



DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Bachelorarbeit in Informatik: Games Engineering

**Development of a Prototype to Quantify
the Uncertainty of the Water Balance Model
LARSIM**

Frank Schraufstetter

DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Bachelorarbeit in Informatik: Games Engineering

**Development of a Prototype to Quantify
the Uncertainty of the Water Balance Model
LARSIM**

**Entwicklung eines Prototyps zur
Quantifizierung der Unsicherheit des
Wasserhaushaltsmodells LARSIM**

Author:	Frank Schraufstetter
Supervisor:	Univ.-Prof. Dr. Hans-Joachim Bungartz
Advisor:	Dr. Tobias Neckel, Florian Künzner, Ivana Jovanovic
Submission Date:	15.03.2019

I confirm that this Bachelorarbeit in Informatik: Games Engineering is my own work and I have documented all sources and material used.

Garching bei München, 15.03.2019

Frank Schraufstetter

Acknowledgments

I would like to thank Professor Bungartz and Dr. Neckel for giving me the opportunity to work in this interesting field of research. Special thanks goes to Florian Künzner and Ivana Jovanovic for providing extraordinary support and guidance. I greatly appreciate their time and effort in providing valuable insight and feedback throughout the project. Finally I would like to thank my family. They encouraged and supported me during the whole project and helped me overcome any obstacles along the way.

Abstract

Hydrological models are prone to uncertainty due to their complexity. The water balance model LARSIM provides a high degree of control through configuration parameters, which have to be calibrated for each application. The uncertainty originating from parameter calibration was evaluated. A prototype was developed to facilitate the propagation of uncertainty through the LARSIM model by using stochastic collocation with pseudo-spectral approach. The snow module parameters were exemplarily used for uncertainty propagation. The output of interest was evaluated using global sensitivity analysis and statistical measurements to determine the impact of uncertain parameters on the simulated runoff. Uncertainty of snow module parameters could not be proven definitively, but the large deviation of simulated and measured runoff indicate uncertainty in the model.

Contents

Acknowledgments	iii
Abstract	iv
1 Introduction	1
2 The water balance simulator LARSIM	2
2.1 The Water balance model in LARSIM	2
2.2 Functionality of the snow module in LARSIM	5
2.2.1 Snow accumulation	6
2.2.2 Energy balance and snowmelt	6
3 Theoretical Background	8
3.1 Uncertainty Quantification	8
3.1.1 Uncertainty and Uncertainty Quantification in a model	8
3.1.2 Stochastic Collocation with pseudo-spectral approach	10
3.2 Global Sensitivity Analysis	12
4 Uncertainty Quantification in LARSIM	14
4.1 Sources of uncertainty in LARSIM	14
4.2 Calibration Analysis	15
4.3 LARSIM configuration setup	16
4.4 Prototype Setup for Uncertainty Quantification in LARSIM	17
5 Prototype for Uncertainty Quantification on LARSIM	20
5.1 Frameworks	20
5.1.1 Uncertainty Quantification Execution Framework	20
5.1.2 Chaospy	20
5.2 Uncertainty Quantification Prototype	21
5.2.1 Overview	21
5.2.2 Preprocessing	22
5.2.3 LARSIM Execution	24

5.2.4	Output processing	26
5.3	Execution on High Performance Computing Systems	27
6	Analysis of the Output of Interest	28
6.1	Global Sensitivity Analysis	28
6.2	Statistical Evaluation	29
6.2.1	Simulation data	29
6.2.2	Discussion	33
7	Conclusion	34
	List of Figures	35
	List of Tables	36
	Bibliography	37

1 Introduction

The prediction and simulation of water runoff in an area is used to provide information about upcoming flood events or low water. This information is forwarded to the appropriate agencies to aid in decision making in critical situations (1). The water balance model LARSIM is used for flood forecasting and the continuous simulation of water runoff in the hydrological cycle, mainly in southern Germany.

The model is very complex and simulates runoff on the basis of meteorological and geological input data and calibrated model parameters. The parameters are calibrated by adjusting them so that the simulated runoff approximates the measured data. Calibration is a very demanding process and it is assumed that the parameters bear some uncertainty.

LARSIM consists of several modules, each controlling one of the water storages. The snow module controls the storage of water in the snow cover and has a high impact on water runoff especially during spring due to snowmelt. It is assumed that parametric uncertainty is easier to detect when choosing a timeframe with extreme weather conditions. Heavy snowfall followed by mild temperatures, which was the case in March and April 2013, can drastically increase runoff by snowmelt and could cause flood events (2).

The focus of this work is to develop a prototype for the quantification of parametric uncertainty in LARSIM. The prototype should handle input processing including the adjustment of LARSIM configuration data according to the selected timeframe and uncertain variables. Stochastic collocation with pseudo-spectral approach is used for uncertainty propagation. The simulation output should then be processed and statistically relevant measurements be generated for evaluation.

The uncertainty within the snow module will be examined by propagating the four uncertain snow parameters through LARSIM. The resulting output will be examined with global sensitivity analysis to identify impactful parameters and statistical evaluation in an attempt to quantify the uncertainty within the module.

An overview of the functionality of LARSIM is provided in chapter 2. Uncertainty quantification and the method used is detailed in chapter 3, followed by a discussion of uncertainty within LARSIM in chapter 4. The developed prototype is presented in chapter 5 and the results of different simulations are discussed in chapter 6.

2 The water balance simulator LARSIM

2.1 The Water balance model in LARSIM

Water balance models are widely used to evaluate critical situations in water management, calculate effects of climate changes and plan land usage. Different hydrological models for runoff prediction are commonly used. Conventional precipitation-runoff-models are mainly used to predict certain aspects of the hydrological cycle, based on events (e.g. floods). Water balance models however are able to predict and simulate important hydrometeorologic data and hydrologic conditions continuously over a long period of time (3), (4).

The water balance model LARSIM, an acronym for **L**arge **A**rea **R**unoff **S**imulation **M**odel, is utilized for the continuous quantification of local and temporal distribution of water in major components of the water balance like precipitation, snow storage, water storage in the catchment and runoff (3). LARSIM is based on the river basin model FGMOD which was designed to be used for the simulation of single flood or low-flood events. A simulation of continuous runoff however was not intended (5).

Singh, 2012, classifies LARSIM as a deterministic, conceptional, distributed model in the mesoscale, which allows its application in catchment sizes ranging from 1km^2 to 100km^2 . It can use either real catchment areas or rasterization (6).

The bavarian area Regen, which is object of this work, uses rasterization with high resolution subareas of 1 km^2 in size.

LARSIM can be used in two different computational configurations: operational mode and simulation mode. Since it was based on FGMOD, it retained its capability for event-based forecasting and can be used in operational mode for flood forecasting which used to be based on precipitation data alone. LARSIM is being used by enviromental authorities in operational mode, mainly for weather forecast and flood prediction, in German speaking areas like Bavaria, Baden-Wuerttemberg and parts of Austria and Switzerland (7). In simulation mode for continuous water balance modeling, LARSIM uses a *.WHM* file which contains the initial state of water storages for the two days prior to simulation start.

It then simulates water runoff based on initial state and measured meteorological data for 63 hours for calibration. After that, it simulates runoff based on provided weather forecast for the remaining time period (8). Simulation mode is used in this thesis to simulate continuous water balance and predict runoff over a two month time period.

The model relies on detailed geological data of the area about the different catchments, for example their linkage through channels and land usage. The impact of geological data on the model output prediction has already been analyzed (9). This data set remains unaltered in this work.

LARSIM requires the following meteorological time series input for the regarded area to function as a water balance model (4):

- Precipitation
- Air temperature
- Relative air humidity or dew point temperature
- Wind speed
- Sunshine duration or global radiation
- Air pressure

These time series are being corrected and adjusted to their respective subareas, which the LARSIM development community details in their official documentation (7). Calculations are based on equidistant time intervals with different variations available. A one-hour interval is suggested for continuous simulation (7), which was used in this work.

Given the correct input data for the region in question, LARSIM is able to simulate the following hydrological processes, among others (10):

- Interception
- Snow accumulation, compaction and melt
- Infiltration, soil water balance, runoff development and percolation
- Evapotranspiration
- Flood-routing in channels and retention in lakes

Figure 2.1 provides an overview of the connection of different hydrological processes within LARSIM. A detailed explanation of all processes can be found in the official documentation (7).

