

Bayesian Uncertainty Quantification of Physical Models in Thermal-Hydraulics System Codes

THÈSE N° 8426 (2018)

PRÉSENTÉE LE 23 FÉVRIER 2018

À LA FACULTÉ DES SCIENCES DE BASE

LABORATOIRE DE PHYSIQUE DES RÉACTEURS ET DE COMPORTEMENT DES SYSTÈMES

PROGRAMME DOCTORAL EN PHYSIQUE

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Damar Canggih WICAKSONO

acceptée sur proposition du jury:

Prof. R. Houdré, président du jury
Prof. A. Pautz, Mr O. Zerkak, directeurs de thèse
Dr J. Baccou, rapporteur
Dr W. Zwermann, rapporteur
Prof. B. Sudret, rapporteur



ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2018

We may be stupid, but we're not clever.

— Stephen Fry, *A bit of Fry and Laurie*, Special Squad Sketch

Dedicated to mediocrity and the strive for a bare minimum.

ACKNOWLEDGMENTS

This doctoral research has been conducted in the framework of the STARS project in the Laboratory for Reactor Physics and Systems Behavior (LRS)¹ at the Paul Scherrer Institut (PSI). Besides providing me with a comfortable working place, PSI through the STARS project also kindly financed all the trips taken during the course of the doctoral research, including NUTHOS-10 in Okinawa, NURETH-16 in Chicago, SAMO 2016 in Capri, NUTHOS-11 in Gyeongju, as well as several trips to the OECD/NEA office in Paris.

It is my cultural reflex to readily admit that any successful work is a collective effort, sometimes with God's blessing thrown into the mix; while any eventual shortage is to be credited to me and me alone. This is especially true for this work which at times – if not most of the times – felt really difficult to carry out. So I am truly grateful for the support and contributions of so many people:

- Prof. Andreas Pautz, the thesis director, for giving me the opportunity to conduct this research and to extend, several times, my stay in Switzerland. I appreciate his support all these years and wish him all the best navigating the Nuclear Energy and Safety Division of Paul Scherrer Institut (PSI) in an evermore turbulent times.
- Mr. Omar Zerkak and Dr. Gregory Perret who have invested in me so much of their time, knowledge, and patience that allowed me to start and finish the thesis. I am truly grateful for those, especially when the thesis was becoming harder and harder to finish.
- Mr. Philippe Jacquemoud, in his own word an IT “all-rounder”, who has been truly helpful providing me with IT-related supports, but perhaps much more importantly, by being a very good friend.
- Dr. Jean Baccou, Prof. Bruno Sudret, and Dr. Winfried Zwermann for their willingness to sit on my PhD committee, evaluating the thesis over the Christmas holiday period and attending the defense. It has been my privilege to have them in my committee and I am truly grateful for their invaluable comments, feedback, and questions.

¹ which by the time this thesis is printed ceases to exist. Perhaps it just metamorphosizes to a better form...

- Dr. Mathieu Hursin and Dr. Dimitri Rochman, who appeared to be genuinely interested in my work and gave invaluable comments during these period.
- Mr. Hakim Ferroukhi and Dr. Ivor Clifford who have always been appreciative and supportive towards my work in their units.
- Mrs. Ruth Ringele and Mrs. Andrea Mohr who have been truly kind and helpful dealing with the administrative matters during my PhD research.
- The present and past members of the LRS, especially the STARS Coffee Club members who have created for me a conducive – fun and mostly productive – working environment.
- Abhishek Saxena, Rita Szijártó, Yaroslav Sych, Olivier Leray, Gregory² and Rebecca Lordan-Perret, Carl Adamsson, Younsuk Yun, Raoul Ngayam-Happy, Riccardo Puragliesi, Kinga Bernatowicz-Gomà, Dionysios and Vera Chionis, Petra Mala³, Vladimir Brankov, and Matthew and Deborah Stark-Studer; friends, some are newer than the others, but good friends all the same. It is easy to forget that I am thousands of kilometers away from home; during this period, these people have been – in one way or another – my “immediate family”.
- Fauziah Satriana who provides me with a sense of being at home anywhere I am with her⁴. Her patience and understanding towards me during the thesis write-up period will remain a mystery I am not willing to rationalize.
- My family back home who all seem to be genuinely proud of what I have achieved⁵.

