the Gaussian kernel with diagonal bandwidth matrix by means of MATLAB®.

The probability that the RV X takes values in a Borel set $B \subset \mathbb{R}^n$ is given by

$$\mathbb{P}(X \in B) := \mathbb{P}\left(\{\theta \in \Theta : X(\theta) \in B\}\right) = \int_B \rho_X(z)\,\mathrm{d}z.$$

This probability can be approximated by integrating the estimator k_S over the set B. We denote this approximation by

$$\mathbb{P}_S(X \in B) := \int_B k_S(z)\,\mathrm{d}z.$$

For the same setting as in Theorem 6.4, we show the convergence of the approximated probability towards the exact probability.

Theorem 6.5 (Convergence of Probability Estimation)
Let k_S be a KDE with bandwidth matrix $H = h_s^2 A_{spd}$ where A_{spd} is a symmetric positive definite matrix and $h_s : (S, \mathcal{X}) \to \mathbb{R}_+$ is a Borel measurable function depending on S and the data $\mathcal{X}$. If $\left(h_s + (Sh_s^n)^{-1}\right) \to 0$ $\mathbb{P}$-almost surely as $S \to \infty$, then it holds

$$|\mathbb{P}_S(X \in B) - \mathbb{P}(X \in B)| \leq \frac{1}{2}\|k_S - \rho_X\|_{L_1} \xrightarrow{S\to\infty} 0 \quad \mathbb{P}\text{-almost surely}$$

for any Borel set $B \subset \mathbb{R}^n$.

Proof. A short sketch of this proof is already published by the author in collaboration with Schuster, Gugat and Lang in [98, Section 2.1]. Let $\mathcal{B}$ be the class of all Borel sets of $\mathbb{R}^n$. Scheffé's lemma [23, Chapter 1] provides the equation

$$\sup_{\widetilde{B}\in\mathcal{B}} \left| \int_{\widetilde{B}} k_S(z)\,\mathrm{d}z - \int_{\widetilde{B}} \rho_X(z)\,\mathrm{d}z \right| = \frac{1}{2}\int_{\mathbb{R}^n} |k_S(z) - \rho_X(z)|\,\mathrm{d}z.$$

Therefore, we obtain the estimate

$$\begin{aligned}|\mathbb{P}_S(X \in B) - \mathbb{P}(X \in B)| &= \left| \int_B k_S(z)\,\mathrm{d}z - \int_B \rho_X(z)\,\mathrm{d}z \right| \\ &\leq \frac{1}{2} \int_{\mathbb{R}^n} |k_S(z) - \rho_X(z)|\,\mathrm{d}z\end{aligned}$$

for any Borel set $B \in \mathcal{B}$. From Theorem 6.5, the convergence of the approximation $\mathbb{P}_S$ to the exact probability follows directly. □

6.4.1 Computational Aspects using Gaussian Kernel

In the following, we consider the probability that the RV X takes values in a n-dimensional rectangle:

$$\mathbb{P}(X \in R_n) \qquad \text{with} \qquad R_n := [a_1, b_1] \times \cdots \times [a_n, b_n],$$

where $a_i < b_i$ and $a_i, b_i \in \mathbb{R}$ for $i = 1, \ldots, n$. This probability is usually of interest if each component of X has to be between a prescribed upper and lower bound. In order to approximate this probability by

$$\mathbb{P}_S\left(X \in R_n\right) = \int_{R_n} k_S(z)\,\mathrm{d}z, \tag{6.12}$$

we consider the KDE k_S with Gaussian kernel and a diagonal bandwidth matrix $H = \operatorname{diag}(h_1^2, \ldots, h_n^2)$ with $h_i > 0$, given in (6.10). First, we show an efficient way to numerically compute the integral of the KDE in (6.12) using the Gauss error function. In the context of gas networks, this approach has already been presented by the author in collaboration with Schuster, Gugat and Lang in [98, Section 2.3]. Finally, we also consider other possibilities to compute the integral using MATLAB® functions and compare their computation times with each other.

Integrating the KDE k_S over the rectangle R_n leads to the approximated probability

$$\mathbb{P}_S\left(X \in R_n\right) = \frac{1}{S \prod_{j=1}^n h_j} \sum_{i=1}^{S} \frac{1}{\sqrt{2\pi}} \int_{R_n} \prod_{j=1}^{n} \exp\left(-\frac{1}{2} \left(\frac{z_j - X_j^{(i)}}{h_j} \right)^2 \right) \mathrm{d}z. \tag{6.13}$$

Since the KDE k_S is continuous, we can apply Fubini's theorem so that only one-dimensional integrals are contained in the formula:

$$\mathbb{P}_S\left(X \in R_n\right) = \frac{1}{S \prod_{j=1}^n h_j} \sum_{i=1}^{S} \int_{a_1}^{b_1} \cdots \int_{a_n}^{b_n} \prod_{j=1}^{n} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2} \left(\frac{z_j - X_j^{(i)}}{h_j} \right)^2 \right) \mathrm{d}z_n \cdots \mathrm{d}z_1 \tag{6.14}$$

$$= \frac{1}{S \prod_{j=1}^n h_j} \sum_{i=1}^{S} \prod_{j=1}^{n} \int_{a_j}^{b_j} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2} \frac{\left(z_j - X_j^{(i)} \right)^2}{h_j^2} \right) \mathrm{d}z_j. \tag{6.15}$$

From integration by substitution with $t = \tau_{ij}(z_j) = \frac{z_j - X_j^{(i)}}{\sqrt{2}h_j}$, it follows

$$\begin{aligned}\mathbb{P}_S\left(X \in R_n\right) &= \frac{1}{S\prod_{j=1}^n h_j} \sum_{i=1}^{S} \prod_{j=1}^{n} \frac{1}{\sqrt{2\pi}} \int_{\tau_{ij}(a_j)}^{\tau_{ij}(b_j)} \exp\left(-t^2\right) \sqrt{2} h_j \, \mathrm{d}t \\ &= \frac{1}{S} \sum_{i=1}^{S} \prod_{j=1}^{n} \frac{1}{\sqrt{\pi}} \int_{\tau_{ij}(a_j)}^{\tau_{ij}(b_j)} \exp\left(-t^2\right) \, \mathrm{d}t.\end{aligned}$$

The integrals of the last formula can be formulated by the Gauss error function [3]

$$\operatorname{erf}(x) := \frac{2}{\sqrt{\pi}} \int_0^x \exp(-t^2) \, \mathrm{d}t \tag{6.16}$$

and therefore, we finally get

$$\mathbb{P}_S\left(X \in R_n\right) = \frac{1}{S\ 2^n} \sum_{i=1}^{S} \prod_{j=1}^{n} \big(\operatorname{erf}(\tau_{ij}(b_j)) - \operatorname{erf}(\tau_{ij}(a_j))\big).$$

The Gauss error function is a common function in several fields of mathematics (e.g. probability theory) and physics. Usually, this function is already implemented in a programming language such that an accurate approximation is provided. Therefore, we expect that the required computation gets significantly faster.

In MATLAB®, the Gauss error function can be computed by the MATLAB® function `erf`. Next, we consider three other MATLAB® function which can also be used to compute the approximated probability $\mathbb{P}_S(X \in R_n)$:

- `mvksdensity`:
  The MATLAB® function `mvksdensity`$(x, \mathcal{X}, \text{'Bandwidth'}, [h_1, \ldots, h_n], \text{'Function'}, \text{'cdf'})$ computes the integral

  $$\int_{-\infty}^{x_1} \cdots \int_{-\infty}^{x_n} k_S(z) \, \mathrm{d}z_n \cdots \mathrm{d}z_1,$$

  where k_S is the KDE based on the Gaussian kernel, the data $\mathcal{X}$ and the bandwidth matrix $H = \operatorname{diag}(h_1^2, \ldots, h_n^2)$. In order to compute the integral of the KDE over the n-dimensional rectangular R_n, it is necessary to call this MATLAB® function for different suitable values x.

- `mvncdf`:
  The MATLAB® function `mvncdf`(x_l, x_u, μ, Σ) computes the cumulative distribution function of the multivariate Gaussian distribution with mean $\mu \in \mathbb{R}^n$ and covariance matrix Σ over the n-dimensional rectangle $\tilde{R}_n = [x_l(1), x_u(1)] \times \cdots \times [x_l(d), x_u(d)]$ with $x_l, x_u \in \mathbb{R}^n$, i.e.

  $$\int_{\tilde{R}_n} \frac{1}{\det(\Sigma)\sqrt{(2\pi)^n}} \exp\left(-\frac{1}{2}(z-\mu)\Sigma^{-1}(z-\mu)^T\right) \mathrm{d}z.$$

  Choosing $\tilde{R}_n = R_n$, $\mu = X^{(i)}$ and $\Sigma = \operatorname{diag}(h_1^2, \ldots, h_n^2)$, the function `mvncdf` computes the n-dimensional integrals in (6.13).

- `normcdf`:
  The MATLAB® function `normcdf`(x, μ, σ) computes the evaluation of the cumulative distribution function of the Gaussian distribution with mean μ and standard deviation $\sigma > 0$ at value x, i.e.

$$\frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{x} \exp\left(-\frac{1}{2}\frac{(z-\mu)^2}{\sigma^2}\right) \mathrm{d}z.$$

  Since it holds $\int_{a_j}^{b_j} = \int_{-\infty}^{b_j} - \int_{-\infty}^{a_j}$, we can compute each integral in (6.15) using this MATLAB® function with input parameters $x \in \{a_j, b_j\}$, $\mu = X^{(i)}$ and $\sigma = h_j$.

For the four approaches presented above, we compare the computation time needed to compute the approximation $\mathbb{P}_S(X \in R_n)$. We consider a random variable $X \sim \mathcal{N}(\mu, \Sigma)$ with mean $\mu \in \mathbb{R}^n_+$ and a full, positive definite covariance matrix $\Sigma \in \mathbb{R}^{n\times n}$. We generate samples from this Gaussian distribution and consider the KDE using the Gaussian kernel and the bandwidth matrix $H = \mathrm{diag}(h_1^2, \ldots, h_n^2)$ with h_i given in (6.8). All calculations have been performed with MATLAB® version R2022a on a Intel(R) Xeon(R) Gold E5-4650 CPU running at 2.7 GHz.

In Table 6.3, we compare the approaches for dimension $n = 2, 3, 4$ using $6 \cdot 10^8$ samples. As expected, the higher the dimension n, the larger the computation time for each MATLAB® function. Using the MATLAB® function `erf`, we get the fastest computation which is about 4-times faster than using `normcdf`. In the case of `mvksdensity` and `mvncdf`, we notice that the time grows faster with the dimension than for `erf` and `normcdf`. We suppose that this difference results from computing n-dimensional integrals instead of one-dimensional integrals as in `erf` and `normcdf`. The absolute value of the difference between the approximations of $\mathbb{P}_S$ was less than 10^{-7}. Because of the short computation time, we recommend to use the MATLAB® function `erf` to compute $\mathbb{P}_S$.

MATLAB® Fct.	Dimension n		
	2	3	4
`mvksdensity`	308.39 s	704.69 s	1538.1 s
`erf`	61.16 s	89.49 s	116.16 s
`normcdf`	259.67 s	378.68 s	489.64 s
`mvncdf`	328.38 s	1188.12 s	$>$ 1 h

Table 6.3: Computation times which are needed to compute the approximation $\mathbb{P}_S(k_S \in R_n)$. We consider four different MATLAB® functions and the dimension $n = 2, 3, 4$. The KDE k_S is based on $6 \cdot 10^8$ samples drawn from a Gaussian distribution with full covariance matrix.

If we use a full bandwidth matrix with the Gaussian product kernel, the computations are similar except that the MATLAB® function `mvksdensity` cannot be used because it is implemented only for diagonal bandwidth matrices. In general, the use of a KDE with a product kernel instead of a radially symmetric kernel simplifies the computation of the approximated probability (6.12) since the n-dimensional integral in (6.12) reduces to a product of one-dimensional integrals.

6.5 Boundary Correction Method

In many practical applications, the RV X is bounded since, for example, physical quantities like the pressure or the density of gas cannot be negative. This property is not maintained in the previous presented kernel density estimation approach. If we use a Gaussian kernel, then the resulting KDE has always an unbounded support. For kernels with bounded support, the KDE usually has a larger support than X since the kernel functions are centered at the samples. If bounds of the RVs X are known, we want to take this information into account and to construct an approximation of the PDF with the same support. One naive approach would be to cut the KDE k_S in (6.2) at the bound and then to normalize it such that the new estimate integrates to one again. This strategy is not recommended since it is in general not clear how to scale the truncated KDE accurately. In the following, we consider the simple approach to reflect the KDE k_S at the given bounds [97]. We follow the presentation in [46], but we formulate the correction method also for unsymmetric kernels.

For simplicity, we start with the univariate case $n = 1$ and the KDE k_S with kernel $\mathcal{K}_1$ and bandwidth h, see (6.5). First, we consider the case that the RV has a lower bound $X_*^l \in \mathbb{R}$. The basic idea is that the part of each kernel $\mathcal{K}_1$ which lies in $(-\infty, X_*^l)$ is reflected into the admissible domain $[X_*^l, \infty)$. In Figure 6.2, we illustrate the approach for the kernel function $\mathcal{K}_1(h^{-1}(z - X^{(i)}))$ centered at the sample $X^{(i)}$, close to X_*^l. We reflect this function about the vertical line $z = X_*^l$. Then, we define a new function by adding the kernel and its reflection in the admissible domain and setting to zero otherwise. Since the function $\mathcal{K}_1$ is a kernel, the new function integrates to one and is therefore also a kernel, the so-called *boundary corrected kernel*. In order to get an estimate of the PDF which has the desired support $[X_*^l, \infty)$, we use the boundary corrected kernels instead of the kernels $\mathcal{K}_1$ in the formula (6.5) of the KDE. The resulting estimator k_S^r is called the *reflection estimator* and is equivalent to directly reflecting the KDE k_S at the bound X_*^l. Using the indicator function, the left-side boundary correction is given by

$$\begin{aligned} k_S^r(z) &= \frac{1}{Sh} \sum_{i=1}^{S} \left[\mathcal{K}_1 \left(\frac{z - X^{(i)}}{h} \right) + \mathcal{K}_1 \left(\frac{-z + 2X_*^l - X^{(i)}}{h} \right) \right] \mathbb{1}_{[X_*^l, \infty)}(z) \\ &= \Big(k_S(z) + k_S(-z + 2X_*^l) \Big) \, \mathbb{1}_{[X_*^l, \infty)}(z). \end{aligned}$$

For an upper bound X_*^u, the right-side boundary correction only differs in the subset for which the indicator function is defined:

$$k_S^r(z) = \frac{1}{Sh} \sum_{i=1}^{S} \left[\mathcal{K}_1 \left(\frac{z - X^{(i)}}{h} \right) + \mathcal{K}_1 \left(\frac{-z + 2X_*^u - X^{(i)}}{h} \right) \right] \mathbb{1}_{(-\infty, X_*^u]}(z).$$

The extension for a RV X with image $[X_*^l, X_*^u]$ is straightforward:

$$\begin{aligned} k_S^r(z) = \frac{1}{Sh} \sum_{i=1}^{S} \Bigg[& \mathcal{K}_1 \left(\frac{z - X^{(i)}}{h} \right) + \mathcal{K}_1 \left(\frac{-z + 2X_*^l - X^{(i)}}{h} \right) \\ & + \mathcal{K}_1 \left(\frac{-z + 2X_*^u - X^{(i)}}{h} \right) \Bigg] \mathbb{1}_{[X_*^l, X_*^u]}(z). \end{aligned} \tag{6.17}$$

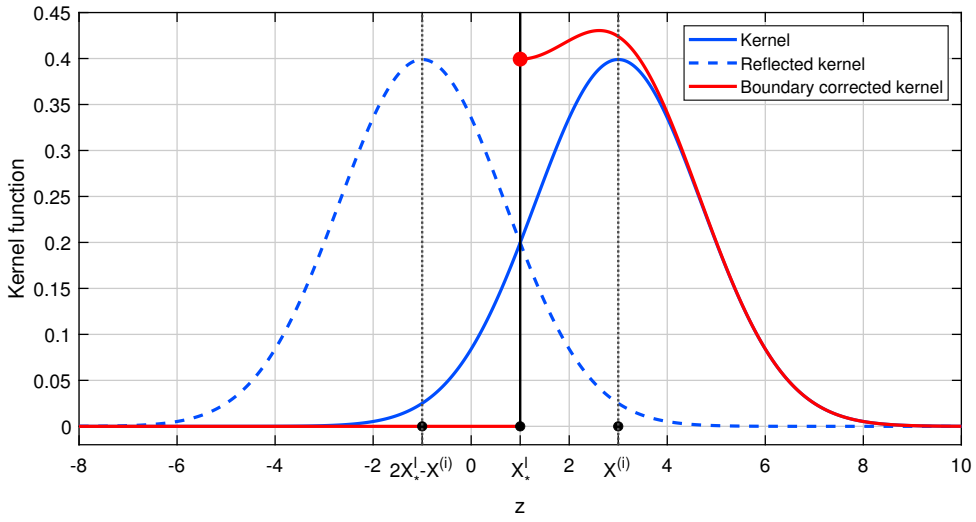


Figure 6.2: Construction of the boundary corrected kernel. A Gaussian kernel (blue) centered at the sample $X^{(i)} = 3$ is reflected about $z = X_*^l = 1$. The reflected kernel (dashed blue) is centered at $2X_*^l - X^{(i)} = -1$. In $[X_*^l, \infty)$, the boundary corrected kernel (red) is the sum of the kernel and its reflection. Otherwise, the new kernel (red) vanishes.