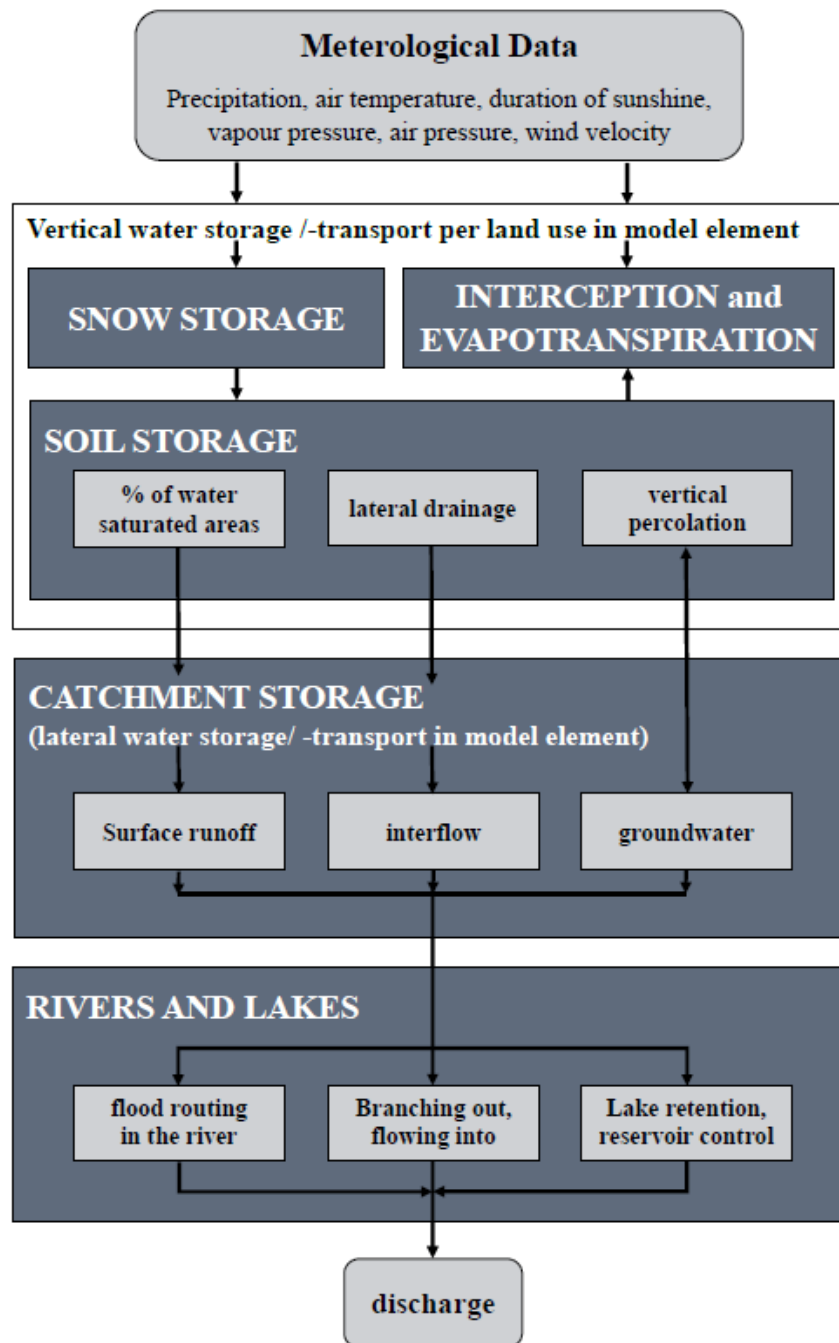


Figure 2.1: Overview of hydrological processes in LARSIM, modified from (11)

Snow processes, interception, evapotranspiration and soil water storage are simulated in subareas separately for each land use category in so-called Grouped Response Units (GRUs). The total runoff of the subarea is the sum of the GRUs runoffs which will be routed to the catchment storage. Released water from the storage is called subarea runoff, which is forwarded to lakes and rivers.

Snow Storage The snow storage is an important part of the LARSIM model because it influences runoff proportions based on the season of the year. It describes the accumulation process of snow and the water flow of snowmelt, being either absorbed by the soil storage or evaporating.

Interception storage Leaves and other parts of vegetation store precipitation up to a certain maximum value. If the storage is full, any more precipitation is passed directly to the ground. Possible evaporation clears out the interception storage.

Soil Storage Soil storage is the center of every hydrological model, and it is highly dependent on the land attributes the model is used on. The module acts as a distributor. Water input (precipitation, snowmelt) either evapotranspires or is channeled into various runoff processes which are stored in their respective catchment storages: surface runoff, interflow and groundwater runoff.

The regular 3-component-model combines surface and fast sub-surface runoff as direct runoff, which makes it difficult to accurately describe flood events. Therefore an extended soil water model with four runoff components is used for this work. The extended model uses an additional compartment which splits direct runoff into fast and slow direct runoff (4).

2.2 Functionality of the snow module in LARSIM

The snow module has a large impact on runoff, especially in spring (4). Heavy precipitation in winter and early spring followed by mild temperatures might lead to flood events (2). These extreme conditions with high variance in data could lead to an easier observation of uncertainty in the model output. It is hypothesized that there is a correlation between uncertainty that arises during the model calibration process and an uncertain runoff prediction in the model output. The snow module is a convenient model to investigate this correlation as a starting point due to its comparatively low complexity.

2.2.1 Snow accumulation

Precipitation is formed at high altitudes and can fall as either rain or snow, depending on the temperature at altitude. Since temperature cannot be measured at these altitudes properly, predictions have to be made based on measurable temperature on ground level. The model assumes precipitation either as snow or rain based on a set threshold temperature at 2m above ground within the area (4).

$$precipitation = \begin{cases} snow, & \text{if } T_L \leq T_{Grenz} \\ rain, & \text{if } T_L > T_{Grenz} \end{cases} \quad (2.1)$$

T_L [°C] measured air temperature 2m above ground

T_{Grenz} [°C] threshold temperature, below which precipitation occurs as snow

T_{Grenz} is calibrated for each gauge station individually and varies depending on the area, ranging from -3°C to 2°C. LARSIM also supports a threshold interval around T_{Grenz} to mitigate the abrupt change between snow and rain. The parameter T_{Spann} implements an interval $[T_{Grenz} - \frac{1}{2}T_{Spann}, T_{Grenz} + \frac{1}{2}T_{Spann}]$, in which a mixture of snow and rain will be linearly interpolated (7).

2.2.2 Energy balance and snowmelt

The temperature of the snow cover is a significant factor in calculating potential snowmelt. It depends on several external factors which can be taken into account during a simulation run.

The energy balance of snow can be simulated in LARSIM by using three different methods, developed by Dr. Knauf: the simplified method, the extended method and the complete energy balance calculation (12). The simplified version only uses precipitation, air temperature and wind speed to calculate the potential snowmelt. The extended version also uses latent heat and shortwave radiation balance for energy balance calculations. These approaches have been enhanced to calculate the complete energy balance W_{tot} by taking latent heat and both short- and longwave radiation balance into account (13). Deciding which method to choose is largely based on available data and computational power. For this thesis the complete energy balance will be calculated based on meteorological data provided by the Landesamt für Umwelt to ensure the most precise results.

The total energy balance W_{tot} of snow can be calculated with (14):

$$W_{tot} = W_G + W_{nied} + W_{sense} + W_{RNS} + W_{latent} + W_{RNL}$$

$\xrightarrow{\text{Knauf simplified}}$
 $\xrightarrow{\text{Knauf extended}}$
 $\xrightarrow{\text{Complete energy balance}}$

with

W_G [W/m²] Ground heat flux
 W_{nied} [W/m²] Heat flux through precipitation
 W_{sense} [W/m²] Flux of sensible heat
 W_{RNS} [W/m²] shortwave radiation balance
 W_{latent} [W/m²] Flux of latent heat
 W_{RNL} [W/m²] longwave radiation balance

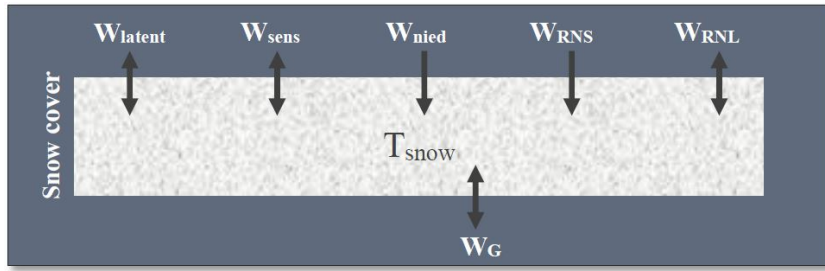


Figure 2.2: Influences of heat sources on snow cover, modified from (14)

The different energy sources all influence the snow cover and its energy balance, as illustrated in figure 2.2. All formulas used in the calculation can be found in the documentation (7), but are not discussed in this work.

The cold content of snow is the required energy to heat snow to a melting temperature of 0 °C. A change in the cold content depends on the energy balance of snow in each timestep. Surplus energy that's available after heating the snow to 0 °C is used to calculate the melting rate (7). Increased water content causes a subsidence of snow and a delayed submission of stored water, which can be calculated using the snow-compaction-model by Bertle (15). Liquid water, either through snowmelt or precipitation, is stored within the snow cover until a predetermined threshold is met. Excess water is released as runoff into the soil storage (3).

3 Theoretical Background

3.1 Uncertainty Quantification

3.1.1 Uncertainty and Uncertainty Quantification in a model

Uncertainty is an inherent part of the real world. As models based on real world events, like weather prediction, become more sophisticated and computational power increases, the main focus is to make model predictions as accurate as possible. Uncertainties can be classified into two categories: aleatoric and epistemic (16).

Aleatoic uncertainty is uncertainty that cannot be reduced by providing more data to the system in question. It is an inherent variability within the system, e.g. a die throw. Epistemic uncertainty stems from a lack of knowledge provided in creating or operating a model. This type of uncertainty can be reduced by adding appropriate data. Approximations and simplifications have to be made when designing a mathematical model based on real-life situations to reduce the computational power required for a simulation run. These model discrepancies can cause *structural uncertainty* in the system. Input variables for these models, like in LARSIM, are numerous and often assumptions and approximations based on past research and provided data, without being able to reflect the reality exactly (16). This *parameter uncertainty* will be examined for the snow module in this thesis, assuming that the model itself is structurally sound.

Uncertainty quantification (UQ) helps to predict the output of a model when subjected to uncertain inputs. The Committee on Mathematical Foundations of Verification, Validation, and Uncertainty Quantification defined Uncertainty Quantification as "*the process of quantifying uncertainties associated with model calculations of true, physical quantities of interest, with the goals of accounting for all sources of uncertainty and quantifying the contributions of specific sources to the overall uncertainty*" (17).