– WD41

² Déjà vu? Actually yes, and triple the thanks!

³ Things are not always easy, nor fair, but perhaps you could hold on for a little bit longer and get that piece of paper nobody seems to care about. Or so it seems...

⁴ which unfortunately not that often, yet...

⁵ But I know better...

ABSTRACT

Nuclear thermal-hydraulics (TH) system codes use several parameterized physical or empirical models to describe complex two-phase flow phenomena. The reliability of their predictions is as such primarily affected by the uncertainty associated with the parameters of the models. Because these model parameters often cannot be measured, nor have inherent physical meanings, their uncertainties are mostly based on expert judgment.

The present doctoral research aims to quantify the uncertainty of physical model parameters implemented in a TH system code based on experimental data. Specifically, this thesis develops a methodology to use experimental data to inform these uncertainties in a more objective manner. The methodology is based on a probabilistic framework and consists of three steps adapted from recent developments in applied statistics: global sensitivity analysis (GSA), metamodeling, and Bayesian calibration.

The methodology is applied to reflood experiments from the FEBA separate effect test facility (SETF), which are modeled with the TH system code TRACE. Reflood is chosen as a relevant phenomenon for the safety analysis of light water reactors (LWRs) and three typical time-dependent outputs are investigated: clad temperature, pressure drops and liquid carryover.

In the first step, GSA allows screening out input parameters that have a low impact on the reflood transient. Functional data analysis (FDA) is then used to reduce the dimensionality of the time-dependent code outputs, while preserving their interpretability. The resulting quantities can be used once more with GSA to investigate, quantitatively, the effect of the input parameters on the overall time-dependent outputs.

In the second step, a Gaussian process (GP) metamodel is developed and validated as a surrogate for the TRACE model. The average prediction error of the metamodel is sufficiently low to predict all considered outputs, and its computational cost is less than 5 [s] as compared to 6 – 15 [min] per TRACE run.

In the final step, the a posteriori model parameter uncertainties are quantified by calibration on a selected test from the FEBA experiments. Several posterior probability density functions (PDFs) corresponding to different calibration schemes – with and without model bias term and for different types of output – are formulated and directly sampled using a Markov Chain Monte Carlo (MCMC) ensemble sampler and the GP metamodel. The posterior samples are then

propagated in a set of FEBA experiments to check the validity of the posterior model parameter values and uncertainties.

The calibration is performed on different types of output to inform model parameters that would have otherwise remained non-identifiable. The calibration scheme with model bias term is able to constrain the prior uncertainties of the model parameters while keeping the nominal TRACE parameters values within the posterior uncertainty interval. That is in contrast with the results of the calibration without model bias term, in which the posterior uncertainties are concentrated on either side of the prior range, and at times do not include the nominal TRACE parameters values. Finally, except for a few outputs – the clad temperature output at the top assembly and the liquid carryover –, the relative performance of all posterior uncertainties is insensitive to boundary conditions of the different FEBA tests.

The proposed methodology was shown to successfully inform the uncertainty of the model parameters involved in a reflood transient. In the future, the methodology can be applied to model parameters involved in other TH phenomena using data from SETFs and, hopefully, contributes to achieve the goal of quantifying uncertainties for transients considered in the safety assessment of LWRs.

KEYWORDS: system thermal-hydraulics (TH), reflood, TRAC/RELAP Computational Engine (TRACE) code, uncertainty quantification (UQ), global sensitivity analysis (GSA), Gaussian process (GP) metamodel, Bayesian calibration

RÉSUMÉ

Les codes de système thermohydraulique nucléaires utilisent plusieurs modèles paramétriques physiques ou empiriques pour modéliser des écoulements diphasiques complexes. La précision de leurs prédictions est de fait directement affectée par les incertitudes des paramètres de ces modèles. Du fait que ces paramètres ne sont souvent ni mesurables ni n'ont de significations physique propres, leurs incertitudes sont généralement déterminés par un jugement d'expert.