In the multivariate case, the boundary correction of the KDE k_S given in (6.2) is performed analogously. For illustration, we consider the case of a two-dimensional RV X which takes values in $A := [L, \infty) \times (-\infty, U]$ with $L, U \in \mathbb{R}$. The reflection estimator has the form

$$k_S^r(z) = \Big(k_S(z_1, z_2) + k_S(z_1, -z_2+2U) + k_S(-z_1+2L, z_2) + k_S(-z_1+2L, -z_2+2U)\Big) \mathbb{1}_A(z) \quad (6.18)$$

with $z = (z_1, z_2)$.

Furthermore, the reflection estimators k_S^r are PDFs on $\mathbb{R}^n$ since they are non-negative and integrable functions which integrate to one, similar to (6.3). For the product kernel approach, generating the estimator k_S^r by reflecting the d-dimensional kernels $\mathcal{K}_n$ is equivalent to reflecting the one-dimensional kernels $\mathcal{K}_1$.

As in Section 6.4, the reflection estimator k_S^r can be used to approximate probabilities:

$$\mathbb{P}_S(X \in B) = \int_B k_S^r(z)\, \mathrm{d}z \approx \mathbb{P}(X \in B)$$

for a Borel set $B \subset \mathbb{R}^n$. If we use the Gaussian kernel and a diagonal bandwidth matrix $H = \operatorname{diag}(h_1^2, \ldots, h_n^2)$ with $h_i > 0$, then we can efficiently compute the approximated probability with the Gauss error function (6.16). For this case, the previous two-dimensional estimator k_S^r can be written as

$$k_S^r(z) = \frac{1}{S \ \det(H)^{1/2}} \sum_{i=1}^{S} \prod_{j=1}^{2} \mathcal{K}_1\left(\frac{z_j - X_j^{(i)}}{h_j}\right) + \mathcal{K}_1\left(\frac{-z_j + 2X_{*,j} - X_j^{(i)}}{h_j}\right) \mathbb{1}_{A_j}(z_j)$$

with $X_* = (L, U)$, $A_1 := [L, \infty)$, $A_2 := (-\infty, U]$ and the Gaussian kernel $\mathcal{K}_1$. The approximation

of the probability that X takes values in $[a_1, b_1] \times [a_2, b_2]$ is now given by

$$\mathbb{P}_S(X \in [a_1, b_1] \times [a_2, b_2]) = \frac{1}{S\ 2^n} \sum_{i=1}^{S} \Big(\operatorname{erf}(\tau_{i1}(b_1)) - \operatorname{erf}(\tau_{i1}(\widetilde{a}_1)) + \operatorname{erf}(\widehat{\tau}_{i1}(\widetilde{b}_2)) - \operatorname{erf}(\widehat{\tau}_{i1}(a_2))\Big) \cdot \Big(\operatorname{erf}(\tau_{i2}(b_1)) - \operatorname{erf}(\tau_{i2}(\widetilde{a}_1)) + \operatorname{erf}(\widehat{\tau}_{i2}(\widetilde{b}_2)) - \operatorname{erf}(\widehat{\tau}_{i2}(a_2))\Big)$$

with $\tau_{ij} = \frac{z_j - X^i}{\sqrt{2}h_j}$, $\widehat{\tau}_{ij} = \frac{-z_j + 2X_{*,j} - X^i}{\sqrt{2}h_j}$, $\widetilde{a}_1 = \max(a_1, L)$ and $\widetilde{b}_2 = \min(b_2, U)$.

6.6 Post-Processing Step of SC Methods: Feasibility Check

The SC methods presented in Chapter 5 can provide an approximation $\widetilde{\Phi}$ of a real-valued function $\Phi : \Gamma \to \mathbb{R}$, also called the quantity of interest (QoI). If the approximated QoI $\widetilde{\Phi}$ is computed by a general sparse grid (Subsection 5.1.2), we have $\widetilde{\Phi} = \mathcal{G}_\Lambda[\Phi]$. For the single-level (Section 5.2) and the multi-level approach (Section 5.3), we have

$$\widetilde{\Phi} = \Phi_h^{(\mathrm{SL})} \quad \text{and} \quad \widetilde{\Phi} = \Phi_K^{(\mathrm{ML})},$$

respectively. We recall that stochastic collocation (SC) methods deal with parametrized problems obtained by a transformation from a probability space to the image space Γ of the underlying random vector $\xi : \Theta \to \Gamma$.

Under the assumption that the QoI Φ has a PDF ρ_Φ, we apply the KDE to approximate its PDF in a post-processing step. First, we independently draw S samples $\mathbf{y}_i$ of the random vector ξ according to its given distribution. At each sample, we evaluate the SC approximation $\widetilde{\Phi}$, i.e. $\widetilde{\Phi}^{(i)} := \widetilde{\Phi}(\mathbf{y}_i)$. This evaluation is extremely cheap and has the advantage that no further deterministic problems have to be solved. Now, we use the generated sampling set $\mathcal{X} = \{\widetilde{\Phi}^{(1)}, \ldots, \widetilde{\Phi}^{(S)}\}$ to compute the recommended KDE k_S with Gaussian kernel and diagonal bandwidth matrix given in (6.10). In order to consider the feasibility of the QoI, we require the probability that the QoI takes values between the prescribed lower and upper bound. As described in Section 6.4, we approximate this probability by integrating the KDE k_S over the given bounds. Since we focus on the KDE (6.10), the usage of the Gauss error function offers an efficient way to compute the probability as pointed out in Subsection 6.4.1. If the QoI is a bounded RV with known bounds, the application of the boundary correction method provides an approximated PDF with the same support as the exact PDF ρ_Φ, see Section 6.5.

Another choice of the QoI in SC methods is the full solution u. In this case, the KDE can be applied to a vector-valued function $\Psi(\widetilde{u}) : \Gamma \to \mathbb{R}^n$, where $\widetilde{u}$ is the solution approximated by an SC method. For example, the minimum and maximum pressure at the exits over the time period. Obviously, the application is similar to the previous case of a real-valued QoI.

To some extent, the PDF can also be approximated in a post-processing step of MC methods. The samples of the QoI which are already computed can be obviously reused. But if more samples are needed, we have to compute new samples by numerically solving further deterministic problems. These computations can be expensive, especially for complex problems.

Chapter 7

Application to Uncertain Gas Transport

In this chapter, we apply the adaptive multi-level stochastic collocation (MLSC) method to uncertain gas transport through pipeline networks and compare the result to the single-level stochastic collocation (SLSC) approach and to the adaptive Monte Carlo methods. In a post-processing step, we use the kernel density estimator (KDE) approach described in Chapter 6 to validate the feasibility of the pressure at the exit nodes. We consider the two gas network instances: GasLib-11 and GasLib-40, parts of the real German gas network, which are taken from the public gas library gaslib.zib.de [96]. The deterministic settings are extended by uncertain gas demands at the exit nodes.

As deterministic scenario, we consider a smooth transformation from an initial stationary state u_A to a new stationary state u_B which differs mainly in the gas demands. For such scenarios, suitable optimization tools can provide a feasible operational control of the compressor stations and valves such that for example lower and upper pressure bounds are fulfilled over the whole selected time period. These conditions and penalties in case of violation are specified in contracts between gas company and consumer. Therefore, we assume that a feasible and optimized control for each network is given which provides the pressure jumps, realized by the compressors, and the status of the valves (open/closed). Deterministic scenarios are now extended by uncertain gas demand at the exit nodes in the final state u_B while the optimal control is fixed. We will investigate the influence of these uncertainties on the compressor costs and validate the pressure of the delivered gas.

For GasLib-11 and GasLib-40, our developed methods were recently applied to such scenarios with uncertain gas transport by the author in collaboration with Lang et al. [68]. The uncertain gas demands were modeled linearly in the stochastic variables and the stochastic collocation (SC) methods outperformed the MC methods. Moreover, the single-level SC approach performed already very efficiently since extremely few collocation points were sufficient to obtain the desired accuracies. Therefore, we suppose that the considered quantity of interest (QoI) depended almost linearly on the stochastic variables. In this thesis, we want to investigate the potential of our adaptive MLSC method for uncertain gas demands which depend nonlinearly on the stochastic variables. The scenarios considered in this chapter are therefore based on [68].

The implementation of the adaptive MLSC and MLMC methods are described in Sections 4.4 and 5.5. In both cases, it is based on the deterministic black box solver ADET which computes an approximate solution of the deterministic network problem (2.11)–(2.15) such that the corresponding error is reduced up to a prescribed accuracy. For the modeling of gas transport in pipelines, we use the model hierarchy of the Euler equations combined with suitable boundary

and coupling conditions which is completely presented in Section 2.5. The adaptive algorithm described in Subsection 2.5.4 is implemented in our in-house software package ANACONDA and therefore is used as solver ADET. For details of the implementation of ANACONDA, we refer to [62]. Furthermore, the adaptive sparse grid algorithm ASPAGRID needed for the SC implementations is provided by the open-source MATLAB® package *Sparse Grid Kit* [104] which we slightly modified, see Subsection 5.1.6. We have performed all calculations on a Intel(R) Xeon(R) Gold 6130 CPU running at 2.1 GHz and worked with MATLAB® version R2022a.

7.1 A Small Gas Network with Compressor Stations and Valve (GasLib-11)

As first example, we consider the GasLib-11 which is illustrated in Figure 7.1. This network has 3 sources, 3 exits, 2 compressor stations, 1 valve and 8 pipes which have all the same length of 55 km [96]. For a well-arranged illustration of the network in Section 2.5, edges of length 0, so-called short edges, are used. This artificial construct can in general be easily integrated into numerical solvers and even simplify its implementation by using the algebraic equations (2.20). In order to obtain a network problem as described in Section 2.5 which contains only edges of length $L > 0$, we remove the short edges and merge all nodes connected continuously by short edges into a new single node with a corresponding combined coupling condition. The pipes, i.e. edges of positive length, which are connected to these nodes are maintained. For the network in Figure 7.1, this degeneration would concern the nodes $S2, C1, J1, V1, J4$ as well as $J3, C2, J5$ identified by new nodes with ingoing pipes e_1, e_8 and e_4, e_5 as well as outgoing edges e_2, e_5 and e_6, e_7, respectively.

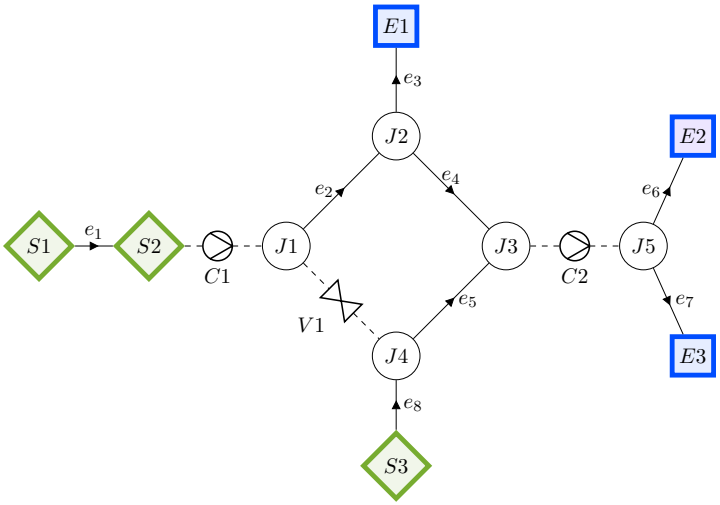


Figure 7.1: Schematic description of the gas network GasLib-11 with 8 pipes (lines with an arrow: $e_1 - e_8$), 3 sources (green diamonds: $S1, S2, S3$), 3 exits (blue squares: $E1, E2, E3$), 2 compressor stations ($C1, CS$) and 1 valve ($V1$). The dashed lines represent additional edges of length 0 leading to a well-arranged network illustration. The arrows indicate the orientation of the edges and is used to identify the flow direction: If the gas flows against the direction of the edge, then the flow has a negative sign.

The stationary initial state u_A and the final state u_B are defined by the prescribed data given in Table 7.1. The simulation starts with the stationary initial state $u_0 = u_A$ which implies that the quasi-stationary semilinear isothermal Euler equations $\mathcal{M}_3$, see (2.17), are initially chosen

for each pipe. After $t = 4$h, the prescribed values for the exits, sources and control of the compressor stations are linearly changed such that the new values determined by the final state u_B are reached at $t = 6$h. The initially open valve is closed at time point $t = 4.5$h. The total simulation time is $T = 24$h with initial time step size $\Delta t = 1$h and initial spatial step size $\Delta x = 10,000$m. We divide the time interval $[0, T]$ into 6 time blocks of 4h and perform the adaptive strategy described in Subsection 2.5.4.

Prescribed data	State u_A ($t = 0 - 4$h)			State u_B ($t = 6 - 24$h)		
Source	$S1$	$S2$	$S3$	$S1$	$S2$	$S3$
Pressure p [bar]	70	65	70	48	46	54
Exit	$E1$	$E2$	$E3$	$E1$	$E2$	$E3$
Flow rate q [m^3/s]	38	38	38	23.6	23.6	23.6
Compressor station	$C1$	$C2$		$C1$	$C2$	
Pressure jump Δp [bar]	0	0		5	15	
Valve	$V1$			$V1$		
Status	open			closed		

Table 7.1: *GasLib-11*: Prescribed data for sources $(S1, S2, S3)$, exits $(E1, E2, E3)$ and control data for compressor stations $(C1, C2)$, valve $(V1)$ for the initial state u_A and final state u_B.

We extend the deterministic final state u_B by uncertain gas demand at all three exits $E1, E2, E3$. The controls and the remaining boundary conditions remain deterministic, including the initial value. We assume that the three consumers behave independently from each other such that we parametrize the uncertain behavior by the stochastic variable $\mathbf{y} = (y_1, y_2, y_3) \in [-1, 1]^3 =: \Gamma$ which is a realization of the random vector $\xi = (\xi_1, \xi_2, \xi_3) : \Theta \to \Gamma$. The random vector ξ consists of three independent and uniformly distributed random variables (RVs) $\xi_i \sim \mathcal{U}[-1, 1]$. In detail, we modify the prescribed deterministic gas demands

$$q_d^{Ei}(t) = \begin{cases} 38 & \text{for } t < 4\text{h}, \\ 38 - 7.2(t-4) & \text{for } 4\text{h} \le t \le 6\text{h}, \\ 23.6 & \text{for } t > 6\text{h} \end{cases}$$

to uncertain conditions

$$q^{Ei}(t, \mathbf{y}) = \begin{cases} 38 & \text{for } t < 4\text{h}, \\ 38 + (t-4)\big(-12.2 + 10(\exp(y_i - 1) + 0.5\exp(-2))\big) & \text{for } 4\text{h} \le t \le 6\text{h}, \\ 13.6 + 20\big(\exp(y_i - 1) + 0.5\exp(-2)\big) & \text{for } t > 6\text{h} \end{cases} \tag{7.1}$$

for $t \in [0, T], \mathbf{y} \in \Gamma = [-1, 1]^3$ and $i = 1, 2, 3$. Consequently, the uncertain gas demands at the exits have the same distribution and are stochastically independent of each other. The expected values of q^{Ei} are equal to the gas demand in the deterministic setting: $\mathbb{E}[q^{Ei}] = q_d^{Ei}$. Figure 7.2 illustrates the deterministic and the uncertain gas demand at the exit nodes.

The extension to uncertainties and its parametrization lead to a parametrized network problem of the form (3.8)–(3.12) where Assumptions 3.4, 3.5 and 3.6 are fulfilled and SC methods can be applied. Without the transformation to the finite-dimensional space Γ, we obtain the classical random network problem of the form (3.3)–(3.7), the basis for MC methods.

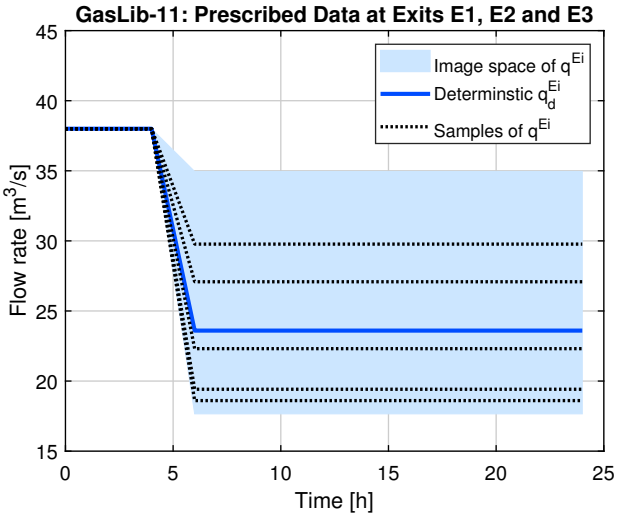


Figure 7.2: *GasLib-11*: Prescribed deterministic gas demand $q_d^{Ei}(t)$ at the exit node Ei for $i = 1, 2, 3$ (blue line). Time-dependent samples of the uncertain gas demand $q^{Ei}(t, \mathbf{y})$ are plotted as black dotted lines. The blue shaded area represents the image space of the uncertain gas demand.