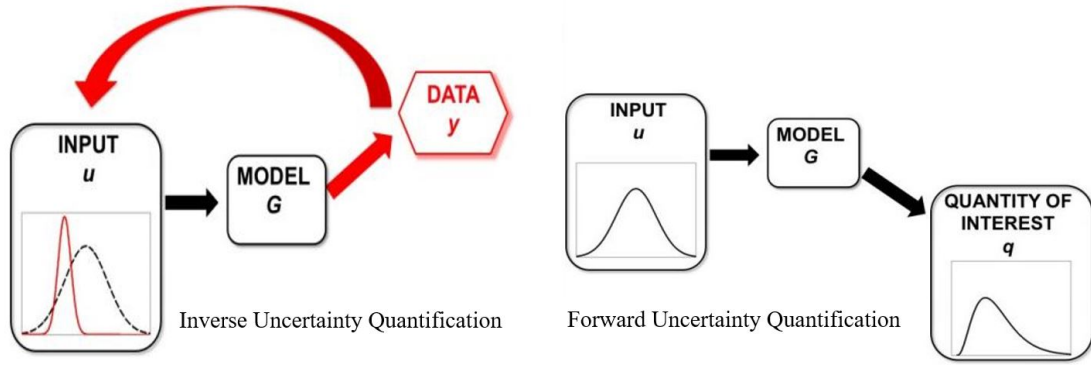


Figure 3.1: Forward and Inverse Uncertainty Quantification, (18)

There are two types of uncertainty quantification: forward and inverse UQ simulations, illustrated in figure 3.1 (18).

Inverse uncertainty quantification estimates the differences between experiments and the underlying mathematical model and predicts the input uncertainty. It is also used to correct model parameters iteratively based on the model output.

Forward uncertainty quantification however quantifies the uncertainty of the output by propagating uncertainty in the input through the model, and this approach will be used in this thesis. Iaccarino et al., 2006, described the three phases in forward propagation: data assimilation, uncertainty propagation and certification. Data assimilation consists of determining the uncertain input variables and defining them with a specific probability distribution function (PDF) (19).

These inputs are being propagated through the model to obtain the PDFs of the output of interest (OoI). The number of simulation runs required for sufficiently accurate OoIs is highly dependent on the uncertainty quantification method chosen. The certification step contains collecting the output of each propagation, analyzing statistical values like mean and standard deviation and measuring the impact of the propagated uncertainty on the output of interest (19).

Stochastic collocation with pseudo-spectral approach is a non-intrusive, computationally inexpensive method for few random variables (20). It fits the requirements for this work and will be detailed in the following.

3.1.2 Stochastic Collocation with pseudo-spectral approach

Several approaches for propagating uncertainty have emerged in the last 20 years (21). The classic Monte Carlo sampling approach, while easy to implement, needs a large number of simulation executions for accurate results. The slow convergence rate of $\frac{1}{\sqrt{N}}$ for N samples makes it computationally ineffective for large-scale models and a high dimension of random variables (22). Wiener et. al., 1938, first proposed using Hermite Polynomial and homogenous chaos for integration theory with respect to the Brownian Motion (23). Hermite polynomials are used for gaussian probability approximations, but are not suitable for other probability distributions. Based on Wiener's approach, generalized polynomial chaos (gPC) expansion has been developed in 2002 to study processes which involved random variables with different probability distributions. It uses the full range of the Askey scheme of orthogonal polynomials in respect with their corresponding probability distributions to achieve up to exponentially fast convergence rates (24).

Stochastic processes are expressed as orthogonal polynomials of the input random variable. A stochastic process $Z(t, x, \zeta)$, depending on vector ζ containing random variables, time t and state vector x , can be represented as an infinite expansion:

$$Z(t, x, \zeta) = \sum_{k=0}^{\infty} u_k(t, x) \Phi_k(\zeta) \quad (3.1)$$

with $u_k(t, x)$ as the coefficients and Φ_k the orthogonal polynomial base for the corresponding random variable in ζ_x . The base selection is optimally chosen according to table 3.1 because the orthogonal polynomials' weight functions correspond to the probability density functions of the respective distributions (21).

	Random Variables ζ	Wiener-Askey chaos $\Phi(\zeta)$	Support
Continuous	Gaussian	Hermite-chaos	$(-\infty, \infty)$
	gamma	Laguerre-chaos	$[0, \infty]$
	beta	Jacobi-chaos	$[a, b]$
	uniform	Legendre-chaos	$[a, b]$
Discrete	Poisson	Charlier-chaos	$\{0, 1, 2, \dots\}$
	binomial	Krawtchouk-chaos	$\{0, 1, 2, \dots, N\}$
	negative binomial	Meixner-chaos	$\{0, 1, 2, \dots\}$
	hypergeometric	Hahn-chaos	$\{0, 1, 2, \dots, N\}$

Table 3.1: Optimal polynomial choices based on random distribution (24)

An infinite expansion, while accurate, is not typically used in practice due to computational limitations. Dealing with a finite set of p random variables with maximum polynomial order n allows truncation of the above formula to K terms in order to approximate the result (24):

$$Z(t, x, \zeta) \approx \sum_{k=0}^K \hat{u}_k(t, x) \Phi_k(\zeta) \quad (3.2)$$

with $K + 1 = \frac{(n+p)!}{n!p}$

$\hat{u}_k(t, x)$ deterministic coefficients (spatio-temporal component)

ζ , vector containing the random variables

$\Phi_k(\zeta)$ orthogonal polynomials for the random component

Smith et al. ,2014, stated that stochastic collocation can be used to specify $u_k(t, x)$ (25). The generalized polynomial chaos expansion, used in combination with stochastic collocation, reduces computational requirements while retaining accuracy and convergence rate of gPC (26). Stochastic collocation uses a set of nodes and weights based on polynomial interpolations and evaluates the model on these nodes. For a set of n random variables, evaluated in model Z with model function $f(t, x, p^j)$ on a given set of collocation points Q , the deterministic coefficients $u_k(x, t)$ in 3.1 can be approximated with

$$\hat{u}_k(t, x) = \frac{1}{\gamma_k} \sum_{j=1}^Q f(t, x, p^j) \Phi_k(p^j) \alpha^j \quad (3.3)$$

$\gamma_k = E[\Phi_k^2]$ are the normalization constants of the polynomial basis and are used to normalize the orthogonal polynomials (27).

$\{p^j, \alpha^j\}_{j=1}^Q$ are a set of nodes and weights. $p^j = (p_1^j, \dots, p_n^j)$ and α^j denote the j -th node and its associated weights. Nodes and weights have to be chosen so that the integration rule approximates the real gPC expansion well enough for sufficiently smooth functions $f(p)$ and a sufficiently large number of collocation points. The nodal set has to be chosen such that the integration rule is accurate and computationally efficient (28).

The node-weight set $\{p^j, \alpha^j\}_{j=1}^Q$ is generated in the data assimilation phase based on the integration rule applied. Uncertainty is propagated by evaluating the model $U(p^j)$ and the orthogonal basis $\Phi_k(p^j)$. During the certification phase, the output of interest can be analyzed (28).

According to Xiu et al., 2010, the coefficients u_k of the polynomial expansion can be used to derive the mean μ_{u_n} and the variance σ^2 of the QoI with

$$\mu_{u_n} \approx \frac{1}{\gamma_0} \sum_{j=1}^Q f(p^j) \Phi_0(p^j) \alpha^j = \hat{u}_0(t, x) \quad (3.4)$$

and

$$\sigma^2 \approx \sum_{i=1}^K \left(\sum_{j=1}^Q \hat{u}_i(p^j) \Phi_i(p^j) \alpha^j \right)^2 \approx \sum_{i=1}^K \hat{u}_i^2(x, t) \quad (3.5)$$

with Q being the number of collocation points and K the order of the polynomial chaos expansion (27).

The so-called *Curse of Dimensionality* dictates how many uncertain variables can be propagated with reasonable computational expense. Global sensitivity analysis is performed to identify impactful uncertain variables and potentially reduce the dimensionality by lowering the number of uncertain variables (16).

3.2 Global Sensitivity Analysis

While Uncertainty Quantification focuses on *quantifying* the uncertainty in model output, sensitivity analysis studies how uncertainty in model input can be *apportioned* to the different uncertain input variables (29). The uncertain parameters are propagated through the model as detailed in chapter 3.1.2. Sensitivity analysis on the model output can determine the influence level of parameters on the model uncertainty. The used uncertainty quantification method, stochastic collocation with pseudo-spectral approach, provides a fast and efficient calculation of the output of interest's variance (3.5). This is the reason a variance-based sensitivity analysis was chosen for this work.

Saltelli et al., 1996, describes possible use cases of variance-based sensitivity analysis. *Factor Prioritization* identifies factors that contribute the most to the total output variance. This ties into another use, the *Factor Fixing*. It identifies factors that, across their range of variability, have no significant contribution to the output variance. These factors can be fixed at any given value of their variance range, reducing the number of uncertain factors in return (30). Different global sensitivity indices, also called *Sobol' indices*, can be calculated: first-order, higher-order and total sensitivity index (29).

The first-order index describes the main effect contribution of each input factor compared to the output covariance.

Let $V(f_i(X_i)) = V[E(Y|X_i)]$ be the variance of factor X_i and $V(Y)$ the unconditional model variance. Then the first-order sensitivity index S_i is:

$$S_i = \frac{V[E(Y|X_i)]}{V(Y)} \quad (3.6)$$

Higher-order sensitivity indices represent the interaction effects between two or more factors. It helps in detecting model behaviour when subjected to particular input combinations. It is calculated by measuring the joint effect of factors and subtracting the first-order effects of each individual factor.

$$V_{ij} = V(f_{ij}(X_i, X_j)) = V(E(Y|X_i, X_j)) - V(E(Y|X_i)) - V(E(Y|X_j)) \quad (3.7)$$

It describes the second-order effect V_{ij} for the factors X_i, X_j . This formula can be easily expanded to include more factors (29).

The total sensitivity index S_{Ti} accounts for the total contribution of a factor X_i on the output variance: the first-order effect and its interaction effects with all other factors (30).

$$S_{Ti} = \frac{E_{X_{\sim i}}(Var_{X_i}(Y|X_{\sim i}))}{Var(Y)} = 1 - \frac{Var_{X_{\sim i}}(E_{X_i}(Y|X_{\sim i}))}{Var(Y)} \quad (3.8)$$

First-order and total sensitivity indices are calculated in this work to identify the impact of uncertain random variables. In accordance to Sobol' these variables are fixed at a certain value within the variance interval and will not be further used as uncertain input, thus reducing the dimensionality of the simulation (31).