Ce travail de thèse a pour but de quantifier les incertitudes des paramètres des modèles physiques implémentés dans les codes de système thermohydraulique en utilisant des données expérimentales. Cette thèse développe plus spécifiquement une méthodologie qui utilise les données expérimentales pour quantifier ces incertitudes de manière plus objective. La méthodologie utilise une approche probabiliste et comprend trois étapes qui proviennent de développements récents dans le domaine des méthodes statistiques appliquées : analyse de sensibilité globale (GSA), méta-modèle, et calibration Bayésienne.

La méthode est appliquée dans le cadre d'expériences de renoyage qui se sont déroulés dans l'installation FEBA et qui sont modélisées avec le code de thermohydraulique TRACE. Le renoyage est choisi car il représente un phénomène d'importance majeure dans le cadre des analyses de sûreté des réacteurs à eau légère (LWR). Trois types de sortie du code qui dépendent du temps sont observés : la température de la gaine, la réduction de pression et la quantité de liquide entraîné hors de la section de test.

Dans la première étape de la méthodologie, l'analyse de sensibilité globale permet d'éliminer des paramètres d'entrées du code qui ont une faible influence sur le transitoire de renoyage. L'analyse de fonctions (functional data analysis (FDA)) permet de réduire le nombre de dimensions des sorties du code dépendant du temps tout en préservant leurs interprétabilités. Ceci permet, à l'aide d'une nouvelle analyse de sensibilité, de quantifier les effets des paramètres d'entrées sur les paramètres de sorties du code considérés dans leur ensemble.

Dans la seconde étape, un méta-modèle basé sur un processus gaussien (GP) est développé et validé comme substitut au modèle TRACE. Les incertitudes sur les prédictions du méta-modèle sont suffisamment faibles pour prédire précisément toute les sorties d'intérêt. Le méta-modèle est évalué en moins de 5 [s] contre 6 – 15 [min] pour le modèle TRACE.

Dans la dernière étape, l'incertitude a posteriori sur les paramètres des modèles est quantifiée par calibration sur une expérience choisie parmi l'ensemble des expériences FEBA considérés dans cette thèse.

Plusieurs densités de probabilités a posteriori correspondant à différents schémas de calibration (avec et sans terme prenant en compte le biais du modèle et pour différents types de sortie du code) sont formulées et directement échantillonnées en utilisant le méta-modèle gaussien et un échantillonneur d'ensemble basé sur la méthode de Monte-Carlo par chaînes de Markov (MCMC). Les échantillons obtenus sont propagés dans l'ensemble des expériences FEBA considérés pour vérifier la validité des valeurs et incertitudes des paramètres des modèles obtenus par calibration.

En utilisant différents types de sorties du code la calibration a permis d'améliorer les incertitudes de certains paramètres qui seraient dans le cas contraire restés à leurs valeurs d'origine. La calibration qui prend en compte le biais du modèle a quant à elle permis de contraindre les incertitudes a priori des paramètres tout en garantissant que leurs valeurs nominales restent dans l'intervalle de confiance a posteriori. Ce n'est pas le cas pour la calibration qui ne prend pas en compte le biais du modèle. Pour cette dernière, les incertitudes a posteriori sont concentrées sur les bords de l'intervalle de confiance a priori des paramètres et parfois n'incluent pas leurs valeurs nominales. Finalement, excepté pour la température de la gaine au sommet de l'assemblage et la quantité de liquide transporté hors du système, les performances de toutes les incertitudes a posteriori obtenues ne sont pas sensibles aux conditions limites des différentes expériences FEBA considérées.

La méthodologie proposée dans cette thèse a permis de réduire les incertitudes des paramètres des modèles utilisés dans la modélisation du transitoire de renoyage. Dans le future, cette méthodologie pourra être mise en œuvre avec des modèles impliqués dans d'autres phénomènes thermohydrauliques en utilisant des données issues d'autres installations pour l'étude d'effet thermohydraulique (SETF), et pourquoi pas ainsi contribué à atteindre le but de quantifier les incertitudes dans les transitoires considérés dans l'analyse de sûreté des réacteurs à eau légère.