As mentioned in Subsection 2.5.3, we consider pressure-controlled compressor stations which consume only electric power since they have an electric motor as drive. The drive provides the power which is required for the compression of the gas by the station and the electric energy consumed for this process by the drive is given by the specific energy consumption. Therefore, we can interpret the specific energy consumption $G_c(u)$ of the drive needed to power the compressor station c as the corresponding compressor costs. Consequently, we define the deterministic output functional (2.21) in the adaptation process as

$$\mathcal{F}[u(\mathbf{y})] = \alpha \sum_{c \in \mathscr{V}_{cs}} \int_0^T G_c\big(u(t, \mathbf{y})\big) \, \mathrm{d}t \qquad \text{for} \quad \mathbf{y} \in \Gamma$$

with the condensed notation $u(t, \mathbf{y}) = (u^{(1)}(\cdot, t, \mathbf{y}), \ldots, u^{(P)}(\cdot, t, \mathbf{y}))$ and a scaling constant $\alpha \in \mathbb{R}$. We denote the set of compressor stations in the network by $\mathscr{V}_{cs}$. Our real-valued QoI $\Phi : \Gamma \to \mathbb{R}$ is then defined by

$$\Phi(\mathbf{y}) = \mathcal{F}[u(\mathbf{y})].$$

The specific energy consumption G_c can be estimated by a quadratic fit

$$G_c\big(u(t, \mathbf{y})\big) = g_{c,0} + g_{c,1} P_c\big(u(t, \mathbf{y})\big) + g_{c,2} P_c\big(u(t, \mathbf{y})\big)^2$$

with given compressor-dependent constants $g_{c,0}, g_{c,1}, g_{c,2}$ and power P_c consumed by the compressor station c [91, 96]. The power P_c required by the compressor station $c \in \mathscr{V}_{cs}$ for the compression of the gas flow is given by

$$P_c\big(u(t, \mathbf{y})\big) = c_{c,P} \, |q_{c,in}(t, \mathbf{y})| \, z\big(p_{c,in}(t, \mathbf{y})\big) \left(\left(\frac{p_{c,out}(t, \mathbf{y})}{p_{c,in}(t, \mathbf{y})} \right)^{\frac{\kappa - 1}{\kappa}} - 1 \right) \tag{7.2}$$

with in- and outgoing pressure $p_{c,in}$, $p_{c,out}$ and ingoing flow rate $q_{c,in}$ [91]. Since the compressor is controlled by the prescribed pressure increase $\Delta p_c(t)$, it holds $p_{c,out}(t, \mathbf{y}) = p_{c,in}(t, \mathbf{y}) + \Delta p_c(t)$. The parameter $c_{c,P}$ is a compressor specific constant, κ the isentropic coefficient of the gas

and $z\big(p_{c,in}(t,\mathbf{y})\big)$ the compressibility factor from the equation of state for real gases. For the compressor stations in GasLib-11, we have $g_{c,0} = 5000, g_{c,1} = 2.5$ and $g_{c,2} = 0$. Moreover, we set $\alpha = 10^{-10}$ in order to scale the expected value of Φ to the order of 0.1.

In order to consider the stochastic dependency of the QoI Φ in advance, we approximate Φ on a $17 \times 17 \times 17$ tensor product grid of Clenshaw-Curtis points for $TolH = 10^{-5}$. We integrate the QoI with respect to a single stochastic variable y_i for all $i = 1, 2, 3$ which allows to visualize well the influence of the uncertainties. The plots in Figure 7.3 show a nonlinear dependency of the QoI on all three stochastic variables. We observe the same dependency on y_2 as on y_3 since the exits $E2$ and $E3$ have the same modeling of uncertain gas demand (7.1) and a similar position in the gas network.

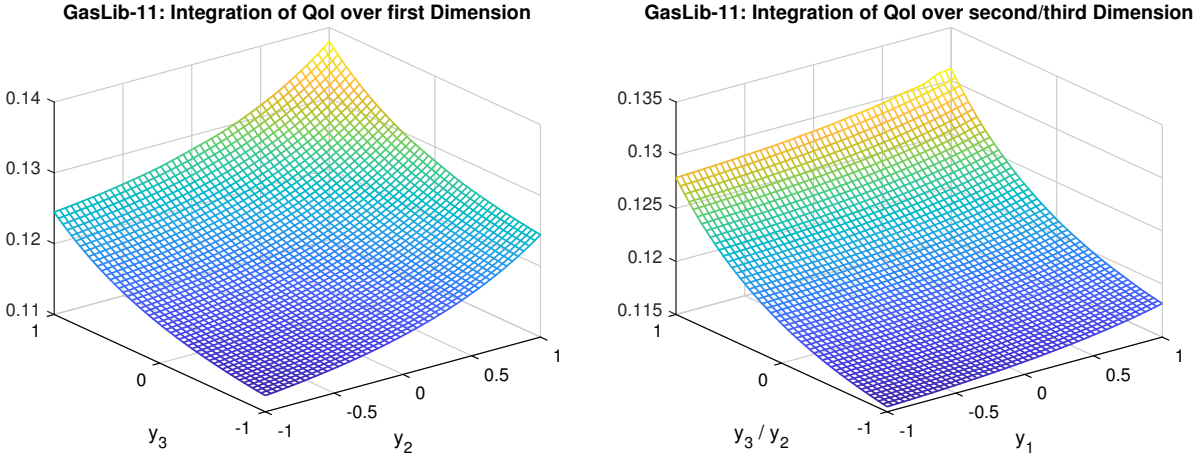


Figure 7.3: *GasLib-11*: Integration of an approximation of the QoI Φ with respect to one stochastic variable $\mathbf{y}_i \in [-1, 1]$ for $i = 1, 2, 3$. Integrating over the second dimension provides the same result as for the third dimension.

Now, we consider our adaptive single-level and our adaptive two-level stochastic collocation method with a reduction factor $q = 0.1$ and Clenshaw-Curtis nodes. The aim is to approximate the expected value of the compressor costs Φ. The implementation of both methods is described by the Algorithms 5.5 and 5.6. First, we start estimating the parameters $C_h, C_H, C_y, C_Y, \gamma$ and μ following the steps in [67].

The quality of the physical error estimation process of ANACONDA is investigated by using physical tolerances $TolH = 10^{-2}, 3 \cdot 10^{-3}, 10^{-3}, 3 \cdot 10^{-4}, 10^{-4}$ and a fixed isotropic Smolyak grid of level $w = 2$ (25 collocation points). We take the expected value of the QoI Φ_{h^*} associated with $TolH^* = 10^{-5}$ as reference value and approximate the physical errors by

$$|\mathbb{E}[\Phi] - \mathbb{E}[\Phi_h]| \approx |\mathcal{E}_{\Lambda^{\mathrm{SMO}}(w)}[\Phi_{h^*}] - \mathcal{E}_{\Lambda^{\mathrm{SMO}}(w)}[\Phi_h]|.$$

Furthermore, we calculate the average of the physical error estimates over the selected isotropic sparse grid and show the results with respect to the computation time in Figure 7.4. We note that the prescribed physical tolerances are always satisfied. In contrast, the error estimate does not always provide an upper bound for the numerical error, but they are very close to each other. For (5.35), we estimate $C_H = C_h = 0.1$. However, we consider the average of the ratios between the error and the corresponding tolerances instead of the maximum value, since we observe smaller ratios for smaller tolerances. A least-squares fit provides the rate $\gamma = 0.8$ in Assumption 5.22 (v).

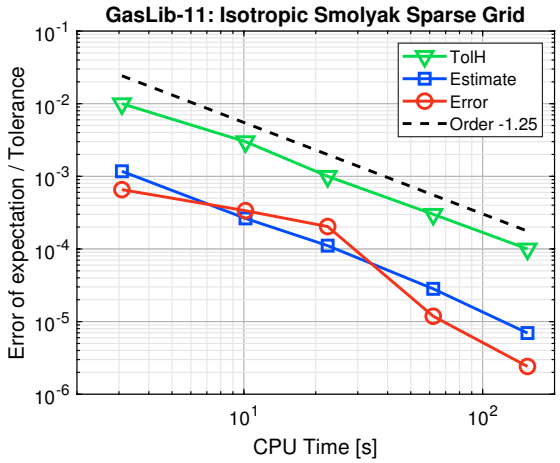


Figure 7.4: *GasLib-11*: Physical errors and averaged physical error estimates using physical tolerances $TolH = 10^{-2}, 3 \cdot 10^{-3}, 10^{-3}, 3 \cdot 10^{-4}, 10^{-4}$ and a fixed isotropic Smolyak sparse grid of level $w = 2$. The estimated order of convergence for $TolH$ in terms of computation time is $-1/\gamma = -1.25$.

The constant C_Y and rate μ which characterize the convergence rate and the quality of the error estimates of the algorithm ASpaGrid are estimated by choosing the stochastic tolerances $TolY = 3 \cdot 10^{-4}, 10^{-4}, 3 \cdot 10^{-5}, 10^{-5}$ and a fixed physical tolerance $TolH_{fix} = 10^{-4}$. We take the expected value of the QoI $\Phi_{h_{fix}}$ associated with $TolY^* = 10^{-5}$ as reference value and approximate the stochastic errors by

$$\left|\mathbb{E}[\Phi_h] - \mathbb{E}\left[\mathcal{A}_Q[\Phi_h]\right]\right| \approx |\mathcal{E}_{Q^*}[\Phi_{h_{fix}}] - \mathcal{E}_Q[\Phi_{h_{fix}}]|.$$

In Figure 7.5, we show the results with respect to the number of collocation points. The prescribed stochastic tolerances are always satisfied and the error estimates provide always an upper bound for the error. However, the ratio between the error and the error estimate is about one order of magnitude. Similar to the estimation of C_H, we choose $C_Y = C_y = 0.05$ by averaging the ratio between the error and the stochastic tolerance. A least-squares fit provides the rate $\mu = 2.4$ in Assumption 5.22 (iv).

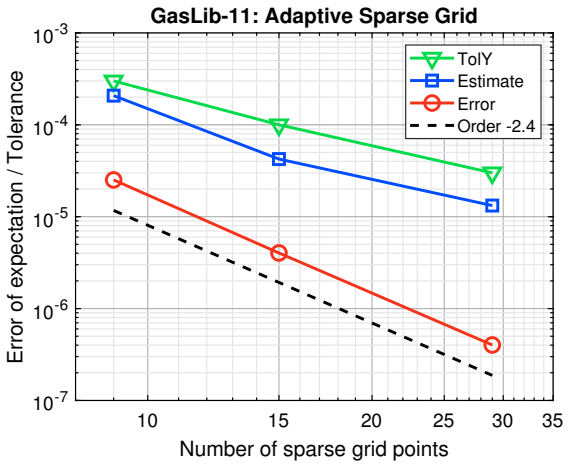


Figure 7.5: *GasLib-11*: Adaptive sparse grid algorithm ASpaGrid with stochastic tolerances $TolY = 3 \cdot 10^{-4}, 10^{-4}, 3 \cdot 10^{-5}$ and fixed physical tolerance $TolH = 10^{-4}$. The estimated order of convergence for $\mathbb{E}[\Phi]$ in terms of collocation points is -2.4.

Next, we run the single-level and the two-level stochastic collocation method for accuracy requirements of $\epsilon = 10^{-4}, 10^{-5}, 10^{-6}$. We consider the errors

$$|\mathbb{E}[\Phi] - \mathbb{E}[\Phi_h^{(\mathrm{SL})}]| \approx |\mathcal{E}_{Q_{ref}}[\Phi_{h_{ref}}] - \mathcal{E}_Q[\Phi_h]|,$$
$$|\mathbb{E}[\Phi] - \mathbb{E}[\Phi_K^{(\mathrm{ML})}]| \approx |\mathcal{E}_{Q_{ref}}[\Phi_{h_{ref}}] - \mathcal{E}_K^{(\mathrm{ML})}[\Phi_{h_K}]|,$$

where we compute a reference solution $\mathcal{E}_{Q_{ref}}[\Phi_{h_{ref}}] = 0.120643668591124$ by the single-level SC method with $\epsilon = 10^{-7}$. The computation of the reference solution requires 73 collocation points. The physical tolerances are given by $TolH_k = \epsilon\, q^{k-1}/(2\, C_H)$ with $C_H = 0.1$ and the stochastic tolerances by (5.28) with $C_Y = 0.05, \mu = 2.4$, and $\gamma = 0.8$ for $k = 0, 1$. The number of the samples taken by the SC methods are given in Table 7.2. The ML approach needs less samples on the fine level $k = 1$ than on the coarse level $k = 0$. The top-left plot of Figure 7.6 shows that both methods provide always approximations which have a higher accuracy than the prescribed one, but not more than a factor of 5. This effect could be caused by cancellations which we omit since we split the total error into two parts and control them separately. We observe that the errors of the two-level method are smaller than the errors of the single-level method for the same accuracy requirements. For the lowest accuracy $\epsilon = 10^{-4}$, the single-level approach needs a few seconds less, but the error of the two-level estimation is smaller. Both approaches perform very reliably. The single-level SC method works already sufficiently fast for low accuracies. For high accuracies, the two-level method needs significantly less computation time - around a factor of 1/3 faster. We would expect a higher difference in the computation time, especially for lower accuracy levels, if the needed computation time of the deterministic solver would be larger and would increase faster with accuracy.

ϵ	SLSC	Two-Level SC	
	Q	Q_0	Q_1
10^{-4}	7	15	7
10^{-5}	25	25	7
10^{-6}	41	41	7

ϵ	SLMC	Two-Level MC	
	M	M_0	M_1
10^{-3}	42	83	10
$3 \cdot 10^{-4}$	284	988	41
10^{-4}	5,539	6,623	179
$3 \cdot 10^{-5}$	51,423	69,259	1,747

Table 7.2: *GasLib-11:* Number of collocation points used by the SLSC and two-level SC methods (left) and averaged number of samples used by the SLMC and two-level MC methods (right).

The theoretical orders of the computational cost in terms of the accuracy ϵ are given in Section 5.4: $r_{\mathrm{MLSC}} = -\gamma = -0.8$ and $r_{\mathrm{SLSC}} = -\gamma - 1/\mu \approx -1.22$. The top-right plot of Figure 7.6 shows that the theoretical rates approximate the observed asymptotic rates quite well. Considering the error in terms of the computational cost in the top-left plot of Figure 7.6, the orders of convergence are predicted by $p_{\mathrm{MLSC}} = 1/r_{\mathrm{MLSC}} \approx -1.25$ and $p_{\mathrm{SLSC}} = 1/r_{\mathrm{SLSC}} \approx -0.82$. A least-squares fit of our results provides slightly smaller orders: $\widetilde{p}_{\mathrm{MLSC}} = -0.94$ and $\widetilde{p}_{\mathrm{SLSC}} = -0.73$. We expect that the difference between the observed and the theoretical convergence orders would be smaller, if we would have reliable a posteriori error estimators instead of heuristic error indicators.

Next, we perform the adaptive single-level and the adaptive two-level ($K = 1$) Monte Carlo method following the implementation in Subsection 4.4.1. We will compare the results with the previous SC approximations. The estimates for the physical constants C_H, C_h and rate γ which we approximated for the SC methods can be reused since the deterministic solver does not change. We choose $M_{init} = 10$ as the number of initial samples which are used to estimate the variance in (4.26) as well as in (4.31) and thus the number of samples at each level.