4 Uncertainty Quantification in LARSIM

4.1 Sources of uncertainty in LARSIM

Uncertainty is always prevalent in complex models, such as the water balance model LARSIM. High quality input data and coherent spatial models are very important for reliable output data. The model uses numerous input data to simulate the water balance. Figure 4.1 provides an overview of the different input and output files of LARSIM. Detailed descriptions are in the LARSIM Online-Help (32).

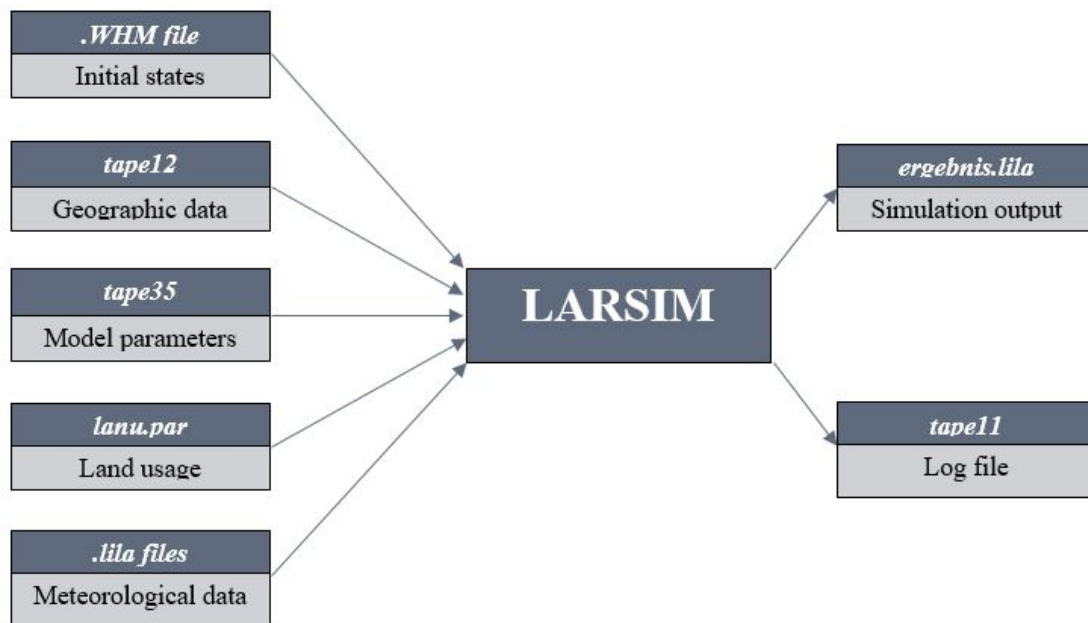


Figure 4.1: Interaction of classes within the prototype

LARSIM requires data about the initial state of its hydrological storages within the simulated area. These states are based on previous simulation runs and can be stored in *.WHM* files. The model uses this data, which usually contains storage data for the two

days prior to the simulation start, as a starting condition. Since this data is generated based on prior simulation runs, the predicted values could be uncertain (33).

The underlying spatial model is using soil maps to parametrize the observed model area. These parameters are stored in the *tape12* file. A provided soil map is used to determine the different soil variables in each compartment, which already has a built-in uncertainty analysis for extreme cases. Mitterer, 2015, analyzed the uncertainty for variable soil data for LARSIM and suggests that the data used is inaccurate in at least some cases (9).

Weather forecast is inherently prone to uncertainty due to the high dimensionality of the climate system. Its output is used for meteorological data input in LARSIM, like precipitation and temperature, and can cause uncertainty within the model. Especially faulty precipitation data impacts the result to a great extent (34). Finally the physical parameters used within the model have to be calibrated to reflect the real world conditions as closely as possible. Calibration is usually performed based on data of a long period of time (several years) with varying discharge values. The goal is to calibrate the parameters such that the simulated runoff values approximate the measured runoff as closely as possible. The calibration guide provides documentation of LARSIM's model calibration (33).

The calibration process is laborious and the result is susceptible to uncertainty due to imperfect input data. This work concentrates on the pre-calibrated parameters for the snow module and exposes them to uncertainty.

4.2 Calibration Analysis

The pre-calibrated parameters provided by the Landesamt für Umwelt are used as a basis for uncertainty quantification. Parameters for the snow model, among others, are stored in the configuration file *tape35* for each gauge station in the area Regen. Two approaches for the modification of the base values have been considered. One approach is to modify the parameters individually for each gauge station, thus being able to observe the runoff interaction between compartments with varying values. However, there are 22 gauge stations within the area. Stochastic collocation with $Q = 10$ collocation points for even only one uncertain variable would result in 22^{10} simulations. This is outside the scope for this work so an alternative has been chosen. The same uncertainty value generated by stochastic collocation will be added to the pre-calibrated gauge value in *tape35* uniformly instead for each gauge. This static change allows for observations of a larger number of uncertain parameters in a reasonable timeframe.

4.3 LARSIM configuration setup

The options in the main configuration file *tape10* were adjusted to accommodate an accurate analysis of the snow module. The options selected for the snow model are in listed in table 4.1.

Model	Options in <i>tape10</i>	Description
Snow module	SCHNEE: KNAUF, 2006	Enhanced Knauf procedure
	SCHNEE: OBERFL-TEMP	Estimates snow cover temperature
	BODENTEMPERATUR	Dynamic ground temperature
	EINGABE KNAUF-PARAMETER	Enables absorption parameter
	SCHNEEALBEDO	Calculates snow albedo
	SCHNEEREGEN	Interpolation of snow-rain mixture
	MAX. SCHNEE-RET. TAPE35	Enables snow retention parameter
	SNOW-COMPACTION 3	Degradation of snow cover

Table 4.1: Excerpt of *tape10* configuration options

OPTION: SCHNEE: KNAUF,2006 defines the snow model used and provides the most precise calculation of the energy balance of snow available in LARSIM. This model also allows the calculation of an estimate temperature at the snow surface, which can differ significantly from the mean snow temperature (OPTION: SCHNEE: OBERFL-TEMP). The ground heat flux is dynamically calculated for the selected model (OPTION: BODENTEMPERATUR). The model parameters for snow melt can be set for each gauge individually by enabling OPTION: EINGABE KNAUF-PARAMETER. This parameter describes the absorption coefficient of fresh snow in combination with OPTION: SCHNEEALBEDO. OPTION: MAX. SCHNEE-RET. TAPE35 enables individual water retention parameters for each gauge. OPTION: SNOW-COMPACTION 3 enables a modified version of the *Bertle-procedure* to model the degradation of snow and is required for individual water retention parameters. A detailed description of all available options is available at the LARSIM Online-Help (32).

The four main parameters that influence the snow model are $Tmit_Sr$, $Tspann_Sr$, $SRet$ and $Abso$. As described in section 2.2.1, precipitation can be interpolated as a mixture of snow and rain within the interval $[(Tmit_Sr - \frac{1}{2}Tspann_Sr), (Tmit_Sr + \frac{1}{2}Tspann_Sr)]$ where $Tmit_Sr$ describes the middle of the interval with $Tspann_Sr$ being the range of the interval. The absorption parameter $Abso$ controls the absorption rate of short radiation waves by the snow cover. Lastly, the parameter $SRet$ describes the retention rate (in %) of liquid water in the snow pack (33).

The unaltered *tape35* file provided calibrated parameter values for *Abso* and *TGr* (the threshold parameter for precipitation falling as rain or snow). These *TGr* values were used in this work and used as initial *Tmit_Sr* values. Values for *Tspann_Sr* and *SRet* were not in *tape35* and had to be added manually. The starting values for these parameters, as outlined in the calibration guide (33), were used for all gauges which is *Tspann_Sr* = 4 and *SRet* = 30.

Parameters	Uniform distribution [a, b]	Value range
<i>Tmit_Sr</i>	$\mathcal{U}(-2, 3)$	[-3, 2]
<i>Tspann_Sr</i>	$\mathcal{U}(-2, 4)$	[2, 8]
<i>Abso</i>	$\mathcal{U}(-0.08, 0.1)$	[0.02, 0.25]
<i>SRet</i>	$\mathcal{U}(-25, 17)$	[5.0, 47.0]

Table 4.2: Distributions and value ranges for uncertain parameters

The calibration guide (33) provides guidelines for reasonable parameter values by setting value intervals. Uniform distributions for each parameter are used that span the full range of the interval. The chosen distributions and the value ranges of each parameter are displayed in table 4.2 The values present in *tape35* for each of the four parameters (see table 4.2) are subjected to uncertainty by adding the corresponding nodes generated with stochastic collocation to them. This generates values that are distributed over the whole range of the provided interval. Extreme values at the interval ends are also taken into account.

4.4 Prototype Setup for Uncertainty Quantification in LARSIM

Two different types of simulation runs were executed: global sensitivity analysis and statistical evaluation runs. The goal of the global analysis run was to identify the most impactful uncertain variables within the snow module. Therefore all four parameters were subjected to uncertainty and the sensitivity indices were analyzed on the output of interest. A comparatively low number of collocation points was chosen to reduce computational effort which stems from the curse of dimensionality.

After identifying the two most impactful parameters, only these uncertain parameters were propagated through the model for the statistical evaluation. These runs used a significantly higher number of collocation points to achieve more precise output of interest for statistical analysis.

The parameters used for this work are uniformly distributed around pre-calibrated values. Stochastic collocation with pseudo-spectral approach was used for all runs. According to table 3.1, Legendre-Chaos was used as polynomial basis $\Phi(\zeta)$. Gauss quadrature is used as an integration rule for node and weight generation, and the order of the polynomial expansion is set to $K = 6$.