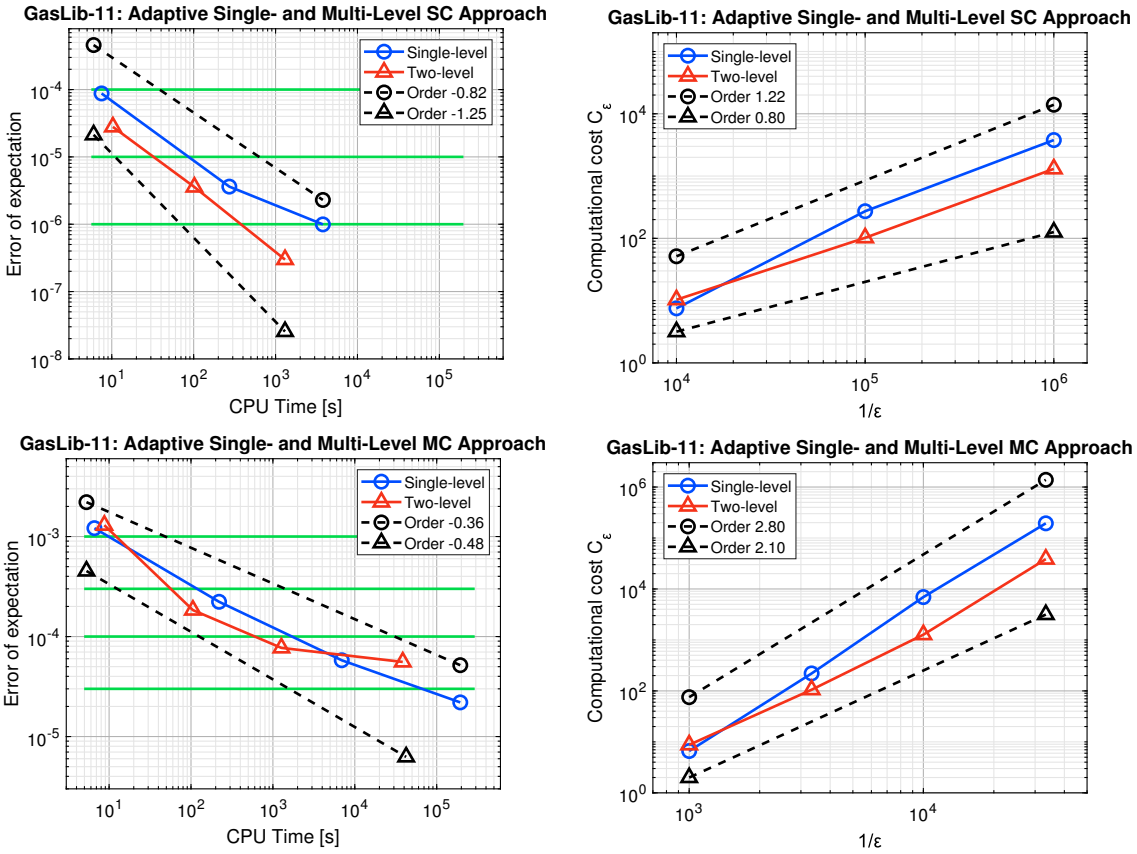


Figure 7.6: *GasLib-11*: Top Left: Errors of the approximated expectations $\mathcal{E}_Q[\Phi_h]$ and $\mathcal{E}_K^{(\mathrm{ML})}[\Phi_{h_K}]$ obtained by the adaptive single-level (blue circles) and two-level (blue triangles, $K = 1$) SC methods for $\epsilon = 10^{-4}, 10^{-5}, 10^{-6}$ (green lines). The predicted convergence orders are -0.82 and -1.25, respectively (dashed black lines). Top Right: Computational cost of the SC methods versus the inverse of the prescribed accuracy. Bottom Left: Mean absolute errors of the approximated expectations $E_M[\Phi_h]$ and $E^K[\Phi_{h_K}]$ obtained by the adaptive single-level (blue circles) and two-level (blue triangles, $K = 1$) MC methods for $\epsilon = 10^{-3}, 3 \cdot 10^{-4}, 10^{-4}, 3 \cdot 10^{-5}$ (green lines). The predicted convergence orders are -0.36 and -0.48, respectively (dashed black lines). Bottom Right: Computational cost of the MC methods versus the inverse of the prescribed accuracy.

In order to compare the SC and MC approaches more precisely, we consider the mean absolute error instead of the mean squared error analyzed in the theory part. Due to (4.27), the mean absolute error fulfills the accuracy requirements as well. As reference value for $\mathbb{E}[\Phi]$, we use again the SLSC approximation $\mathcal{E}_{Q_{ref}}[\Phi_{h_{ref}}]$ computed for $\epsilon = 10^{-7}$. We approximate the mean of the absolute error by an MC estimate with 10 different realizations of $E_M[\Phi_h]$ and $E^K[\Phi_{h_K}]$:

$$\mathbb{E}\big[|\mathbb{E}[\Phi] - E_M[\Phi_h]|\big] \approx \widehat{E_{10}}\big[|\mathcal{E}_{Q_{ref}}[\Phi_{h_{ref}}] - E_M[\Phi_h]|\big] = \frac{1}{10}\sum_{m=1}^{10} |\mathcal{E}_{Q_{ref}}[\Phi_{h_{ref}}] - \widehat{E_M}[\Phi_h]|,$$

$$\mathbb{E}\big[|\mathbb{E}[\Phi] - E^K[\Phi_{h_K}]|\big] \approx \widehat{E_{10}}\big[|\mathcal{E}_{Q_{ref}}[\Phi_{h_{ref}}] - E^K[\Phi_{h_K}]|\big] = \frac{1}{10}\sum_{m=1}^{10} |\mathcal{E}_{Q_{ref}}[\Phi_{h_{ref}}] - \widehat{E^K}[\Phi_{h_K}]|. \tag{7.3}$$

Consequently, we compute 10 independent realizations of the MC and MLMC estimator which means that we perform 10 simulations of both algorithms. The averaged number of the samples

used by the MC methods are given in Table 7.2. We observe that the MLMC method needs significantly less samples on the fine level $k = 1$ than on the coarse level $k = 0$.

The bottom-left plot of Figure 7.6 shows the SLMC and MLMC error for accuracy $\epsilon = 10^{-3}$, $3 \cdot 10^{-4}, 10^{-4}, 3 \cdot 10^{-5}$. For both methods, the achieved accuracy is almost always better than the prescribed one. The computation time needed for the ML approximations is smaller than for the SL approach, except for the lowest accuracy. For the highest accuracy, the MLMC method provides nearly the same approximation as for the next lower accuracy while increasing significantly the computation time. This inefficient behavior can occur if the error indicators underestimate the exact errors. Comparing the bottom-left plot with the top-left plot of Figure 7.6, we see that the SC methods significantly outperform the MC methods.

The theoretical rate of the computational cost for the SL approach is given by Lemma 4.23: $r_{\text{SLMC}} = -2 - \gamma = -2.8$. In order to get the theoretical rate for the MLMC method in Theorem 4.29, we estimate the rate β in Assumption 4.27 a posteriori. We average the variance over the 10 simulations and obtain $\beta = 0.7$ by a least-squares fit. Consequently, we have $r_{\text{MLMC}} = -2 - (\gamma - \beta) = -2.1$. In the bottom-right plot of Figure 7.6, the theoretical rates of the computational cost in terms of the prescribed accuracy are clearly visible which points the improvement of ML structures out. Note that the observed rate β differs from the result of Lemma 4.30 where $\beta = 2$ was proved. One possible reason could be a slow decrease of the variance in Assumption 4.27 caused by switching the models in the deterministic solver.

7.1.1 Feasibility Check of the Pressure at the Exits

In order to check the feasibility of the pressure at the three exit nodes $E1, E2$ and $E3$, we want to compute the probability that the pressure bounds are satisfied at all exits simultaneously:

$$\mathbb{P}\left(p^{Ei}(t,\cdot) \in [p^{\min}, p^{\max}] \text{ for all } t \in [0,T] \text{ and } i \in \{1,2,3\}\right). \tag{7.4}$$

We denote the lower pressure bound by $p^{\min} \in \mathbb{R}_+$ and the upper bound by $p^{\max} \in \mathbb{R}_+$. The previous joint probability specifies how likely the contractual pressure bounds are violated for at least one of the three exits. In order to compute the probability, we will use the KDE approach presented in Chapter 6 where we also briefly described the main aspects of its application in a post-processing step of SC methods (Section 6.6). In order to keep the computational cost low, we generate the required samples of the pressures at the exits by evaluating the corresponding SC approximation $\widetilde{p}^{Ei}$ of p^{Ei} which are computed by our SC method.

Using the SC approximations, the probability (7.4) can be rewritten as

$$\mathbb{P}\left(\min_{t\in[0,T]} \widetilde{p}^{Ei}(t,\cdot) \in [p^{\min}, p^{\max}] \text{ and } \max_{t\in[0,T]} \widetilde{p}^{Ei}(t,\cdot) \in [p^{\min}, p^{\max}] \ \forall i \in \{1,2,3\}\right). \tag{7.5}$$

The computation of the previous probability requires the probability density function (PDF) of the six-dimensional random vector

$$X := \left(\underline{p}^{E1}, \overline{p}^{E1}, \underline{p}^{E2}, \overline{p}^{E2}, \underline{p}^{E3}, \overline{p}^{E3}\right),$$

where we set

$$\underline{p}^{Ei}(\mathbf{y}) = \min_{t\in[0,T]} \widetilde{p}^{Ei}(t,\mathbf{y}) \quad \text{and} \quad \overline{p}^{Ei}(\mathbf{y}) = \max_{t\in[0,T]} \widetilde{p}^{Ei}(t,\mathbf{y}), \qquad i = 1,2,3.$$

A short formulation of probability (7.5) is therefore $\mathbb{P}(X \in [p^{\min}, p^{\max}]^6)$. In order to check whether the dimension of the PDF of X can be reduced, we compute the 6 single probabilities

$$\mathbb{P}\left(\underline{p}^{Ei} \in [p^{\min}, p^{\max}]\right) \quad \text{and} \quad \mathbb{P}\left(\overline{p}^{Ei} \in [p^{\min}, p^{\max}]\right)$$

for $i = 1, 2, 3$. If a probability is equal to one, then we can remove the corresponding RV from the vector X since the pressure bounds are always satisfied for this single quantity. For dynamic flow network problems with known exact solution, the computation of probability (7.5) based on a KDE has already been discussed by the author in collaboration with Schuster, Gugat and Lang [98, Chapter 3], but without the dimension reduction.

In order to approximate the time-dependent pressure at the three exits, we use the adaptive sparse grid obtained for the compressor costs Φ by our adaptive SLSC method for the accuracy $\epsilon = 10^{-6}$. Since the pressures at the exit nodes influence the compressor cost, we expect that the approximate pressures have a sufficiently adequate accuracy. We denote the multi-index set corresponding to the sparse grid by Λ_Φ. Using the sparse grid formula (5.8) yields the SC approximation

$$\widetilde{p}^{Ei}(t, \mathbf{y}) := \mathcal{G}_{\Lambda_\Phi}\left[p_h^{Ei}(t, \cdot)\right](\mathbf{y}) = \sum_{\mathbf{i} \in \Lambda_\Phi} \Delta_{\mathbf{i}}\left[p_h^{Ei}(t, \cdot)\right](\mathbf{y}) = \sum_{\mathbf{i} \in \Lambda_\Phi} c_{\mathbf{i}} \left(\sum_{\mathbf{j}=\mathbf{1}}^{\mathbf{m(i)}} p_h^{Ei}\left(t, \mathbf{y}_{\mathbf{j}}^{\mathbf{i}}\right) \mathcal{L}_{\mathbf{j}}^{\mathbf{i}}(\mathbf{y}) \right), \tag{7.6}$$

where $\mathbf{y}_{\mathbf{j}}^{\mathbf{i}} \in \mathbb{H}_{\Lambda_\Phi}$ are the used collocation points and $c_{\mathbf{i}} \in \mathbb{Z}$ are given by (5.5). Consequently, we can compute the minimum of the SC approximation (7.6) over time:

$$\min_{t \in [0,T]} \widetilde{p}^{Ei}(t, \mathbf{y}) = \min_{t \in [0,T]} \left(\mathcal{G}_{\Lambda_\Phi}\left[p_h^{Ei}(t, \cdot)\right](\mathbf{y})\right) = \min_{t \in [0,T]} \left(\sum_{\mathbf{i} \in \Lambda_\Phi} c_{\mathbf{i}} \left(\sum_{\mathbf{j}=\mathbf{1}}^{\mathbf{m(i)}} p_h^{Ei}\left(t, \mathbf{y}_{\mathbf{j}}^{\mathbf{i}}\right) \mathcal{L}_{\mathbf{j}}^{\mathbf{i}}(\mathbf{y}) \right) \right).$$

The computation of the maximum of (7.6) over time is performed in the same way. Another possibility is to apply the sparse grid operator $\mathcal{G}_{\Lambda_\Phi}$ directly to the minimum or maximum pressure over time. For the minimum pressure, the SC approximation is then given by

$$\begin{aligned} \widetilde{\min_{t \in [0,T]}} p^{Ei}(t, \mathbf{y}) &:= \mathcal{G}_{\Lambda_\Phi}\left[\min_{t \in [0,T]} p_h^{Ei}(t, \cdot)\right](\mathbf{y}) = \sum_{\mathbf{i} \in \Lambda_\Phi} \Delta_{\mathbf{i}}\left[\min_{t \in [0,T]} p_h^{Ei}(t, \cdot)\right](\mathbf{y}) \\ &= \sum_{\mathbf{i} \in \Lambda_\Phi} c_{\mathbf{i}} \left(\sum_{\mathbf{j}=\mathbf{1}}^{\mathbf{m(i)}} \min_{t \in [0,T]} p_h^{Ei}\left(t, \mathbf{y}_{\mathbf{j}}^{\mathbf{i}}\right) \mathcal{L}_{\mathbf{j}}^{\mathbf{i}}(\mathbf{y}) \right). \end{aligned} \tag{7.7}$$

The SC approximation of the maximum pressure is defined analogously. Note that the minimum or maximum pressure over time can have kinks or discontinuities in the stochastic space, even if the time-dependent pressure has a certain stochastic smoothness. This situation can occur if the minimum of the pressure is taken at different time points for different points $\mathbf{y}_i \in \Gamma$. If we directly interpolate the minimum and maximum pressure over the time as in (7.7), possible kinks and discontinuities in the stochastic space could not be reconstructed since global polynomials are used for the sparse grid interpolation. Consequently, we could obtain an inaccurate interpolant and compute falsified probabilities. For exit node $E2$, we observe that both approaches provide different approximations of $\min_t p^{E2}(t, \mathbf{y})$. The left plot in Figure 7.7 shows the minimum of the SC approximation (7.6) of the pressure which is evaluated at $y_3 = 1$ and has a kink in the stochastic space. In contrast, the SC approximation (7.7) of the minimum of the pressure is constant since we obtain only the same minimum value for each collocation point, see right plot

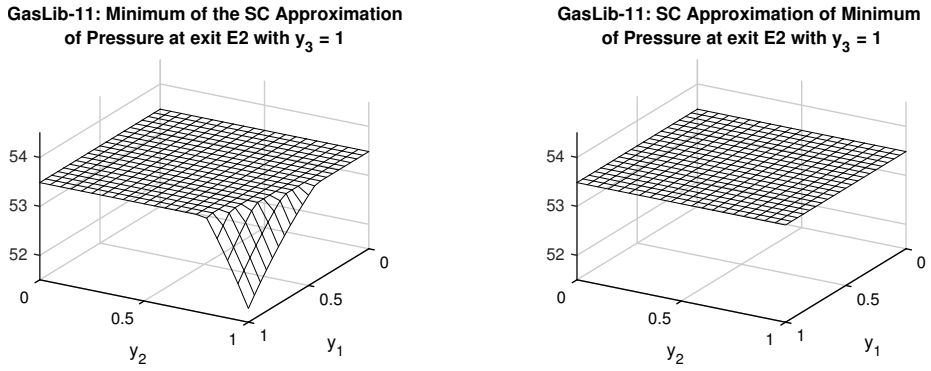


Figure 7.7: *GasLib-11*: Left: Minimum of the SC approximation of the pressure at exit $E2$ over the time plotted versus y_1 and y_2 for fixed $y_3 = 1$, i.e. $\min_{t\in[0,T]} \widetilde{p}^{E2}(t, y_1, y_2, 1)$. Right: For the same exit and $y_3 = 1$, SC approximation of the minimum of the pressure at exit $E2$ over the time, i.e. $\widetilde{\min}_{t\in[0,T]} p^{E2}(t, y_1, y_2, 1)$. In both cases, the sparse grid $\mathbb{H}_{\Lambda_\Phi}$ which we obtain for the expectation of the compressor cost is reused.

in Figure 7.7. In this case, the reused sparse grid $\mathbb{H}_{\Lambda_\Phi}$ which we originally obtain for the expected valued of the compressor cost does not detect the kink.

Before we blindly apply the KDE approach, we want to check whether the minimum and maximum pressure over time at each exit are even RVs which have an absolutely continuous distribution and thus a PDF. We independently draw $S = 50^3$ samples $\mathbf{y}^{(j)}$ of the random vector $\xi = (\xi_1, \xi_2, \xi_3)$ according to its distribution. The components ξ_i are defined by $\xi_i \sim \mathcal{U}[-1, 1]$ and are used to model the uncertain gas demands (7.1). At each sample $\mathbf{y}^{(j)}$, we evaluate the SC approximations $\widetilde{p}^{Ei}(t, \mathbf{y}^{(j)})$ and compute its minimum and maximum over the time period $[0, T]$. For GasLib-11, we prescribe the pressure bounds $p^{\min} = 43$ bar and $p^{\max} = 64$ bar.

In Figure 7.8, we illustrate 30 of these evaluations. Due to the pressure bounds, we are already expecting that the quantities $\overline{p}^{E1}, \underline{p}^{E2}, \underline{p}^{E3}$ are feasible. Since the RV ξ is bounded, the uncertain gas demand and the pressure at the exit nodes are bounded as well. If the number of samples are sufficiently large, we can determine the bounds by computing the minimum and maximum of the samples $\widetilde{p}^{Ei}(t, \mathbf{y}^{(i)})$ for each time point $t \in [0, T]$ and $i = 1, 2, 3$ which are also shown in Figure 7.8. The minimum and maximum of these values over the time points are again the bounds for the RV $\overline{p}^{Ei}$ and $\underline{p}^{Ei}$. Considering Figure 7.8, we observe that the minimum of the plotted samples of $\widetilde{p}^{E2}$ is always equal to the constant initial pressure $\widetilde{p}_0^{E2}$. However, the lower bound of $\underline{p}^{E2}$ is smaller than the initial pressure. Therefore, we have a mixed RV: It has a discrete as well as a continuous part and its cumulative distribution function has a jump discontinuity at the value of the initial pressure. This mixed RV has a kink in the stochastic space shown in left plot of Figure 7.7 and no PDF exists. For $\underline{p}^{E3}$, we observe the same property. One approach to compute the single probability is to start with the discrete part and to determine the jump height by counting the samples which are equal to the initial pressure. For exit $E2$ and $E3$, the initial pressure is taken with probability

$$\mathbb{P}\left(\underline{p}^{Ei}(\cdot) = \widetilde{p}_0^{Ei}\right) \approx 0.9989$$

with $i = 1, 2$. Therefore, it is not necessary to consider the continuous part in detail and we can approximate the mixed RV by a discrete one with $\mathbb{P}\left(\underline{p}^{Ei}(\mathbf{y}) \in [p^{\min}, p^{\max}]\right) \approx 1$.

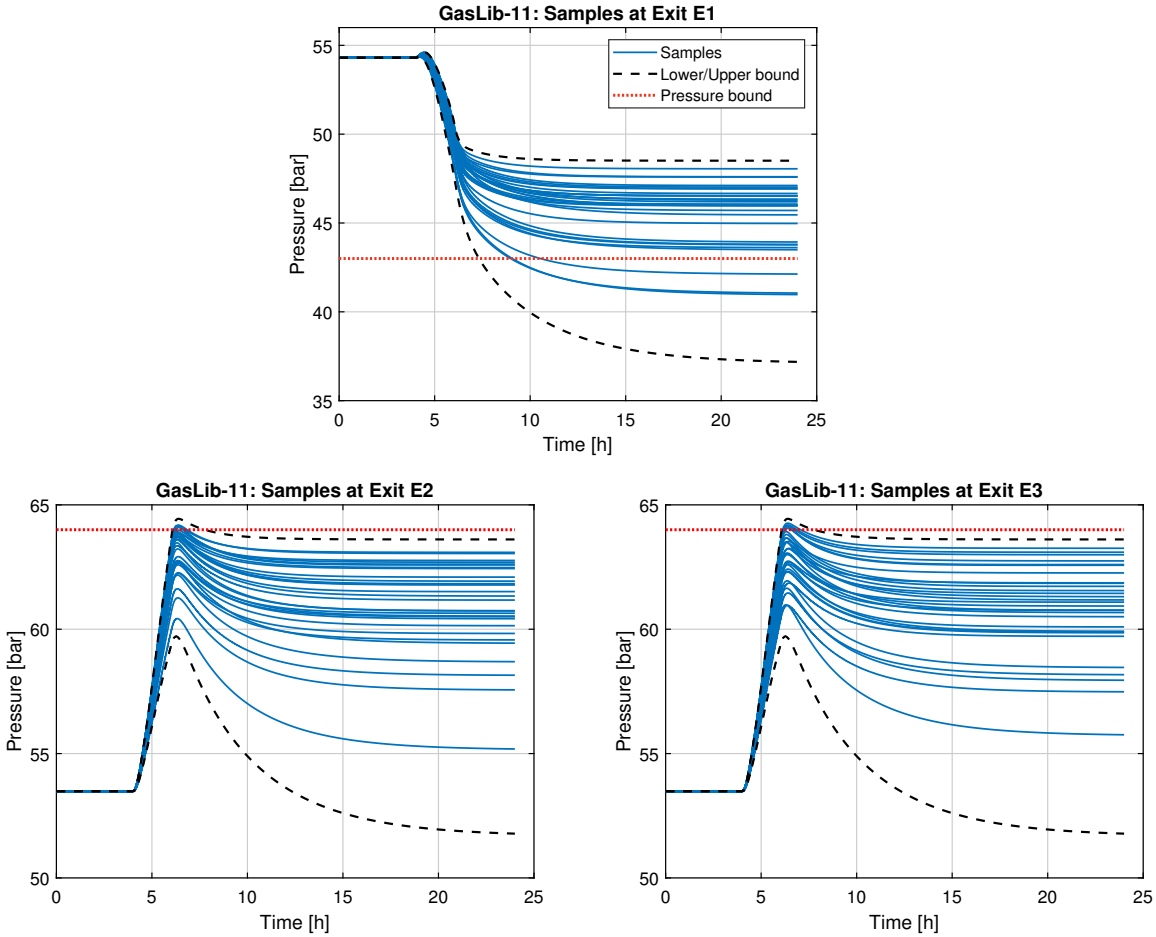


Figure 7.8: *GasLib-11*: Samples of the pressure at exits $E1, E2$ and $E3$ with their lower and upper bound (dashed black line). The pressure bounds $p^{\min} = 43$ bar and $p^{\max} = 64$ bar are plotted as red dotted lines and are violated by some samples.

Furthermore, we note that $\underline{p}^{E1}, \overline{p}^{E1}, \overline{p}^{E2}$ and $\overline{p}^{E3}$ take only values in an interval that does not contain the initial pressure at the corresponding exit. Therefore, we can assume that the quantities have an absolutely continuous distribution and thus have an unknown PDF. We approximate each PDF of the RVs $\underline{p}^{E1}, \overline{p}^{E1}, \overline{p}^{E2}$ and $\overline{p}^{E3}$ by the one-dimensional smooth KDE (6.5) with Gaussian kernel and bandwidth (6.6). The previously computed samples generate the sampling set $\mathcal{X}$ for the KDE, e.g. $\mathcal{X} = \{\underline{p}^{E1}(\mathbf{y}^{(1)}), \ldots, \underline{p}^{E1}(\mathbf{y}^{(S)})\}$. As aforementioned, the RVs $\underline{p}^{E1}, \overline{p}^{E1}, \overline{p}^{E2}$ and $\overline{p}^{E3}$ are bounded. In order to maintain this property in the approximated PDF, we use the two-side boundary correction (6.17) of the previous KDEs which is presented in Section 6.5 and provides a discontinuous approximated PDF.

In Figure 7.9, we compare the classic KDE and its boundary correction, i.e. the reflection estimator. We observe that the approximated PDFs of the maximum pressure at the exits $E2$ and $E3$ coincide due to the structure of the gas network and to the uncertain gas demands (7.1). If many samples are close to the bounds as for the maximum pressure at the exit $E1$, the boundary correction produces an approximation which differs significantly from the classic KDE around the lower and upper bound. Otherwise, the classic KDE and its boundary correction are very similar which we observe for the other RVs. The difference between the two KDEs has only an influence on the probability if at least one pressure bound is close to the initial pressure. In order to realize an automatic validation check for arbitrary pressure bounds, we prefer to use the boundary correction method directly even if its application would not be necessary.

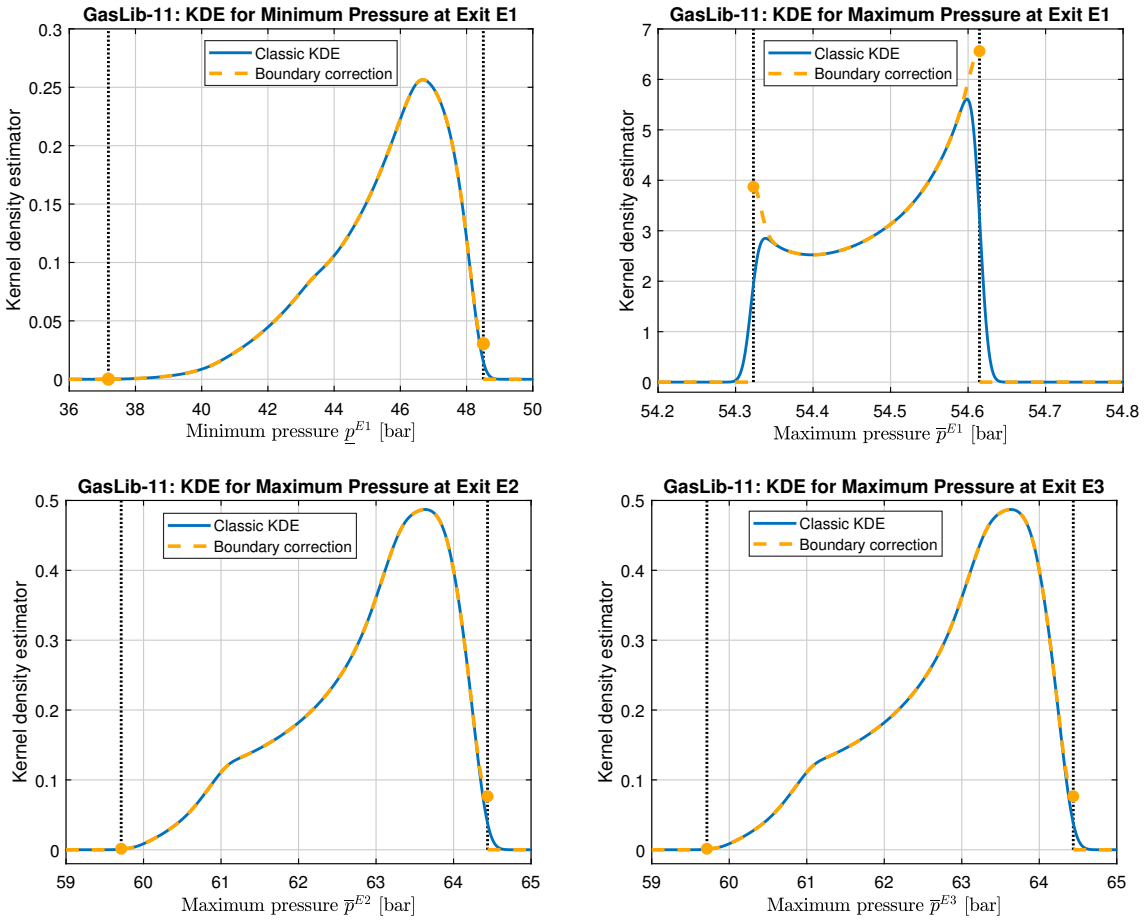


Figure 7.9: *GasLib-11*: Smooth Kernel density estimators (blue solid line) and its discontinuous two-side boundary correction (orange dashed line) for the minimum pressure at $E1$ and the maximum pressure at $E1, E2$ and $E3$. The lower and upper bound of the random variables are plotted as black dotted lines.

Using the approximated PDF of $\underline{p}^{E1}, \overline{p}^{E1}, \overline{p}^{E2}, \overline{p}^{E3}$, we compute the single probabilities by integrating the KDE and its boundary correction over the pressure bounds, see Section 6.4 and Section 6.5, respectively. As pointed out in Subsection 6.4.1, we use the MATLAB® function `erf` due to lower computation times. For both approximated PDFs, we obtain the same probabilities:

$$\mathbb{P}\big(\underline{p}^{E1} \in [p^{\min}, p^{\max}]\big) \approx \mathbb{P}_S\big(\underline{p}^{E1} \in [p^{\min}, p^{\max}]\big) = 0.89,$$

$$\mathbb{P}\big(\overline{p}^{Ei} \in [p^{\min}, p^{\max}]\big) \approx \mathbb{P}_S\big(\overline{p}^{Ei} \in [p^{\min}, p^{\max}]\big) = \begin{cases} 1 & \text{for } i = 1, \\ 0.90 & \text{for } i = 2, \\ 0.90 & \text{for } i = 3. \end{cases}$$

The desired joint probability (7.5) can be rewritten to

$$\mathbb{P}\left(\underline{p}^{E1} \in [p^{\min}, p^{\max}] \text{ and } \overline{p}^{Ei} \in [p^{\min}, p^{\max}] \quad \text{for } i = 2, 3\right)$$

since we can omit the quantities which are always feasible, i.e. quantities with probability of 1. The random vector X is reduced to

$$X_r := (\underline{p}^{E1}, \overline{p}^{E2}, \overline{p}^{E3}).$$

In order to approximate the previous probability, we need to compute the PDF of the RV X_r by applying the three-dimensional KDE (6.10) with Gaussian kernel and diagonal bandwidth matrix (6.8). Of course, the previous samples of $\underline{p}^{E1}, \overline{p}^{E2}$ and $\overline{p}^{E3}$ can be used as samples for X_r. Finally, we obtain the required probability

$$\mathbb{P}\left(X_r \in [p^{\min}, p^{\max}]^3\right) \approx \mathbb{P}_S\left(X_r \in [p^{\min}, p^{\max}]^3\right) = 0.730. \tag{7.8}$$

The KDE with boundary correction provides the same probability, since the prescribed pressure bounds are again far enough from the bounds of the RV X_r.

Another equivalent formulation of the probability (7.5) is

$$\mathbb{P}\left(\min_{\substack{t\in[0,T],\\ i=1,2,3}} \widetilde{p}^{Ei}(t,\cdot) \in [p^{\min}, p^{\max}] \text{ and } \max_{\substack{t\in[0,T],\\ i=1,2,3}} \widetilde{p}^{Ei}(t,\cdot) \in [p^{\min}, p^{\max}]\right). \tag{7.9}$$

Here, we always have a two-dimensional RV. However, taking minimum or maximum over all exits can result in a RV which has no PDF even if $\min_t \widetilde{p}^{Ei}$ and $\max_t \widetilde{p}^{Ei}$ have a PDF. Therefore, this reduction step has to be performed carefully and not in a fully automatic way. In our case, the images of $\min_t \widetilde{p}^{Ei}$ and $\max_t \widetilde{p}^{Ei}$ are either equal or disjoint intervals for all three exits, see Figure 7.8. Consequently, we obtain

$$\min_{\substack{t\in[0,T],\\ i=1,2,3}} \widetilde{p}^{Ei}(t,\cdot) = \min_{t\in[0,T]} \widetilde{p}^{E1}(t,\cdot) \quad \text{and} \quad \max_{\substack{t\in[0,T],\\ i=1,2,3}} \widetilde{p}^{Ei}(t,\cdot) = \max_{t\in[0,T]} \widetilde{p}^{E2}(t,\cdot) = \max_{t\in[0,T]} \widetilde{p}^{E3}(t,\cdot)$$

which defines a two-dimensional RV with a PDF. We approximate the PDF by the two-dimensional KDE (6.10) and its boundary correction. Both approximations yield the same probability of 0.753 which differs slightly from the result in (7.8). Since the considered RV is now two- instead of three-dimensional and we reuse all drawn random values $\mathbf{y}^{(i)}$ from the beginning, we can assume that this probability is more accurate. Nevertheless, the difference between the probabilities (7.8) and (7.9) should decrease with increasing the number of samples.

7.2 A Large Gas Network with Compressor Stations (GasLib-40)

The second example is GasLib-40 which is shown in Figure 7.10. It consists of 3 sources, 29 exits, 6 compressor stations and 39 pipes. The pipes have a length between 3.1 km and 86.7 km which is not represented in Figure 7.10. For a well-arranged illustration of the network, we use again edges of length 0 (short edges). The transformation to a network problem without short edges is described in detail for the example GasLib-11 in the previous section. In this case, we would perform the degeneration to single nodes for the node sets $\{C1, E16\}$, $\{C2, E26\}$, $\{C3, J4, E28\}$, $\{C4, J2, S3\}$, $\{C5, S2\}$ and $\{C6, E21\}$.

We divide the 29 exits into 8 different local regions (REs):

RE1 $= E1 - E2$, RE2 $= E3 - E12$, RE3 $= E13 - E15$, RE4 $= E16 - E20$,
RE5 $= E21 - E22$, RE6 $= E23 - E26$, RE7 $= E27$, RE8 $= E28 - E29$.

Each region is characterized by equal gas demand and uncertainties.