The area Regen was selected as model area for this work. Regen is a district in the state Bavaria. Its 950 km² are fully covered by the Bavarian Forest, which is split in two parts by a rocky landscape made of quartz, the "Pfahl". The satellite image 4.2) illustrates the landscape. The area is named after the river Regen which has one of its headstreams, Schwarzer Regen, streaming through the whole area from west to east. The 14 nature reserves within the



Figure 4.2: Satellite image of the district Regen (35)

district guarantee wide areas of untouched nature (36, 37). The combination of untouched nature and the interconnectivity of runoff through various streams and rivers could lead to interesting results, which is why this area was chosen as model area.

A timeframe with extreme weather conditions which emphasize the impact of the snow module on runoff was selected for the simulation runs. Spring 2013 was not a usual spring. Starting out with summerly temperatures of up to 18 °C at the beginning of March, cold storm "Xaver" caused winter-like conditions. According to the German Meteorological Service (DWD), 2013 is ranking 6th coldest March since the beginning of recording 1881.

There was a closed snow cover across most regions of Germany.

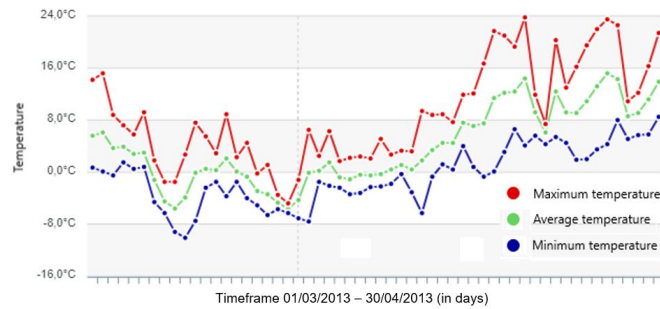


Figure 4.3: Temperatures March/April 2013 in Zwiesel (district Regen) (38)

Figure 4.3 illustrates the temperature differences in the timeframe at the gauge station Zwiesel in the area Regen. Following this very cold March was a warm April, which resulted in heavy snowmelt (39, 40). It was assumed that subjecting snow module parameters to uncertainty could lead to flood events during this time. Therefore, the timeframe of 1st of March to 31st of April was chosen to examine the behaviour of LARSIM in these unusual conditions.

The configuration setups for both types of execution are displayed in table 4.3. Global sensitivity analysis runs were performed to calculate sensitivity indices for each uncertain parameters to identify their impact on the output of interest (see chapter 6.1). Statistical Evaluation runs were afterwards performed with only the two most impactful parameters with the other two parameters at fixed values in an attempt to quantify the uncertainty (see chapter 6.2).

Execution type	Configuration type	Setup	Values
Sensitivity Analysis (used in 6.1)	Uncertain Parameters	<i>Tmit_Sr</i>	$\mathcal{U}(-2, 3)$
		<i>Tspann_Sr</i>	$\mathcal{U}(-2, 4)$
		<i>Abso</i>	$\mathcal{U}(-0.08, 0.1)$
		<i>SRet</i>	$\mathcal{U}(-25, 17)$
	Stochastic Collocation	Polynomial Basis	Legendre-Chaos
		Integration rule	Gauss quadrature
		Collocation points	$Q = 11$
		PCE order	$K = 6$
Statistical Evaluation (used in 6.2)	Uncertain Parameters	<i>Tmit_Sr</i>	$\mathcal{U}(-2, 3)$
		<i>SRet</i>	$\mathcal{U}(-25, 17)$
	Fixed Parameters	<i>Tspann_Sr</i>	4
		<i>Abso</i>	<i>tape35</i> values
	Stochastic Collocation	Polynomial Basis	Legendre-Chaos
		Integration rule	Gauss quadrature
		Collocation points	$Q = 30$
		PCE order	$K = 6$

Table 4.3: Prototype configuration setup

The operations of generating these values, simulating the runoff with LARSIM and processing the output is handled by the developed prototype, which will be explained in the next chapter.

5 Prototype for Uncertainty Quantification on LARSIM

5.1 Frameworks

5.1.1 Uncertainty Quantification Execution Framework

The Uncertainty Quantification Execution Framework (UQEF) was developed by Florian Künzner. The framework was designed to handle the computation of uncertainty in the variables and the execution of the provided model. It supports uncertainty quantification by stochastic collocation with pseudospectral approach and classic Monte Carlo simulation runs. The library *chaospy* provides functionality for *assimilation* and *certification* of uncertainty quantification, and UQEF was developed to provide the missing functionality of uncertainty *propagation*. Three different execution modes are provided: linear, parallel and execution with the Message Passing Interface (MPI) for parallel computing.

The design of UQEF allows the user to focus on processing the input and output of the model itself while the framework manages the correct execution. It has been used for scientific research in a number of publications, for example in using surrogate models for uncertainty quantification (41). The Uncertainty Quantification Execution Framework provides the foundation of this work. An official publication by Künzner detailing the exact functionality of UQEF is in progress.

5.1.2 Chaospy

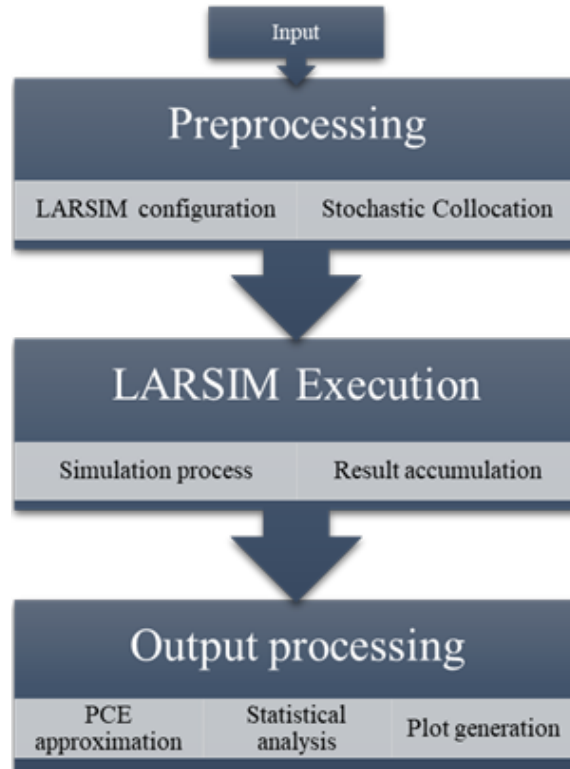
Chaospy is an open source library for the modeling of uncertainty using Monte Carlo and polynomial chaos expansion. It supports quadrature integration and various polynomial and distribution functions. It also provides statistical analysis tools to analyze distributions and polynomials. The Gaussian quadrature function of *chaospy* is invoked by UQEF to generate nodes and weights for the provided probability distributions.

After the results are collected, the polynomials are approximated and statistical values like mean, standard deviation and the sobol indices are generated by using chaospy functions. A detailed description of the functionality of chaospy can be found in the documentation or the publication (42, 43)

5.2 Uncertainty Quantification Prototype

5.2.1 Overview

The prototype was developed in Python Version 3. It was designed to be able to manipulate model input data without having to alter the source code for every execution.



The prototype was logically structured into three major components: Preprocessing, LARSIM Execution and Output processing (figure 5.1).

Preprocessing handles the input data, adjusts LARSIM configuration files accordingly and prepares nodes and weights for stochastic collocation.

LARSIM Execution runs model simulations for each uncertain parameter combination according to the specified execution method and collects the simulation results.

Output processing approximates polynomials for the collected result data on each timestep based on distribution, nodes and weights from stochastic collocation, calculates statistical measurements and plots the final results.

Figure 5.1: Logical structure of the simulation prototype

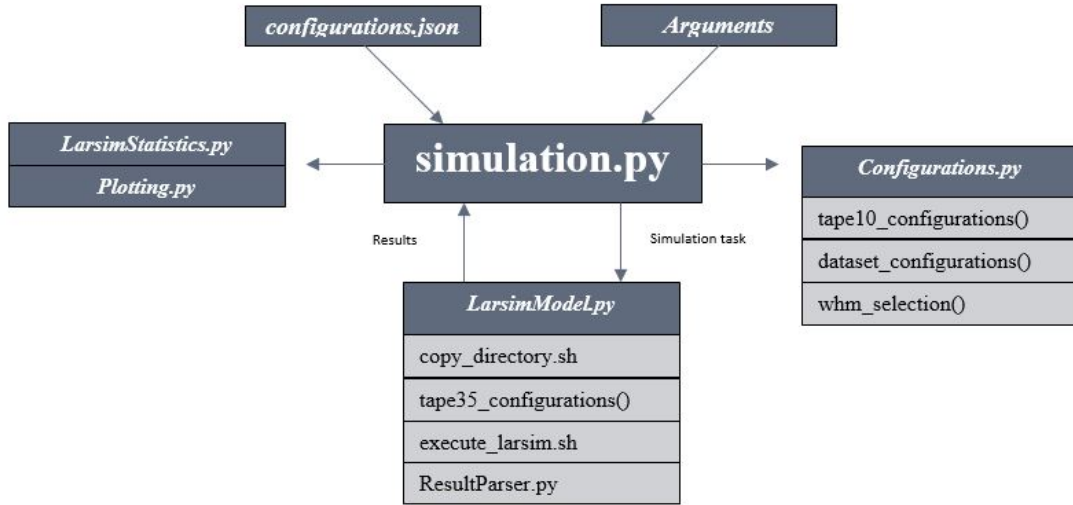


Figure 5.2: Interaction of classes within the prototype

Figure 5.2 describes the flow of the prototype and the interaction between the classes and functions which supplement the main file *simulation.py*. The most important functions within the prototype are explained in detail.