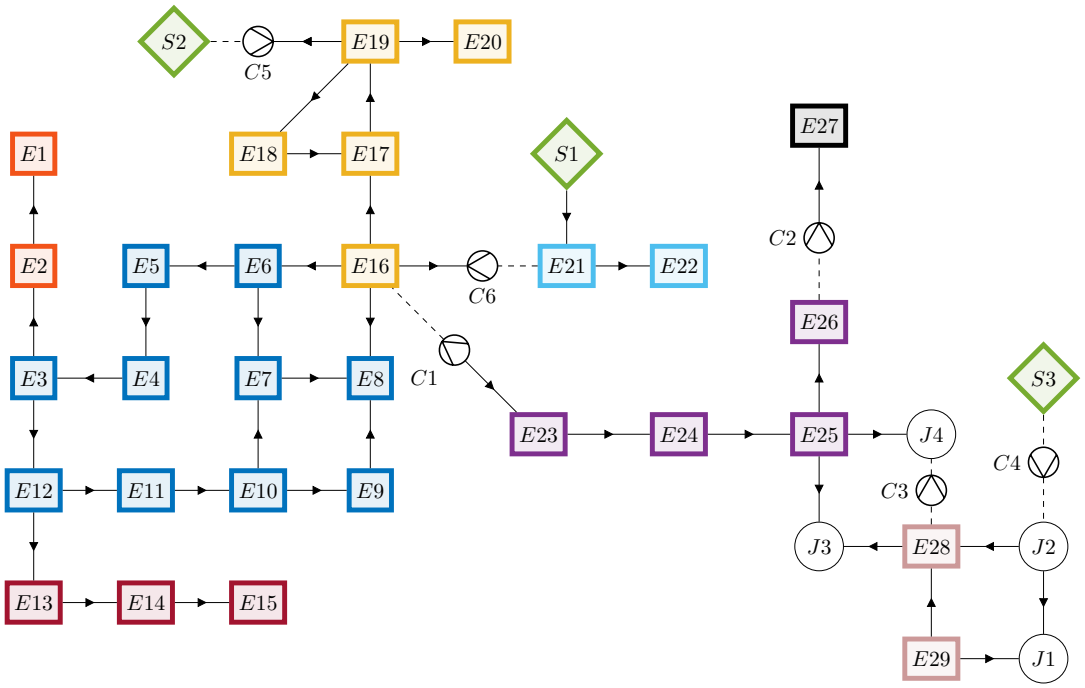


Figure 7.10: Schematic description of the gas network GasLib-40 with 39 pipes (lines with arrows), 3 sources (green diamonds: $S1, S2, S3$), 29 exits (colored squares: $E1 - E29$) and 6 compressor stations ($C1 - C6$). The dashed lines represent additional pipes of edges 0 leading to a well-arranged network illustration. The arrows indicate the orientation of the edges and is used to identify the flow direction: If the gas flows against the direction of the edge, then the flow has a negative sign.

The stationary initial state u_A and the final state u_B are defined by prescribing the data from Table 7.3 at the corresponding sources, exits and compressor stations. The simulation starts with the stationary initial state $u_0 = u_A$ which implies that the quasi-stationary semilinear isothermal Euler equations $\mathcal{M}_3$, see (2.17), are used as initial model on each pipe. The prescribed data for the gas demand at the exits and for the control of the compressor stations are shown in Figure 7.11. For the sources $S1$ and $S2$, the prescribed values are linearly changed after $t = 4$h such that the new values determined by the final state u_B are reached at $t = 6$h. The total simulation time is $T = 12$h with initial time step size $\Delta t = 0.5$h and initial spatial step size $\Delta x = 1,000$m. We divide the time interval $[0, T]$ into 4 time blocks of 3h and perform the adaptive strategy described in Subsection 2.5.4.

Next, we extend the deterministic final state u_B by uncertain gas demand for all eight exit regions $RE1 - RE8$. The controls and the other prescribed data remain deterministic, including the initial value. We assume that the consumers of different exit regions behave independently from each other such that we parametrize the uncertain behavior by the stochastic variable $\mathbf{y} = (y_1, \dots, y_8) \in [-1, 1]^8 =: \Gamma$ which is a realization of the random vector $\xi = (\xi_1, \dots, \xi_8)$. The random vector $\xi : \Theta \to \Gamma$ consists of eight independent and uniformly distributed RVs $\xi_i \sim \mathcal{U}[-1, 1]$. We describe the uncertain flow rate in the final state u_B at all exits of region REi by

$$q^{REi}(u_B, \mathbf{y}) = q_d^{REi}(u_B)\big(1 + 0.3 \cdot \cos(\pi\, y_i)\big), \qquad i = 1, \dots, 8,$$

where $q_d^{REi}(u_B)$ is the corresponding deterministic flow rate for state u_B given in Table 7.3. Additionally, we have $\mathbb{E}[q^{REi}(u_B, \cdot)] = q_d^{REi}(u_B)$ for $i \in \{1, \dots, 8\}$.

Prescribed data	Initial state u_A					Final state u_B				
Source	$S1$	$S2$	$S3$			$S1$	$S2$	$S3$		
Pressure p [bar]	70					60				
Flow rate q [m^3/s]		53.2	53.2				58	53.2		
Exit region	$RE1$	$RE2$	$RE3$	$RE4$	$RE5$	$RE1$	$RE2$	$RE3$	$RE4$	$RE5$
Flow rate q [m^3/s]	5.5	5.5	5.5	5.5	5.5	8.5	8.0	6.0	6.0	6.5
Exit region	$RE6$	$RE7$	$RE8$			$RE6$	$RE7$	$RE8$		
Flow rate q [m^3/s]	5.5	5.5	5.5			4.0	7.5	7.0		
Compressor station	$C1$	$C2$	$C3$	$C4$	$C5$	$C1$	$C2$	$C3$	$C4$	$C5$
Pressure jump [bar]	0	0	5	0	0	5	15	7	12	5
Compressor station	$C6$					$C6$				
Pressure jump [bar]	0					12				

Table 7.3: *GasLib-40*: Prescribed data for sources $(S1, S2, S3)$, exit regions $(RE1 - RE8)$ and control data for compressor stations $(C1 - C6)$ for the initial state u_A and final state u_B.

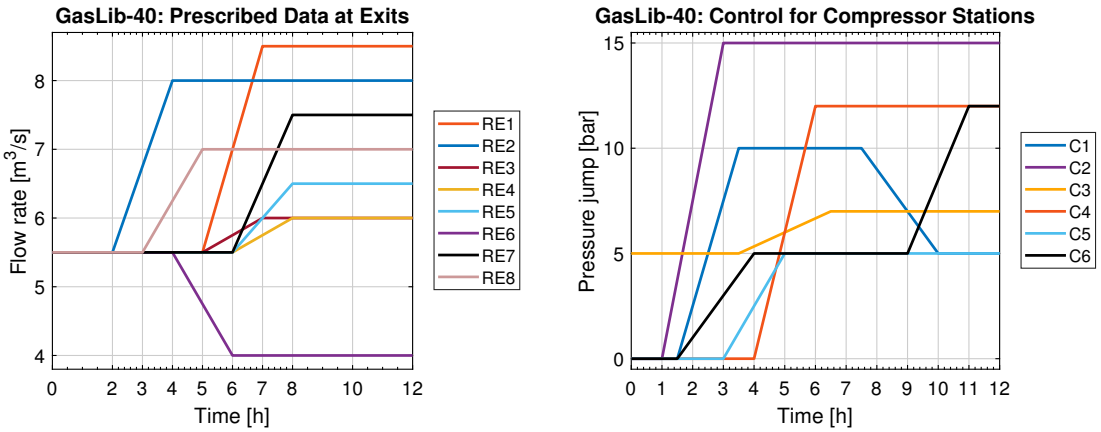


Figure 7.11: *GasLib-40*: Prescribed time-dependent gas demand at the exit regions $RE1 - RE8$ (left) and control of the compressor stations $C1 - C6$ (right) for a transition from state u_A to u_B.

The QoI Φ is again defined by the specific energy consumption of the compressor stations representing the compressor costs. As described for GasLib-11, we have

$$\Phi(\mathbf{y}) = \mathcal{F}[u(\mathbf{y})] = \alpha \sum_{c \in \mathscr{V}_{cs}} \int_0^T g_{c,0} + g_{c,1} P_c\big(u(\mathbf{y})\big) + g_{c,2} P_c\big(u(\mathbf{y})\big)^2 \, \mathrm{d}t$$

with $\mathscr{V}_{cs} = \{C1, \ldots, C6\}$ and $g_{c,0} = 2629, g_{c,1} = 2.47428571429, g_{c,2} = 1.37142857143 \cdot 10^{-5}$ for all compressor stations in GasLib-40. The power $P_c(u(\mathbf{y}))$ required by the compressor station c is defined in (7.2). In order to obtain an expected value of Φ in the order of 0.1, we choose $\alpha = 10^{-10}$.

Now, we apply our adaptive single-level and our adaptive two-level stochastic collocation method with a reduction factor $q = 0.1$ and Clenshaw-Curtis nodes in order to approximate the expected value of the compressor costs Φ. We estimate the parameters $C_h, C_H, C_y, C_Y, \gamma$ and μ as in the first example where we follow the steps in [67].

In order to estimate the constant C_Y and rate μ, we apply the adaptive sparse grid algorithm ASpaGrid for tolerances $TolY = 3 \cdot 10^{-3}, 10^{-3}, 3 \cdot 10^{-4}, 10^{-4}, 3 \cdot 10^{-5}$ and a fixed physical tolerance $TolH = 10^{-4}$. We take the expected value of the QoI Φ for $TolY = 3 \cdot 10^{-5}$ as reference value. In Figure 7.12, the prescribed stochastic tolerances are always satisfied and the error estimates provide always an upper bound for the error. Compared to the first example, the error and the error estimate are now closer such that we get $C_Y = C_y = 0.4$. A least-square fit provides the rate $\mu = 1.1$ in Assumption 5.22 (iv).

In order to estimate the constants C_H and γ, we compute the expected value of Φ for a fixed isotropic Smolyak grid of level $w = 2$ (142 collocation points) and physical tolerances $TolH = 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}$. The approximation of $\mathbb{E}[\Phi]$ for $TolH = 10^{-5}$ is taken as reference value. The numerical errors and the average of the physical error estimates are shown with respect to the computation time in Figure 7.12. We observe that the prescribed physical tolerances are always satisfied and the error estimate provides always an upper bound for the numerical error. For (5.35), we estimate $C_H = C_h = 0.3$. Figure 7.12 suggests that the order of convergence for small tolerances $TolH$ in terms of computation time is slower than for large tolerances. Therefore, we consider the results only for $TolH = 10^{-3}, 10^{-4}$ and set $\gamma = 1.2$ in Assumption 5.22 (v).

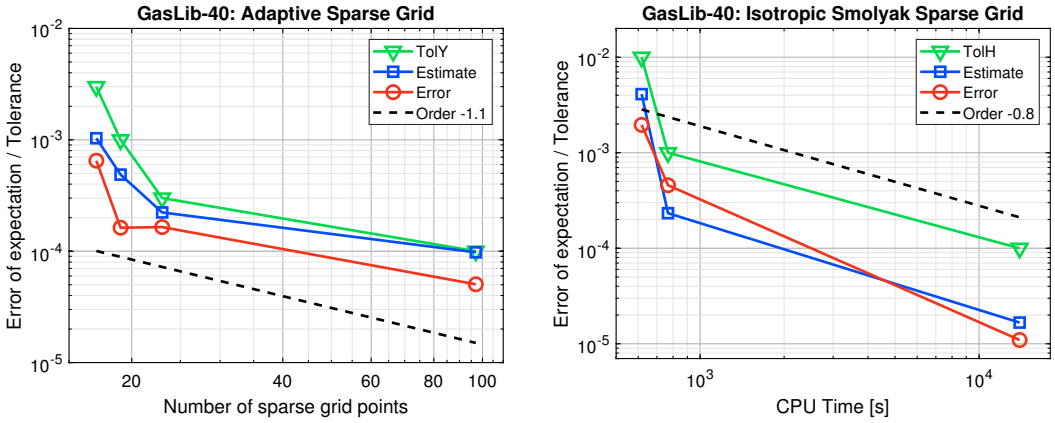


Figure 7.12: *GasLib-40*: Errors and corresponding error estimates: Adaptive sparse grid algorithm ASpaGrid with stochastic tolerances $TolY = 3 \cdot 10^{-3}, 10^{-3}, 3 \cdot 10^{-4}, 10^{-4}$ and fixed physical tolerance $TolH = 10^{-4}$ (left). The estimated order of convergence for $\mathbb{E}[\Phi]$ in terms of collocation points is -1.1. Isotropic Smolyak sparse grid of level $w = 2$ with physical tolerances $TolH = 10^{-2}, 10^{-3}, 10^{-4}$ (right).

Next, we run the single-level and the two-level stochastic collocation method for total accuracies of $\epsilon = 10^{-3}, 10^{-4}, 10^{-5}$. We compute a reference solution $\mathcal{E}_{Q_{ref}}[\Phi_{h_{ref}}] = 0.120726272394500$ by the single-level SC method with $\epsilon = 10^{-6}$ which requires 273 collocation points. The number of the samples taken by the SC methods are given in Table 7.4. The ML approach needs less samples on the fine level $k = 1$ than on the coarse level $k = 0$. The physical tolerances are given by $TolH_k = \epsilon\, q^{k-1}/(2\, C_H)$ with $C_H = 0.3$ and the stochastic tolerances by (5.28) with $C_Y = 0.4, \mu = 1.1$, and $\gamma = 1.2$ for $k = 0, 1$. The errors for the expected value of Φ versus the computational cost are shown in the top-left plot of Figure 7.13. The two-level SC method performs very reliably: It provides always approximations which have a higher accuracy than the prescribed one. In contrast, the achieved accuracy of the single-level method is partially higher than the required one, but only up to a factor of 3. For the lowest accuracy $\epsilon = 10^{-3}$, the single-level approach needs only half of the computation time, but the error of the two-level

estimation is smaller. We observe that the errors of the two-level method are always smaller than the errors of the single-level method for the same accuracy requirements, but only up to a factor of 10. As in the first example, the single-level SC method works sufficiently fast for low accuracies. For the highest accuracy, the two-level method needs significantly less computation time - around a factor of 1/3 faster.

ϵ	SLSC	Two-Level SC	
	Q	Q_0	Q_1
10^{-3}	17	23	17
10^{-4}	53	127	23
10^{-5}	143	151	43

ϵ	SLMC	Two-Level MC	
	M	M_0	M_1
10^{-3}	10	11	10
$3 \cdot 10^{-4}$	39	99	10
10^{-4}	314	428	11
$3 \cdot 10^{-5}$	3,066	4,079	31

Table 7.4: *GasLib-40*: Number of collocation points used for SLSC and two-level SC methods (left) and averaged number of samples used for SLMC and two-level MC methods (right).

The orders of the computational cost in terms of the accuracy ϵ are predicted in Section 5.4: $r_{\text{MLSC}} = -\gamma \approx -1.2$ and $r_{\text{SLSC}} = -\gamma - 1/\mu \approx -2.1$. The top-right plot of Figure 7.13 shows that the theoretical rate for the ML method approximate the observed asymptotic rate quite well. The observed rate for the SL method is significantly better than the theoretical rate. Considering the error in terms of the computational cost in the top-left plot of Figure 7.13, the orders of convergence are predicted by $p_{\text{MLSC}} = 1/r_{\text{MLSC}} \approx -0.83$ and $p_{\text{SLSC}} = 1/r_{\text{SLSC}} \approx -0.47$. However, a least-squares fit results in a faster convergence order $\widetilde{p}_{\text{MLSC}} = \widetilde{p}_{\text{SLSC}} = -1.2$ for both approaches. One possible reason could be cancellations which are overlooked in the error splitting.

Next, we perform the adaptive single-level and the adaptive two-level ($K = 1$) Monte Carlo method as described for the first example. We reuse the physical constants $C_H, C_h = 0.3$ and rate $\gamma = 1.2$ which we estimated for the SC methods. We choose $M_{init} = 10$ as the number of initial samples which are used to estimate the variance in (4.26) and in (4.31) and thus the number of samples at each level. Instead of the mean squared error, we consider the absolute error of the expected value of the MC estimators which fulfills the accuracy requirements as well, see (4.27). We approximate the mean of the SLMC and MLMC estimator by averaging over 10 independent realizations like in (7.3). As reference value for $\mathbb{E}[\Phi]$, we use again the SLSC approximation $\mathcal{E}_{Q_{ref}}[\Phi_{h_{ref}}]$ computed for $\epsilon = 10^{-6}$. The averaged number of the samples required by the MC methods are listed in Table 7.4. The MLMC method uses almost always less samples on the fine level $k = 1$ than on the coarse level $k = 0$.

The bottom-left plot of Figure 7.13 shows the SLMC and the MLMC error for accuracy $\epsilon = 10^{-3}, 3 \cdot 10^{-4}, 10^{-4}, 3 \cdot 10^{-5}$. For both methods, the achieved accuracy is always better than the prescribed accuracy. The advantage of the two-level structure arises only after $\epsilon = 10^{-4}$. Comparing the bottom-left plot with the top-left plot of Figure 7.13, we observe that the SC methods outperform the SLMC method after $\epsilon = 10^{-4}$ and the MLMC method only after $\epsilon = 3 \cdot 10^{-4}$. For a prescribed accuracy which is lower than $\epsilon = 10^{-4}$, the MC and SC methods perform similar. We suppose that the stochastic variations of the QoI could be smoothed out by considering the expected value of the QoI such that few samples are sufficient to get a good approximation.

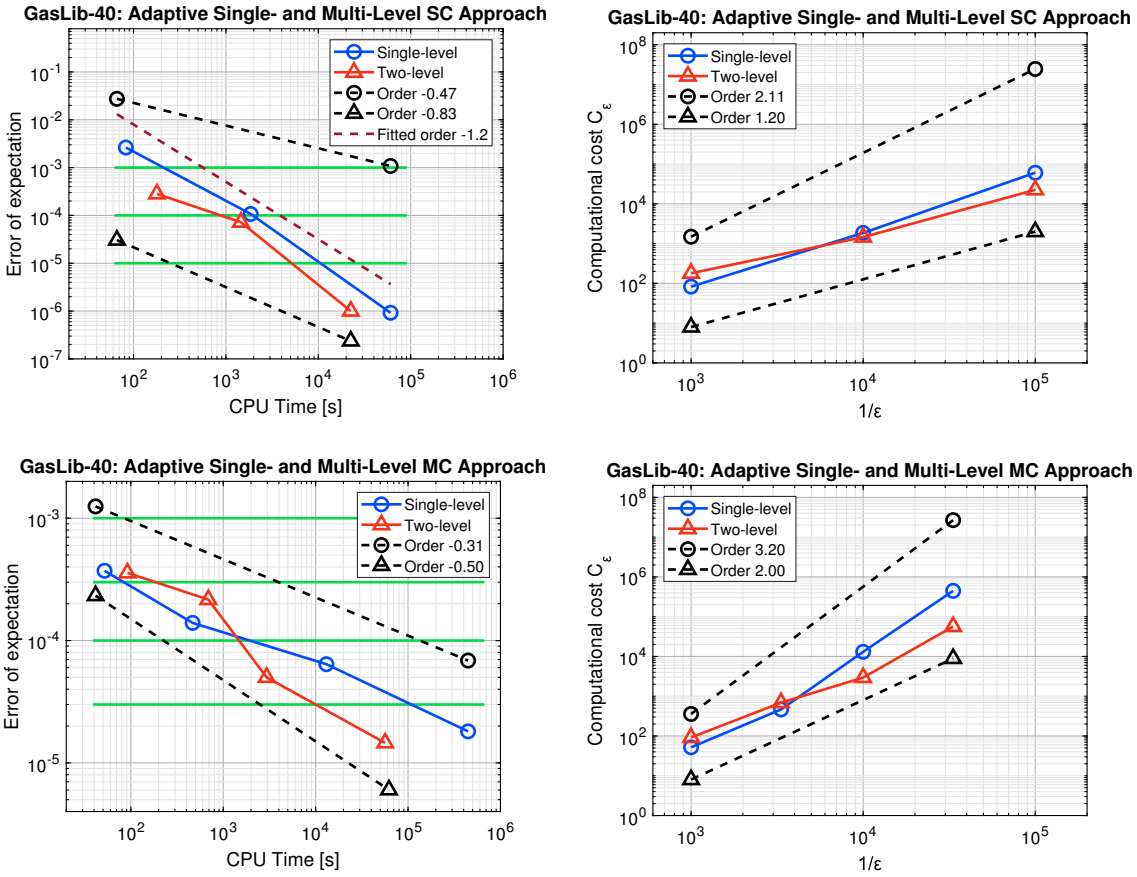


Figure 7.13: *GasLib-40*: Top Left: Errors of the approximated expectations $\mathcal{E}_Q[\Phi_h]$ and $\mathcal{E}_K^{(\mathrm{ML})}[\Phi_{h_K}]$ obtained by the adaptive single-level (blue circles) and two-level (blue triangles, $K = 1$) SC methods for $\epsilon = 10^{-3}, 10^{-4}, 10^{-5}$ (green lines). The predicted convergence orders are -0.47 and -0.83, respectively (dashed black lines). Top Right: Computational cost of the SC methods versus the inverse of the prescribed accuracy. Bottom Left: Mean absolute errors of the approximated expectations $E_M[\Phi_h]$ and $E^K[\Phi_{h_K}]$ obtained by the adaptive single-level (blue circles) and two-level (blue triangles, $K = 1$) MC methods for $\epsilon = 10^{-3}, 3 \cdot 10^{-4}, 10^{-4}, 3 \cdot 10^{-5}$ (green lines). The predicted convergence orders are -0.31 and -0.5, respectively (dashed black lines). Bottom Right: Computational cost of the MC methods versus the inverse of the prescribed accuracy.

The theoretical rate of the computational cost for the single-level MC approach is given by $r_{\mathrm{SLMC}} = -2 - \gamma = -3.2$, see Lemma 4.23. In order to get the theoretical rate for the MLMC method in Theorem 4.29, we estimate a posteriori the rate β in Assumption 4.27. We average the variance over the 10 simulations and obtain $\beta = 1.4$ by a least-squares fit. Consequently, we have $r_{\mathrm{MLMC}} = -2$. In the bottom-right plot of Figure 7.13, the theoretical rates of the computational cost in terms of the accuracy are quite well visible.

Chapter 8

Conclusion and Outlook

In order to conclude this work, we summarize the content and emphasize the main results and their underlying ideas. Finally, we give an outlook on two interesting topics for future research.

8.1 Conclusion

In this work, we have considered hyperbolic PDEs with random data on networks. We have developed and extensively analyzed a fully error-controlled quantification of the transport of uncertainties through the network. For Monte Carlo (MC) and stochastic collocation (SC) methods, we have successfully combined adaptive strategies in the stochastic and physical space with a multi-level structure such that a prescribed accuracy of the simulation was achieved while the computational effort was reduced. Moreover, we have proposed a cost-efficient approach to validate the feasibility of relevant output quantities.

We focused on MC and SC methods since they allow to reuse existing deterministic solvers. We assumed that the physical approximations are computed with a sample-dependent resolution in space, time and model hierarchy which can be controlled by a user-given physical tolerance due to a posteriori error estimates. First, we developed an adaptive single-level (SL) approach of both methods where we efficiently combine adaptive strategies in the stochastic space with adaptive physical approximations. The extension to a multi-level (ML) structure was realized by coupling physical approximations with different accuracies such that the computational cost was minimized. As quantity of interest (QoI), we considered not only a functional of the uncertain solution but also the solution itself.

Considering MC methods, we determined the number of samples and the physical tolerance needed to achieve the prescribed total accuracy in the SL strategy. For the multi-level structure, we determined the physical tolerances and the optimal number of samples at each level such that the total computational cost is minimized and a prescribed accuracy is again achieved. The complexity analysis provided an upper bound for the computational cost and showed the improved asymptotic rates of the ML strategy.

For the SC methods, we realized the adaptive stochastic strategy by adaptive sparse grids with a posteriori error estimates such that we were able to control the discretization of the stochastic approximations by user-given stochastic tolerances. Compared to the MC approach, we followed the same concept for the convergence analysis and determined the optimal stochastic tolerances instead of the number of samples at each level. From the complexity analysis, we obtained again an upper bound for the computational cost and determined the order of the computational cost in terms of the prescribed accuracy under certain assumptions on the convergence of the adaptive strategies.

Moreover, we developed an efficient approach for the validation of real-valued and function-valued output quantities: In order to compute the probability that the output quantity meets some prescribed bounds, we approximated its usually unknown probability density function (PDF) by a kernel density estimator (KDE) and integrated this estimator over the prescribed bounds. We proved the convergence of the KDE as well as of the computed probability. In addition, we computed the probability using different MATLAB® functions and gained the fastest computation time for the Gauss error function `erf`. In order to generate the samples for the KDE with low cost, we recommend to perform the algorithm in a post-processing step of SC methods since SC methods provide an approximation of the output quantity, in contrast to MC methods. In this case, we only need to evaluate the SC approximation instead of numerically solving further deterministic problem.

We applied our developed adaptive single-level and multi-level methods to two examples of gas networks. For both examples, we considered uncertain gas demands at the exits. The error-controlled SC methods as well as the MC methods performed very reliable, especially for high accuracies. As expected, the SC methods always outperformed the MC methods for high accuracy requirements. Moreover, the extension to the multi-level structure significantly improved the asymptotic rates of the complexity compared to the corresponding SL methods. On the one hand, we observed that it is sufficient to choose the SLSC approach for low accuracy levels since the computation time was only a few minutes. On the other hand, the MLSC method applied for high accuracy levels led to a significant reduction of the computation time and thus revealed its potential. However, the theoretical convergence orders of the error in terms of the computation time did not always coincide with the observed ones. We expect to achieve the theoretical convergence orders if we had reliable a posteriori error estimators instead of heuristic error indicators. Furthermore, we demonstrated the applicability of the KDE approach as a post-processing step of the SC method and checked the feasibility of the pressure at the exits of the gas network.

8.2 Outlook

For the SC methods, we have focused on global Lagrange polynomials for the interpolation such that we had to consider QoIs which are sufficiently smooth with respect to the stochastic variables. In the case of discontinuities or kinks in the stochastic space, the convergence of the method can be very slow or even fail. The accuracy of the sparse grid approximation could decline. For example, in gas networks, a sample-dependent regulation of the gas flow or pressure realized by control valves and compressor stations could lead to output quantities having kinks or discontinuities in the stochastic space. In Subsection 7.1.1, we have observed for a deterministic control of the compressor stations that taking the minimum of the pressure at exits over time can already cause kinks in the stochastic space (see Figure 7.7). Consequently, a future work is to extend our developed adaptive SC methods to discontinuous QoIs.

In [74], the authors developed an adaptive sparse grid method which uses piecewise linear functions and adaptively adds more collocation points in stochastic discontinuity regions which can be detected by the hierarchical surpluses. Recently, this method was extended to cubic splines applied in smoother regions in order to achieve a faster convergence [12] and to high-order piecewise polynomials [105]. Besides locally adaptive sparse grids, there are also other approaches. In [36], piecewise polynomial functions defined on simplices are used to approximate QoIs with kinks in the stochastic space. The resulting simplex SC method was applied to stationary gas networks with uncertain demands. Furthermore, Jakeman et al. [56] proposed a multi-element

SC method including a polynomial annihilation technique which locates the stochastic discontinuities based on sparse grids. For the further development of our error-controlled adaptive SC methods, we would suggest to focus on locally adaptive sparse grids equipped with error indicators and to combine it with an automatic detection of discontinuities or kinks in the stochastic space.

Moreover, we have demonstrated the application of the KDE to compute the probability that a chosen QoI is feasible, i.e. meets the prescribed bounds. The samples needed for the KDE can be generated cost-efficiently by evaluating the sparse grid approximation of the QoI in a post-processing step of an SC method. A future extension could be to investigate probabilistic constrained optimization problems [100] of the form

$$\begin{aligned} \min \quad & \psi(c), \\ \text{s.t.} \quad & \mathbb{P}\big(g_i(u_h, c) \geq 0 \ \forall i \in \{1, \dots, k\}\big) \geq \alpha \in (0, 1) \end{aligned} \tag{8.1}$$

with the control variable c, the constraint function $g : L^2(\Theta, \mathbb{L}^1) \times \mathbb{R}^m \to \mathbb{R}^k$, the objective function ψ and the stochastic solution u_h of a partial differential equation problem with random data given for a fixed control c. The included probability can be computed by integrating the PDF of g which is usually unknown. In this case, we can approximate the PDF by applying the KDE approach which leads to an approximated probabilistic constraint and therefore to an approximated optimization problem. An advanced application for dynamic gas networks with uncertain demands would be to compute (deterministic) optimal controls of compressor stations such that the probability for the pressure at the exit nodes to be between the given bounds for the full time period is at least the requested probability level α. In the context of stationary gas networks and transient flow networks, the KDE has already been used to compute probabilities in probabilistic constrained optimization problems where the upper pressure bound is minimized, see the work of the author in collaboration with Schuster, Gugat and Lang [98]. However, the samples were still generated by a classical MC sampling and not so far by evaluating the sparse grid approximation from an SC method since the exact analytical solution of each deterministic problem was known. The usage of the KDE allowed to approximate and differentiate probabilistic constraints for time-dependent problems and to derive necessary optimality conditions for the approximated problem [98].

If the exact solutions of the deterministic problems are unknown, we could use our error-controlled adaptive SC methods for the KDE sampling and also investigate how the error of KDE can be controlled. Further open questions are how the solution of the approximated optimization problem can be computed for a prescribed accuracy and whether an adjoint approach can be realized to improve the efficiency of the solution algorithm. Furthermore, an alternative to probabilistic constraints is to incorporate the probability as a weighted penalty term into the objective function. For this setting, the above-mentioned open questions remain the same.

List of Acronyms

IBOX	Implicit BOX
i.i.d.	Independent and identically distributed
KDE	Kernel density estimator
MC	Monte Carlo
ML	Multi-level
MLMC	Multi-level Monte Carlo
MLSC	Multi-level stochastic collocation
MSE	Mean squared error
PDE	Partial differential equation
PDF	Probability density function
QoI	Quantity of interest
RMSE	Root mean squared error
RV	Random variable
SC	Stochastic collocation
SG	Stochastic Galerkin
SL	Single-level
SLMC	Single-level Monte Carlo
SLSC	Single-level stochastic collocation
UQ	Uncertainty quantification

Bibliography

[1] R. Abgrall and S. Mishra. *Uncertainty quantification for hyperbolic systems of conservation laws.* In Handbook of Numerical Analysis, pages 507–544. Elsevier, 2017.

[2] R. A. Adams and J. J. F. Fournier. *Sobolev spaces.* Academic Press, Amsterdam Boston, 2003.

[3] L. C. Andrews. *Special functions of mathematics for engineers.* SPIE Press, second edition edition, 1998.

[4] K. Atkinson and W. Han. *Theoretical Numerical Analysis: A Functional Analysis Framework.* Springer New York, 2009.

[5] J.-P. Aubin. *Applied Functional Analysis.* John Wiley & Sons, 2011.

[6] I. Babuška, F. Nobile, and R. Tempone. *A stochastic collocation method for elliptic partial differential equations with random input data.* SIAM Review, 52(2):317–355, 2010.

[7] J. Bäck, F. Nobile, and R. Tempone. *Stochastic spectral Galerkin and collocation methods for PDEs with random coefficients: a numerical comparison.* In J. Hesthaven and E. Ronquist, editors, Spectral and High Order Methods for Partial Differential Equations. Volume 76, Lecture Notes in Computational Science and Engineering, pages 43–62. Springer, 2011.

[8] J. Bäck, R. Tempone, F. Nobile, and L. Tamellini. *On the optimal polynomial approximation of stochastic PDEs by Galerkin and collocation methods.* Mathematical Models and Methods in Applied Sciences, 22(09):1250023, 2012.

[9] M. K. Banda, M. Herty, and A. Klar. *Coupling conditions for gas networks governed by the isothermal Euler equations.* Networks and Heterogeneous Media, 1(2):295–314, 2006.

[10] A. Barth, C. Schwab, and N. Zollinger. *Multi-level Monte Carlo finite element method for elliptic PDEs with stochastic coefficients.* Numerische Mathematik, 119(1):123–161, 2011.

[11] V. Barthelmann, E. Novak, and K. Ritter. *High dimensional polynomial interpolation on sparse grids.* Advances in Computational Mathematics, 12(4):273–288, 2000.

[12] A. Bhaduri and L. Graham-Brady. *An efficient adaptive sparse grid collocation method through derivative estimation.* Probabilistic Engineering Mechanics, 51:11–22, 2018.

[13] C. Bierig and A. Chernov. *Convergence analysis of multilevel Monte Carlo variance estimators and application for random obstacle problems.* Numerische Mathematik, 130(4):579–613, 2014.

[14] J. Brouwer, I. Gasser, and M. Herty. *Gas pipeline models revisited: model hierarchies, nonisothermal models, and simulations of networks.* Multiscale Modeling & Simulation, 9(2):601–623, 2011.

[15] J. Charrier, R. Scheichl, and A. L. Teckentrup. *Finite Element error analysis of elliptic PDEs with random coefficients and its application to multilevel Monte Carlo methods.* SIAM Journal on Numerical Analysis, 51(1):322–352, 2013.

[16] F. Clarke. *Functional Analysis, Calculus of Variations and Optimal Control.* Springer-Verlag GmbH, 2013.

[17] C. W. Clenshaw and A. R. Curtis. *A method for numerical integration on an automatic computer.* Numerische Mathematik, 2(1):197–205, 1960.

[18] K. A. Cliffe, M. B. Giles, R. Scheichl, and A. L. Teckentrup. *Multilevel Monte Carlo methods and applications to elliptic PDEs with random coefficients.* Computing and Visualization in Science, 14(1):3–15, 2011.

[19] R. M. Colombo and M. Garavello. *On the Cauchy problem for the p-system at a junction.* SIAM Journal on Mathematical Analysis, 39(5):1456–1471, 2008.

[20] P. R. Conrad and Y. M. Marzouk. *Adaptive Smolyak pseudospectral approximations.* SIAM Journal on Scientific Computing, 35(6):A2643–A2670, 2013.

[21] C. M. Dafermos. *Hyperbolic Conservation Laws in Continuum Physics.* Springer Berlin Heidelberg, 2010.

[22] G. Detommaso, T. Dodwell, and R. Scheichl. *Continuous level Monte Carlo and sample-adaptive model hierarchies.* SIAM/ASA Journal on Uncertainty Quantification, 7(1):93–116, 2019.

[23] L. Devroye and L. Györfi. *Nonparametric density estimation: the L_1 view.* Wiley, New York, 1985.

[24] P. Domschke. *Adjoint-based control of model and discretization errors for gas transport in networked pipelines.* Verl. Dr. Hut, München, 2011.