5.2.2 Preprocessing

The timeframe of the simulation and the investigated uncertain parameters are defined in the JSON-file *configurations.json*. The timeframe object consists of start and end date. The dates are parsed into a datetime object which is used to manipulate the LARSIM input data. *Tape10* is adjusted to the new timeframe. LARSIM also requires the correct initial state (*.WHM*) file and meteorological data (*.lila*) files in its working directory to execute the simulation. A full data set of initial states and meteorological data for each gauge in the area Regen, ranging from 2003 to 2018, was provided by the Landesamt für Umwelt. These files need to be parsed, selecting only the data within the time interval, to reduce execution time on LARSIM and disk space required. These generated files are then moved to the LARSIM working directory.

The uncertain variables consist of a name, a probability distribution ("normal" or "uniform" distribution is currently supported), attributes defining the distribution and a value interval (see table 3.1).

Listing 5.1: Uncertain variable Tmit_Sr

```
{
  "name": " Tmit_Sr",
  "distribution": "uniform",
  "mean": -1,
  "std": 0.5,
  "uniform_low": -2,
  "uniform_high": 3,
  "lower_limit": -3,
  "upper_limit": 3
}
```

These variables are parsed from the configuration file and the distributions are set based on the defining attributes. Variable names and distributions are saved in an array of tuples which is forwarded to the UQEF to generate nodes and weights required for the stochastic collocation method.

Listing 5.2: Variable distribution forwarding

```
distributions = []
for i in data["Variables"]:
    if i["distribution"] == "normal":
        distributions.append((i["name"], cp.Normal(i["mean"], i["std"])))
    elif i["distribution"] == "uniform":
        distributions.append((i["name"], \
            cp.Uniform(i["uniform_low"], i["uniform_high"])))
nodeNames = []
for items in distributions:
    nodeNames.append(items[0])
simulationNodes = uqef.simulation.Nodes(nodeNames)
for items in distributions:
    simulationNodes.setDist(items[0], items[1])
```

The *simulationNodes* object is generated by UQEF and contains a multidimensional array of collocation points that is used during the simulation execution.

Settings concerning the simulation execution and the selection of the uncertainty quantification method are handled with command-line arguments. These settings are parsed in the main program and forwarded to the UQEF which executes the simulation based on these settings. All available argument parameters are listed in table 5.1.

Options	Arguments	Description
Uncertainty Quantification	--uq_method	"sc": Stochastic Collocation "mc": Monte Carlo
	--sc_q_order	Collocation points in each direction (Q)
	--sc_p_order	Order of PCE (K) (see 3.3)
	--mc_numevaluations	Number of Monte Carlo simulations
Execution	default: linear	Execution with only one CPU core
	--num_cores	Defines amount of CPU cores
	--parallel	Execution with set number of cores
	--mpi	Execution with MPI
	--mpi_method "old/new"	MPI methods available in UQEF
	--mpi_combined_parallel	Enables combined parallel option
General settings	--or "dir_name"	Output directory for Plots/Data
	--model "model_name"	Defines the analyzed model

Table 5.1: Command-line arguments for the prototype

5.2.3 LARSIM Execution

After the preprocessing, simulations are executed. The number of simulation runs depends on the number of collocation points and increases exponentially by the number of uncertain variables. Parallel processing is greatly encouraged to reduce the processing time, which is why MPI-based and parallel execution was used. Work orders for the different simulation runs are distributed to threads by the UQEF based on the execution strategy defined. In this work, the total work load is distributed dynamically to all available CPU cores. Each core launches an individual instance of the class *LarsimModel* where it executes the LARSIM simulations for the assigned tasks in parallel. After it returns the combined results, a new set of tasks is assigned to the cores until the total work load is processed.

LARSIM configuration files are stored in the folder *WHM Regen* within the LARSIM framework. The LARSIM executable requires to be executed from the current working directory which contains the necessary simulation configuration files. This behavior cannot be changed efficiently to account for parallel simulation runs. Each task modifies *tape35* to integrate its simulation parameters. If all cores are using the same working directory, they would interfere with each other. In order to run multiple simulations in parallel with differing configuration files, a copy of *WHM Regen* has to be created for each task.

Since each task only modifies *tape35*, the other configuration files can be directly copied to ensure a correct execution. UQEF provides a task counter which was used to number the copied folders. The uncertainty parameters for each task are added to the corresponding *tape35* values in their respective folders. Working directories are changed per process within Python, not per thread. Different instances of LARSIM are run on each thread simultaneously, so system calls were used for each execution to change the working directory to the correct task folder and start the LARSIM executable. System calls are individual to each thread and do not interfere with parallel LARSIM executions.

Listing 5.3: Pseudocode for `LarsimModel.py` functionality

```
for each task do:
    call(["./copy_directory.sh"])
    modify_tape35(task_parameter, folder_path)
    call(["./execute_LARSIM.sh"])
    output_of_interest = result_parser(folder_path + "/ergebnis.lila")
    results.append(output_of_interest)
return results
```

LARSIM stores simulation results in the file *ergebnis.lila*. This list-based file contains the simulation output of three meteorological parameters for each gauge station: precipitation, measured runoff and simulated runoff. The internal time step for LARSIM is set to 1-hour intervals, providing output values for every hour of the simulation timeframe for each parameter.

The result file is parsed and compiled into a dictionary which contains data in the following layout:

Listing 5.4: Result layout

```
result = {
    station: {
        precipitation: [values],
        runoff simulation: [values],
        measured runoff: [values],
    }
}
```

Each of these task results are collected and appended to an array which contains all task results. This array is then forwarded to data analysis for further processing.

5.2.4 Output processing

The collected result array contains dictionaries of all simulation runs. These dictionaries need to be merged to have an array of all simulation values for each timestep, which is required to approximate the PCE at every timestep. This is accomplished by iteration over each dictionary, collecting the individual value at each timestep and storing them in an array.

Listing 5.5: Merge process of simulation values

```
collected_result = {}
for identification in rawSamples[0]:
    collected_result[identification] = []
    for i in range(0, len(rawSamples[0][identification]["runoff simulation"])):
        temp = []
        for j in range(0, len(rawSamples)):
            temp.append(rawSamples[j][identification]["runoff simulation"][i])
        collected_result[identification].append(temp)
```

The simulated runoff parameter is the output of interest while precipitation and measured runoff are not altered by the uncertain variables. Therefore only the simulated runoff values have to be approximated to polynomials in each timestep. The prototype uses a function provided by chaospy to calculate these polynomials based on the value array and the corresponding set of nodes and weights.

Statistical values are calculated for each timestep based on the given polynomials. First order and total sensitivity indices are calculated for global sensitivity analysis. Mean, standard deviation as well as 10th (P10) and 90th percentiles (P90) were calculated for a detailed plot analysis.

The prototype automatically generates several plots based on the library *matplotlib* (44) for analysis: a set of plots for each gauge station, one containing the mean, standard deviation and P10/P90 of the simulated runoff values plotted against the measured runoff for the whole time interval. A second set of plots displays the absolute and relative error of the simulated runoff compared to the measured runoff. The final set of plots shows both sensitivity indices for sensitivity analysis. It also generates an overview of mean simulated runoff values of all gauge stations in the area Regen to quickly identify points of interest.

The plots and the raw data are saved in the corresponding output folder defined in the input arguments.

5.3 Execution on High Performance Computing Systems

The simulation runs were executed on the Leibniz-Rechenzentrum linux cluster *Cool-Muc2*. The cluster is equipped with 384 28-way Haswell-based CPUs (Intel Xeon E5-2697 v3) for a total of 10752 cores and 24.5 TB RAM. The solutions for Python (python/3.5_intel) and MPI (mpi.intel/2017) were provided by Intel. In order to start an execution run, it has to be submitted to the implemented cluster management and job scheduling system *SLURM* (45). The submission is facilitated with a shell script that specifies the arguments for the cluster and the prototype and generates a batch file which is then forwarded to *SLURM*.

The cluster nodes assigned to the simulation job feature 28 CPU cores and 55GB RAM each. The average runtime of one LARSIM execution was 32 seconds. The total prototype runtime depends greatly on the amount of uncertain variables and collocation points used. The total runtime for the global sensitivity analysis run (4 uncertain variables, 11 collocation points) was 59h for the execution of $11^4 = 14641$ LARSIM simulations and the statistical evaluation. Parallel execution on one node with 20 cores was used for this run to ensure stability on the cluster. Executions with 2 uncertain variables and 25 collocation points had a runtime of 5.2h, using 3 nodes on the cluster for MPI execution.

6 Analysis of the Output of Interest

6.1 Global Sensitivity Analysis

The output of interest based on the simulation runs detailed in chapter 4.4 was analyzed for all gauge stations. The gauge station Lohberg (station id: LOHB) is used for demonstration purposes in the following.

The first aim is to identify the most impactful parameters of the snow module from a stochastic perspective. Therefore, uncertainty quantification is used in the simulation run with all four uncertain variables (see table 4.3) to perform global sensitivity analysis. First order sensitivity and total sensitivity indices were calculated for the whole timeframe. Figure 6.1 displays the impact of each uncertain parameter on the total runoff (in 100%) by visualizing the total sensitivity indices.

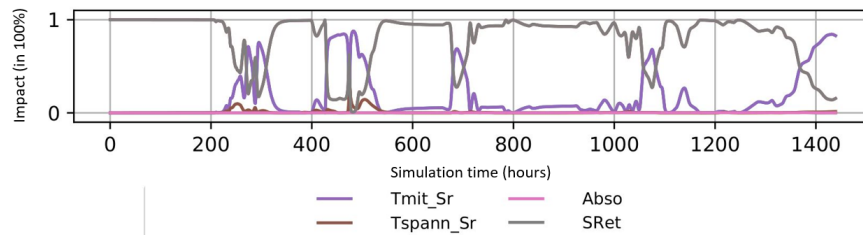


Figure 6.1: Total sensitivity of uncertain variables in station LOHB

The data indicates that the two most impactful parameters are *SRet* and *Tmit_Sr*, while *Tspann_Sr* and *Abso* have little to no impact on the output of interest. An impact shift of parameters *SRet* and *Tmit_Sr* at certain time intervals can be observed. Comparing the time intervals of the impact shift with the temperatures in figure 4.3 could lead to the following assumption: A correlation between air temperature and sensitivity values of both parameters is assumed because *Tmit_Sr* becomes more impactful when the temperatures are around or below 0 °C. Freezing temperatures could reduce the impact of the water retention parameter *SRet* because there is less liquid water in the snow cover.

As the temperatures rise consistently above the freezing point, parameter $SRet$ retains its position as most impactful parameter.

Based on the obtained information from global sensitivity analysis, the statistical evaluation runs are performed with only the two most impactful uncertain parameters. This enables the execution of uncertainty quantification with a significantly higher number of collocation points within reasonable computational expense.

6.2 Statistical Evaluation

6.2.1 Simulation data

The following simulations are performed according to the statistical evaluation execution type of table 4.3. The output of interest, the water runoff, was statistically analyzed based on the PCE for each timestep. Mean, standard deviation and the interval (10th percentile (P10), 90th percentile (P90)) were calculated. Validated runoff measurements were parsed from the corresponding *.lila* file. These measurements (measured runoff) and the runoff values of a simulation run without uncertain parameters (unaltered runoff) were used as baseline values for comparison. Mean, percentiles and standard deviation of the simulated runoff as well as measured runoff of the whole timeframe are displayed in figure 6.2.

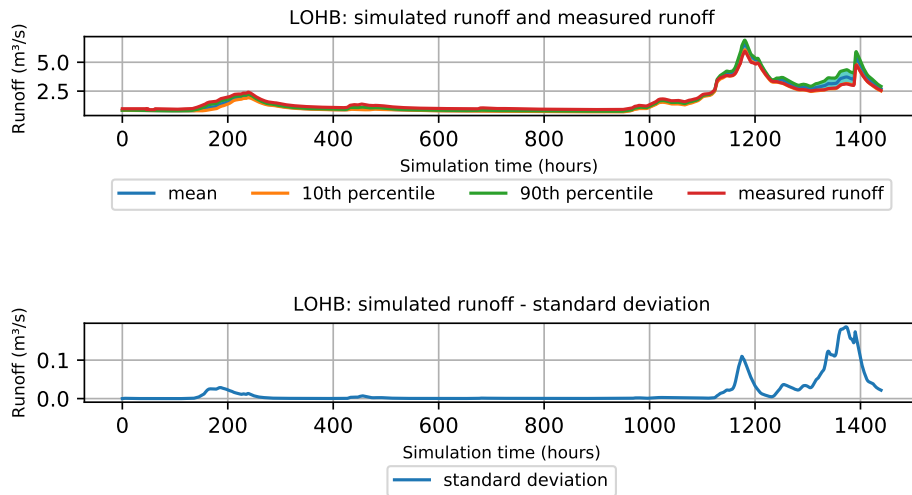


Figure 6.2: Station LOHB: Simulated runoff (mean, percentiles, standard deviation) and measured runoff

The data shows very little uncertainty in the first 1100 hours of the simulation with noticeably increased uncertainty in the last quarter of the simulation. The standard deviation was greatly increased in the time interval of around April 15th to April 27th (hours 1150 - 1400) across all stations, which was also reflected in the large interval of (P10, P90) at the time. This behavior could be caused by the sudden increase and decline in runoff values at the hour-interval (1150, 1250), where the runoff values for LOHB increased by up to 271% of the initial runoff. The equally rapid decline after the peak is likely to have resulted in the displayed uncertainty. The data indicates a similar behavior at sudden runoff changes across all stations.

The mean values of the simulated run are denoted as simulated runoff from now on. The comparison between unaltered and simulated runoff in figure 6.3 showed that both simulations deviate from the measured values for most of the considered timeframe, but the error is noticeably increased during the last quarter of the timeframe.

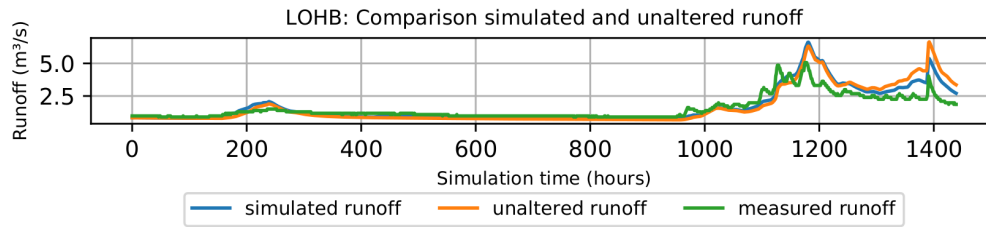


Figure 6.3: Station LOHB: Simulated runoff, unaltered runoff and measured runoff

Taking a closer look at the previously mentioned time interval shows the large value discrepancy between both simulated runs compared to the measurement (figure 6.4). Since even the unaltered runoff with pre-calibrated parameters deviates from the measured runoff, it can be assumed that there is some calibration error which prevents an accurate forecast of runoff in the investigated timeframe. The extreme weather conditions in the timeframe are also a likely cause of the poor simulation quality.

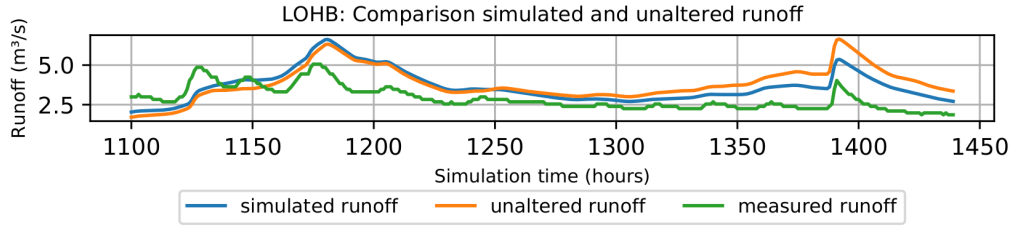


Figure 6.4: Station LOHB: April 15th-27th: Comparison of simulated runoff and measured runoff

The deviation is further analyzed by calculating the approximation error between the three data sets: simulated runoff, unaltered runoff and measured runoff. The relative approximation error ϵ is calculated according to this formula:

$$\epsilon = \left| \frac{v - v_{approx}}{v} \right| \quad (6.1)$$

with v : being the measured runoff and v_{approx} being the predicted runoff values from LARSIM simulations. The percentage error ($100\% \cdot \epsilon$) is used here as it provides a clearer display (46).

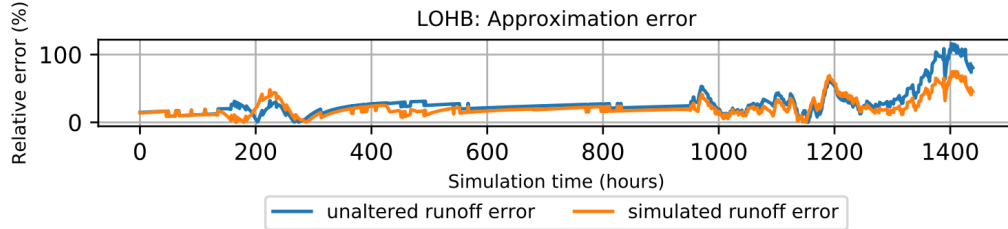


Figure 6.5: LOHB: Relative error of simulated runoff and unaltered runoff to measured runoff

Figure 6.5 shows the relative error of simulated and unaltered runoff compared to measured runoff in the timeframe. The simulated runoff error is generally lower than the unaltered runoff error throughout the timeframe, especially during the mentioned time interval. The data shows however that the simulated values do not approximate the measured values well. In order to compare the error found in station LOHB with the other stations in the area, average error values for all stations have been calculated.

The following values have been calculated over the timeframe to evaluate the relative error: mean, median, minimum and maximum error. The values for station LOHB and the average values across all stations for simulated runoff and unaltered runoff can be found in table 6.1.

Station	Measurement	Simulation	Values (in %)
Station LOHB	Mean error	Simulation	21.35
		Unaltered	23.79
	Median error	Simulation	18.49
		Unaltered	23.79
	Minimum error	Simulation	0.01
		Unaltered	0.01
All Stations	Maximum error	Simulation	75.28
		Unaltered	116.14
	Mean error	Simulation	21.48
		Unaltered	22.73
	Median error	Simulation	17.73
		Unaltered	18.37
	Minimum error	Simulation	0.03
		Unaltered	0.4
	Maximum error	Simulation	282.79
		Unaltered	360.34

Table 6.1: Approximation error comparison

The data displayed in table 6.1 indicates that the difference in simulation quality between simulated and unaltered runoff is minor across all stations. While the maximum error was noticeably lower on the simulated run, the mean error difference between simulated and unaltered runoff is insignificant. This indicates that the error could not be reduced noticeably by subjecting the snow module parameters to uncertainty. The mean error for both simulated runoff (21.48%) and unaltered runoff (22.73%) itself however is very high, which hints at a high overall model uncertainty.

6.2.2 Discussion

The uncertainty of the simulated runoff is best observed during and after sudden changes in runoff values. The hypothesis was proposed in chapter 2 that there might be a correlation between parameter uncertainty, originating from calibration, and uncertain simulation output. The simulation output provides a range of values, and some values in the (P10, P90) interval are closer to the actual measurements. On this basis, snow module parameters could probably be improved to increase simulation quality, but further testing is required for a definitive answer.

Simulated runoff and unaltered runoff showed a noticeable discrepancy from the measured runoff, especially in the time interval of April 15th to April 27th. The difference in approximation error between both simulations is negligible across all stations. While the error could not be considerably reduced by the uncertain parameters of the snow module, the poor simulation accuracy suggests that there is uncertainty in the model. It could be the case that the snow module did not have the high impact on the total runoff as previously expected. The error could then be the result of uncertainty in other pre-calibrated parameters.