[25] P. Domschke, A. Dua, J. J. Stolwijk, J. Lang, and V. Mehrmann. *Adaptive refinement strategies for the simulation of gas flow in networks using a model hierarchy.* ETNA - Electronic Transactions on Numerical Analysis, 48:97–113, 2018.

[26] P. Domschke, B. Hiller, J. Lang, V. Mehrmann, R. Morandin, and C. Tischendorf. *Gas network modeling: an overview.* 2021.

[27] P. Domschke, O. Kolb, and J. Lang. *Adjoint-based control of model and discretisation errors for gas flow in networks.* International Journal of Mathematical Modelling and Numerical Optimisation, 2(2):175–193, 2011.

[28] P. Domschke, O. Kolb, and J. Lang. *Adjoint-based control of model and discretization errors for gas and water supply networks.* In Studies in Computational Intelligence, pages 1–17. Springer Berlin Heidelberg, 2011.

[29] P. Domschke, O. Kolb, and J. Lang. *Adjoint-based error control for the simulation and optimization of gas and water supply networks.* Applied Mathematics and Computation, 259:1003–1018, 2015.

[30] P. Domschke, O. Kolb, and J. Lang. *An adaptive model switching and discretization algorithm for gas flow on networks.* Procedia Computer Science, 1(1):1331–1340, 2010.

[31] P. Domschke, O. Kolb, and J. Lang. *Fast and reliable transient simulation and continuous optimization of large-scale gas networks.* Mathematical Methods of Operations Research, 2022.

[32] M. Eigel, C. Merdon, and J. Neumann. *An adaptive multilevel Monte Carlo method with stochastic bounds for quantities of interest with uncertain data.* SIAM/ASA Journal on Uncertainty Quantification, 4(1):1219–1245, 2016.

[33] D. Elfverson, F. Hellman, and A. Målqvist. *A multilevel Monte Carlo method for computing failure probabilities.* SIAM/ASA Journal on Uncertainty Quantification, 4(1):312–330, 2016.

[34] M. Feischl and A. Scaglioni. *Convergence of adaptive stochastic collocation with finite elements.* arXiv:2008.12591, 2020.

[35] E. Fokken, S. Göttlich, and M. Herty. *Efficient simulation of coupled gas and power networks under uncertain demands.* European Journal of Applied Mathematics:1–27, 2022.

[36] B. Fuchs and J. Garcke. *Simplex stochastic collocation for piecewise smooth functions with kinks.* International Journal for Uncertainty Quantification, 10(1):1–24, 2020.

[37] A. Genz and B. D. Keister. *Fully symmetric interpolatory rules for multiple integrals over infinite regions with Gaussian weight.* Journal of Computational and Applied Mathematics, 71(2):299–309, 1996.

[38] T. Gerstner and M. Griebel. *Dimension-adaptive tensor-product quadrature.* Computing, 71(1):65–87, 2003.

[39] T. Gerstner and M. Griebel. *Numerical integration using sparse grids.* Numerical Algorithms, 18:209–232, 1998.

[40] R. Ghanem, D. Higdon, and H. Owhadi. *Handbook of Uncertainty Quantification.* Springer International Publishing, 2017.

[41] J. Giesselmann, F. Meyer, and C. Rohde. *A posteriori error analysis and adaptive non-intrusive numerical schemes for systems of random conservation laws.* BIT Numerical Mathematics, 60(3):619–649, 2020.

[42] M. B. Giles. *Multilevel Monte Carlo path simulation.* Operations Research, 56(3):607–617, 2008.

[43] E. Godlewski and P.-A. Raviart. *Numerical Approximation of Hyperbolic Systems of Conservation Laws.* Springer New York, 2021.

[44] S. Göttlich, O. Kolb, and K. Lux. *Chance-constrained optimal inflow control in hyperbolic supply systems with uncertain demand.* Optimal Control Applications and Methods, 42(2):566–589, 2020.

[45] C. Gotzes, H. Heitsch, R. Henrion, and R. Schultz. *On the quantification of nomination feasibility in stationary gas networks with random load.* Mathematical Methods of Operations Research, 84(2):427–457, 2016.

[46] A. Gramacki. *Nonparametric Kernel Density Estimation and Its Computational Aspects.* Springer International Publishing, 2018.

[47] S. Grundel, N. Hornung, B. Klaassen, P. Benner, and T. Clees. *Computing surrogates for gas network simulation using model order reduction.* In Surrogate-Based Modeling and Optimization, pages 189–212. Springer New York, 2013.

[48] M. Gugat, M. Herty, A. Klar, G. Leugering, and V. Schleper. *Well-posedness of networked hyperbolic systems of balance laws.* In Constrained Optimization and Optimal Control for Partial Differential Equations. Springer Basel, Basel, 2012, pages 123–146.

[49] D. Guignard and F. Nobile. *A posteriori error estimation for the stochastic collocation finite element method.* SIAM Journal on Numerical Analysis, 56(5):3121–3143, 2018.

[50] M. Hanke-Bourgeois. *Grundlagen der Numerischen Mathematik und des Wissenschaftlichen Rechnens.* Vieweg+Teubner Verlag, 2009.

[51] W. Härdle, A. Werwatz, M. Müller, and S. Sperlich. *Nonparametric and Semiparametric Models.* Springer Berlin Heidelberg, 2004.

[52] N.-B. Heidenreich, A. Schindler, and S. Sperlich. *Bandwidth selection for kernel density estimation: a review of fully automatic selectors.* AStA Advances in Statistical Analysis, 97(4):403–433, 2013.

[53] S. Heinrich. *Multilevel Monte Carlo methods.* In Large-Scale Scientific Computing, pages 58–67. Springer Berlin Heidelberg, 2001.

[54] T. Hytönen, J. van Neerven, M. Veraar, and L. Weis. *Analysis in Banach Spaces.* Springer International Publishing, 2016.

[55] IEA. *Outlook for biogas and biomethane: prospects for organic growth.* Report, 2020.

[56] J. D. Jakeman, A. Narayan, and D. Xiu. *Minimal multi-element stochastic collocation for uncertainty quantification of discontinuous functions.* Journal of Computational Physics, 242:790–808, 2013.

[57] S. Karni and A. Kurganov. *Local error analysis for approximate solutions of hyperbolic conservation laws.* Advances in Computational Mathematics, 22(1):79–99, 2005.

[58] S. Karni, A. Kurganov, and G. Petrova. *A smoothness indicator for adaptive algorithms for hyperbolic systems.* Journal of Computational Physics, 178(2):323–341, 2002.

[59] A. Klenke. *Probability Theory.* Springer International Publishing, 2020.

[60] A. Klimke. *Uncertainty Modeling using Fuzzy Arithmetic and Sparse Grids.* PhD thesis, Universität Stuttgart, 2006.

[61] T. Koch, B. Hiller, M. Pfetsch, and L. Schewe, editors. *Evaluating Gas Network Capacities.* MOS-SIAM, 2016.

[62] O. Kolb. *Simulation and optimization of gas and water supply networks.* Verl. Dr. Hut, München, 2011.

[63] O. Kolb, J. Lang, and P. Bales. *An implicit box scheme for subsonic compressible flow with dissipative source term.* Numerical Algorithms, 53(2-3):293–307, 2009.

[64] R. Kornhuber and E. Youett. *Adaptive multilevel Monte Carlo methods for stochastic variational inequalities.* SIAM Journal on Numerical Analysis, 56(4):1987–2007, 2018.

[65] D. P. Kouri. *An Approach for the Adaptive Solution of Optimization Problems Governed by Partial Differential Equations with Uncertain Coefficients.* PhD thesis, Stanford University, 2012.

[66] D. Kröner. *Numerical Schemes for Conservation Laws.* John Wiley & Sons, 1997.

[67] J. Lang, R. Scheichl, and D. Silvester. *A fully adaptive multilevel stochastic collocation strategy for solving elliptic PDEs with random data.* Journal of Computational Physics, 419:109692, 2020.

[68] J. Lang, P. Domschke, and E. Strauch. *Adaptive single- and multilevel stochastic collocation methods for uncertain gas transport in large-scale networks.* In Mesh Generation and Adaptation, pages 113–135. Springer International Publishing, 2022.

[69] J. Lang and P. Mindt. *Entropy-preserving coupling conditions for one-dimensional Euler systems at junctions.* Networks and Heterogeneous Media, 13(1):177–190, 2018.

[70] F. Leja. *Sur certaines suites liées aux ensembles plans et leur application à la représentation conforme.* Annales Plolonici Mathematics, 4(1):8–13, 1957.

[71] R. J. LeVeque. *Finite Volume Methods for Hyperbolic Problems.* Cambridge University Press, 2013.

[72] M. Liu, Z. Gao, and J. S. Hesthaven. *Adaptive sparse grid algorithms with applications to electromagnetic scattering under uncertainty.* Applied Numerical Mathematics, 61(1):24–37, 2011.

[73] G. J. Lord, C. E. Powell, and T. Shardlow. *An Introduction to Computational Stochastic PDEs.* Cambridge University Press, 2014.

[74] X. Ma and N. Zabaras. *An adaptive hierarchical sparse grid collocation algorithm for the solution of stochastic differential equations.* Journal of Computational Physics, 228(8):3084–3113, 2009.

[75] P. Mindt, J. Lang, and P. Domschke. *Entropy-preserving coupling of hierarchical gas models.* SIAM Journal on Mathematical Analysis, 51(6):4754–4775, 2019.

[76] S. Mishra and C. Schwab. *Sparse tensor multi-level Monte Carlo finite volume methods for hyperbolic conservation laws with random initial data.* Mathematics of Computation, 81(280):1979–2018, 2012.

[77] S. Mishra, C. Schwab, and J. Šukys. *Multi-level Monte Carlo finite volume methods for nonlinear systems of conservation laws in multi-dimensions.* Journal of Computational Physics, 231(8):3365–3388, 2012.

[78] S. Mishra, N. H. Risebro, C. Schwab, and S. Tokareva. *Numerical solution of scalar conservation laws with random flux functions.* SIAM/ASA Journal on Uncertainty Quantification, 4(1):552–591, 2016.

[79] A. Narayan and J. D. Jakeman. *Adaptive Leja sparse grid constructions for stochastic collocation and high-dimensional approximation.* SIAM Journal on Scientific Computing, 36(6):A2952–A2983, 2014.

[80] F. Nobile, R. Tempone, and C. G. Webster. *An anisotropic sparse grid stochastic collocation method for partial differential equations with random input data.* SIAM Journal on Numerical Analysis, 46(5):2411–2442, 2008.

[81] F. Nobile, L. Tamellini, F. Tesei, and R. Tempone. *An adaptive sparse grid algorithm for elliptic PDEs with lognormal diffusion coefficient.* In Lecture Notes in Computational Science and Engineering, pages 191–220. Springer International Publishing, 2016.

[82] E. Novak and K. Ritter. *High dimensional integration of smooth functions over cubes.* Numerische Mathematik, 75(1):79–97, 1996.

[83] E. Novak and K. Ritter. *Simple cubature formulas with high polynomial exactness.* Constructive Approximation, 15(4):499–522, 1999.

[84] M. Ohlberger. *A review of a posteriori error control and adaptivity for approximations of non-linear conservation laws.* International Journal for Numerical Methods in Fluids, 59(3):333–354, 2009.

[85] A. J. Osiadacz. *Different transient models - limitations, advantages and disadvantages.* PSIG report 9606, Pipeline Simulation Interest Group, 1996.

[86] E. Parzen. *On estimation of a probability density function and mode.* The Annals of Mathematical Statistics, 33(3):1065–1076, 1962.

[87] T. N. L. Patterson. *The optimum addition of points to quadrature formulae.* Mathematics of Computation, 22(104):847–856, 1968.

[88] G. A. Reigstad. *Existence and uniqueness of solutions to the generalized Riemann problem for isentropic flow.* SIAM J. Appl. Math, 75(2):679–702, 2015.

[89] N. H. Risebro, C. Schwab, and F. Weber. *Correction to: Multilevel Monte Carlo front-tracking for random scalar conservation laws.* BIT Numerical Mathematics, 58(1):247–255, 2017.

[90] L. A. Roald, K. Sundar, A. Zlotnik, S. Misra, and G. Andersson. *An uncertainty management framework for integrated gas-electric energy systems.* Proceedings of the IEEE, 108(9):1518–1540, 2020.

[91] D. Rose, M. Schmidt, M. C. Steinbach, and B. M. Willert. *Computational optimization of gas compressor stations: MINLP models versus continuous reformulations.* Mathematical Methods of Operations Research, 83(3):409–444, 2016.

[92] M. Rosenblatt. *Remarks on some nonparametric estimates of a density function.* The Annals of Mathematical Statistics, 27(3):832–837, 1956.

[93] T. Roubíček. *Nonlinear Partial Differential Equations with Applications.* Springer Basel, 2012.

[94] D. Schaden. *Variance reduction with multilevel estimators.* PhD thesis, Technische Universität München, 2021.

[95] B. Schieche. *Unsteady Adaptive Stochastic Collocation Methods on Sparse Grids.* PhD thesis, TU Darmstadt, 2012.

[96] M. Schmidt, D. Aßmann, R. Burlacu, J. Humpola, I. Joormann, N. Kanelakis, T. Koch, D. Oucherif, M. Pfetsch, L. Schewe, R. Schwarz, and M. Sirvent. *GasLib-a library of gas network instances.* Data, 2(4):40, 2017.

[97] E. F. Schuster. *Incorporating support constraints into nonparametric estimators of densities.* Communications in Statistics - Theory and Methods, 14(5):1123–1136, 1985.

[98] M. Schuster, E. Strauch, M. Gugat, and J. Lang. *Probabilistic constrained optimization on flow networks.* Optimization and Engineering, 23:1–50, 2022.

[99] D. W. Scott. *Multivariate Density Estimation.* John Wiley & Sons, 2015.

[100] A. Shapiro, D. Dentcheva, and A. Ruszczyński. *Lectures on Stochastic Programming: Modeling and Theory.* Society for Industrial and Applied Mathematics, second edition edition, 2014.

[101] B. W. Silverman. *Density estimation for statistics and data analysis.* Chapman and Hall, London New York, 1986.

[102] S. A. Smolyak. *Quadrature and interpolation formulas for tensor products of certain classes of functions.* Dokl. Akad. Nauk SSSR, 148:1042–1045, 1963.

[103] A. Sommariva. *Fast construction of Fejér and Clenshaw-Curtis rules for general weight functions.* Computers & Mathematics with Applications, 65(4):682–693, 2013.

[104] L. Tamellini and F. Nobile. *Sparse grids matlab kit.* https://sites.google.com/view/sparse-grids-kit, 2009-2017. Version 17-5.

[105] Z. Tao, Y. Jiang, and Y. Cheng. *An adaptive high-order piecewise polynomial based sparse grid collocation method with applications.* Journal of Computational Physics, 433:109770, 2021.

[106] A. L. Teckentrup, P. Jantsch, C. G. Webster, and M. Gunzburger. *A multilevel stochastic collocation method for partial differential equations with random input data.* SIAM/ASA Journal on Uncertainty Quantification, 3(1):1046–1074, 2015.

[107] J. Thiedau. *Dynamic Optimization of Gas Transmission Networks for Storage of Renewable Energy.* PhD thesis, Gottfried Wilhelm Leibniz Universität Hannover, 2018.

[108] L. Trefethen. *Six myths of polynomial interpolation and quadrature.* Mathematics Today, 47, 2011.

[109] L. N. Trefethen. *Is Gauss quadrature better than Clenshaw-Curtis?* SIAM Review, 50(1):67–87, 2008.

[110] B. A. Turlach. *Bandwidth Selection in Kernel Density Estimation: A Review.* Technical report, 1999.

[111] M. P. Wand and M. C. Jones. *Comparison of smoothing parameterizations in bivariate kernel density estimation.* Journal of the American Statistical Association, 88(422):520–528, 1993.

[112] G. Wasilkowski and H. Wozniakowski. *Explicit cost bounds of algorithms for multivariate tensor product problems.* Journal of Complexity, 11(1):1–56, 1995.

[113] D. Werner. *Funktionalanalysis.* Springer-Verlag GmbH, 2011.

[114] D. Xiu and J. S. Hesthaven. *High-order collocation methods for differential equations with random inputs.* SIAM Journal on Scientific Computing, 27(3):1118–1139, 2005.

Curriculum Vitae

Elisa Strauch

2022	**Promotion in Mathematik (Dr. rer. nat.)** Technische Universität Darmstadt
seit 2018	**Mitglied im Sonderforschungsbereich Transregio 154** *Mathematische Modellierung, Simulation und Optimierung am Beispiel von Gasnetzwerken*
seit 2017	**Wissenschaftliche Mitarbeiterin in der AG *Numerik und Wissenschaftliches Rechnen*** Fachbereich Mathematik, Technische Universität Darmstadt
2017	**Master of Science Mathematik** Technische Universität Darmstadt
2014	**Bachelor of Science Mathematik** Johannes Gutenberg-Universität Mainz

Auszeichnung

2022	**Ruth Moufang-Promotionspreis** des Fachbereichs Mathematik der Technischen Universität Darmstadt