Another reason for the discrepancy could be the timeframe chosen for the simulation runs. Parameter calibration, as described in chapter 4, is performed on large data sets of several years. It is assumed that this calibration was not optimized to cover the extreme weather conditions of the timeframe, which resulted in the large approximation error.

7 Conclusion

In this work a prototype was developed for the quantification of uncertainty in the water balance model LARSIM. The calibrated parameters of LARSIM's snow module were subjected to uncertainty by modifying model parameter data in order to produce uncertain simulation output. The uncertainty was propagated with the prototype by using stochastic collocation with pseudo-spectral approach. Global sensitivity analysis on the four snow module parameters indicated that parameters $Tmit_Sr$ and $SRet$ have the most impact on the uncertain output. Further simulation runs with these two parameters were executed and statistical values were calculated in an effort to quantify the parametric uncertainty. While uncertainty in the snow module could not be proven definitively, the large approximation error displayed in the investigated timeframe indicates uncertainty in the model.

The prototype should be further optimized to allow for more consistent runs on the HPC system. The prototype does not currently support large amounts of simulation runs (>3000) and further development is advised to investigate this issue. The prototype currently enables the manipulation of configuration parameters in *tape35*. Additional functionality could be added to facilitate the manipulation of the other LARSIM input files for the uncertainty quantification of other modules. Multiprocessing for the certification stage should also be integrated to improve the runtime.

In order to further quantify the uncertainty in the model, it is recommended to repeat the analysis of the output of interest for different timeframes and with different sets of uncertain parameters. The option discussed in chapter 4.2 to modify parameters for each station individually should also be pursued to analyze the interaction between stations.

List of Figures

2.1	Overview of hydrological processes in LARSIM, modified from (11) . .	4
2.2	Influences of heat sources on snow cover, modified from (14)	7
3.1	Forward and Inverse Uncertainty Quantification, (18)	9
4.1	Interaction of classes within the prototype	14
4.2	Satellite image of the district Regen (35)	18
4.3	Temperatures March/April 2013 in Zwiesel (district Regen) (38)	18
5.1	Logical structure of the simulation prototype	21
5.2	Interaction of classes within the prototype	22
6.1	Total sensitivity of uncertain variables in station LOHB	28
6.2	Station LOHB: Simulated runoff (mean, percentiles, standard deviation) and measured runoff	29
6.3	Station LOHB: Simulated runoff, unaltered runoff and measured runoff	30
6.4	Station LOHB: April 15 th -27 th : Comparison of simulated runoff and measured runoff	31
6.5	LOHB: Relative error of simulated runoff and unaltered runoff to mea- sured runoff	31

List of Tables

3.1	Optimal polynomial choices based on random distribution (24)	10
4.1	Excerpt of <i>tape10</i> configuration options	16
4.2	Distributions and value ranges for uncertain parameters	17
4.3	Prototype configuration setup	19
5.1	Command-line arguments for the prototype	24
6.1	Approximation error comparison	32

Bibliography

- (1) B. L. für Umwelt. *Hochwassernachrichtendienst - Organisation*. Mar. 9, 2019. URL: <https://www.lfu.bayern.de/wasser/hochwassernachrichtendienst/organisation/index.htm>.
- (2) U. G. Survey. *Snowmelt - The water cycle*. Mar. 9, 2019. URL: <https://water.usgs.gov/edu/watercyclesnowmelt.html>.
- (3) M. Bremicker. *Das Wasserhaushaltsmodell LARSIM – Modellgrundlagen und Anwendungsbeispiele* –. 2000.
- (4) M. B. Karl Ludwig. *The Water Balance Model LARSIM - Design, Content and Applications*. 2006.
- (5) L. K. *The program system FGMOD for calculation of runoff processes in river basins*. 1982.
- (6) V. P. Singh. *Watershed Modeling. Computer Models of Watershed Hydrology*. 2012.
- (7) LARSIM-Entwicklergemeinschaft. *Das Wasserhaushaltsmodell LARSIM – Modellgrundlagen und Anwendungsbeispiele* –. 2018.
- (8) M. B. A. Luce I. Haag. *Einsatz von Wasserhaushaltsmodellen zur kontinuierlichen Abflussvorhersage in Baden-Württemberg*. 2006.
- (9) J. Mitterer. *Abschätzung der Unsicherheiten von Hochwasserabflüssen des Wasserhaushaltsmodells LARSIM auf Grund variabler Bodendaten*. 2015.
- (10) M. B. Annette Luce Ingo Haag. *Einsatz von Wasserhaushaltsmodellen zur kontinuierlichen AbflussVorhersage in Baden-Württemberg*. 2006.
- (11) M. B. Anette Luce Ingo Haag. *Einsatz von Wasserhaushaltsmodellen zur kontinuierlichen Abflussvorhersage in Baden-Württemberg*. 2006.
- (12) D. Knauf. *Die Berechnung des Abflusses aus einer Schneedecke*. 1980.
- (13) K. Ludwig. *Weiterentwicklung des Schneemodells und Vereinheitlichung der Energiebilanzansätze in LARSIM*. 2006.
- (14) J. M. Prof. Dr.-Ing. Markus Disse. *Flood Risk and Flood Management, Technical University Munich, Department of Civil, Geo and Environmental Engineering*. 2017.
- (15) F. A. Bertle. *Effect of snow compaction on runoff from rain on snow*. 1966.
- (16) T. Sullivan. *Introduction to Uncertainty Quantification*. 2015.
- (17) V. Committee on Mathematical Foundations of Verification and N. R. C. Uncertainty Quantification. *Assessing the Reliability of Complex Models*. 2012.

- (18) A. M. Stuart. *Uncertainty Quantification in Bayesian Inversion*. 2010.
- (19) G. Iaccarino. *Quantification of Uncertainty in Flow Simulations Using Probabilistic Methods*. 2008.
- (20) D. Xiu. *Fast Numerical Methods for Stochastic Computations: A Review*. 2008.
- (21) A. K. et al. *Review of Polynomial Chaos-Based Methods for Uncertainty Quantification in Modern Integrated Circuits*. 2018.
- (22) B. Lapeyre. *Introduction to Monte-Carlo Methods*. 2007.
- (23) N. Wiener. *The homogenous chaos*. 1938.
- (24) G. E. K. Dongbin Xiu. *The Wiener-Askey Polynomial Chaos for stochastic differential equations*. 2002.
- (25) R. C. Smith. *Uncertainty Quantification: Theory, Implementation and Application*. 2014.
- (26) D. X. Youssef Marzouk. *A Stochastic Collocation Approach to Bayesian Inference in Inverse Problems*. 2009.
- (27) D. Xiu. *Numerical Methods for Stochastic Computations*. 2010.
- (28) D. Xiu. *Efficient Collocational Approach for Parametric Uncertainty Analysis*. 2006.
- (29) A. B. Owen. *Variance Components and Generalized Sobol' Indices*. 2013.
- (30) A. S. Toshimitsu Homma. *Importance measures in global sensitivity analysis of nonlinear models*. 1996.
- (31) I. M. Sobol'. *Sensitivity Estimates for Nonlinear Mathematical Models*. 1993.
- (32) LARSIM-Entwicklergemeinschaft. *LARSIM-Hilfe Revision 192*. 2017.
- (33) I. H. et al. *Guideline for the Calibration of LARSIM Water Balance Models for operational Application in Flood Forecasting*. 2016.
- (34) T. N. Palmer. *Predicting uncertainty in forecasts of weather and climate*. 2000.
- (35) G. GEODIS Brno. *District Regen satellite map*. Mar. 5, 2019. URL: <https://www.google.com/maps/place/Regen/@49.0095783,13.0639571,54308m/data=!3m1!1e3!4m5!3m4!1s0x47753c90c2e85ccb:0x92ce5ca7f8f099f0!8m2!3d48.9771334!4d13.1283578!5m1!1e4>.
- (36) L. Regen. *Zahlen, Statistiken, Karten: Landkreis Regen*. Mar. 5, 2019. URL: <https://www.landkreis-regen.de/zahlen-statistiken-karten/>.
- (37) A. B. Wald. *Abenteuer Natur*. Mar. 5, 2019. URL: <https://www.arberland-bayerischer-wald.de/abenteuer-natur/158/5368>.
- (38) Wetterkontor.de. *Wetterrückblick Zwiesel (Bayerischer Wald, 612 m)*. Mar. 5, 2019. URL: <https://www.wetterkontor.de/de/wetter/deutschland/rueckblick.asp?id=217&datum=30.04.2013&t=8>.
- (39) D. Wetterdienst. *Pressemitteilung: Deutschlandwetter im März 2013*. 2013.
- (40) D. Wetterdienst. *Pressemitteilung: Deutschlandwetter im April 2013*. 2013.
- (41) F. D. F. Künzner. *FAST AND FLEXIBLE UNCERTAINTY QUANTIFICATION THROUGH A DATA-DRIVEN SURROGATE MODEL*. 2018.
- (42) J. Feinberg. *Chaospy documentation*. 2019.

Bibliography

- (43) H. P. L. Jonathan Feinberg. *Chaospy: An open source tool for designing methods of uncertainty quantification*. 2015.
- (44) J. D. H. et al. *Matplotlib 3.0.3 Documentary*. Mar. 8, 2019. URL: <https://matplotlib.org/>.
- (45) S. LLC. *Slurm Workload Manager Documentation*. Mar. 7, 2019. URL: <https://slurm.schedmd.com/overview.html>.
- (46) E. W. Weisstein. "Relative Error." *From MathWorld—A Wolfram Web Resource*. Mar. 11, 2019. URL: <http://mathworld.wolfram.com/RelativeError.html>.