MOTS-CLEFS: système thermohydraulique, reflood, code TRAC/RELAP Computational Engine (TRACE), quantification d'Incertitude (UQ), analyse de sensibilité globale (GSA), méta-modèle processus gaussien (GP), calibration Bayésienne

INTISARI

Kode thermo-hidrolika sistem tenaga nuklir menggunakan beberapa model parametrik, baik empiris maupun mekanistik, untuk menggambarkan fenomena-fenomena aliran dua fase yang kompleks. Keadanan prediksi kode thermo-hidrolika sistem dipengaruhi oleh ketidakpastian yang berhubungan dengan parameter-parameter di dalam model-model tersebut. Karena parameter-parameter tersebut seringkali tidak bisa diukur secara langsung, dan bahkan tidak memiliki arti fisik yang melekat, ketidakpastian yang berhubungan dengan parameter-parameter tersebut biasanya ditentukan dengan pertimbangan ahli.

Tujuan dari riset doktoral ini adalah untuk melakukan kuantifikasi ketidakpastian dari parameter-parameter yang diimplementasikan di dalam kode thermo-hidrolika sistem berdasarkan data dari eksperimen. Khususnya, disertasi ini mengembangkan sebuah metodologi untuk memanfaatkan data dari eksperimen guna memperbarui ketidakpastian tersebut dengan cara yang lebih objektif. Metodologi yang diajukan ini dikembangkan berdasarkan kerangka kerja probabilistik dan terdiri dari tiga langkah yang diadaptasi dari perkembangan terkini dalam statistika terapan: analisis sensitivitas global (*global sensitivity analysis*, GSA), pemetamodelan, dan kalibrasi Bayes.

Metodologi tersebut kemudian diterapkan pada eksperimen *reflood* di fasilitas uji efek terpisah FEBA, yang dimodelkan dengan kode thermo-hidrolika sistem TRACE. *Reflood* dipilih sebagai fenomena yang relevan dalam analisis keselamatan reaktor air ringan. Investigasi dilakukan terhadap tiga keluaran utama gayut-waktu: temperatur cladding, penurunan tekanan, dan *carryover* cairan.

Di langkah yang pertama, analisis sensitivitas global mampu menyaring parameter-parameter yang kurang berpengaruh terhadap keluaran simulasi *reflood*. Kemudian, analisis data fungsi (*functional data analysis*, FDA) digunakan untuk mereduksi dimensi keluaran gayut-waktu, sembari mempertahankan penafsiran keluaran tersebut. Besaran-besaran yang dihasilkan dapat digunakan dengan analisis sensitivitas global untuk menginvestigasi, secara kuantitatif, efek parameter masukan terhadap keluaran gayut-waktu secara menyeluruh.

Di langkah yang kedua, sebuah metamodel berdasarkan proses Gauss (*Gaussian process*, GP) dikembangkan dan divalidasi untuk digunakan sebagai pengganti model TRACE. Kesalahan prediksi rerata metamodel tersebut cukup rendah untuk memprediksi secara akurat semua keluaran-keluaran yang disebut di atas. Terlebih lagi, biaya komputasi evaluasi dengan metamodel membutuhkan kurang dari 5 detik untuk tiap evaluasi, dibandingkan dengan waktu yang dibutuhkan TRACE untuk tiap evaluasi antara 6 sampai 15 menit.

Di langkah yang terakhir, ketidakpastian dari parameter-parameter model dikuantifikasi secara a posteriori melalui kalibrasi berdasarkan data dari uji terpilih FEBA. Beberapa fungsi densitas peluang (*probability density function*, PDF) posterior yang terkait dengan beberapa skema kalibrasi – baik dengan mempertimbangkan suku ketidaksesuaian model (*model bias term*) maupun tidak, dan dengan mempertimbangkan berbagai macam tipe keluaran – diformulasikan. Dari formulasi tersebut, sampel langsung diambil secara acak menggunakan algoritma Monte Carlo Rantai Markov (*Markov Chain Monte Carlo*, MCMC) ansambel dan metamodel proses Gauss; dan kemudian dipropagasikan untuk beberapa uji FEBA guna memastikan validitas nilai dan ketidakpastian dari parameter-parameter model tersebut.

Kalibrasi dilakukan terhadap beberapa tipe keluaran untuk memperbarui ketidakpastian dari parameter-parameter model. Jika keluaran-keluaran tersebut tidak dipertimbangkan, maka ketidakpastian dari beberapa parameter-parameter model tidak dapat diperbarui. Skema kalibrasi dengan suku ketidaksesuaian model mampu membatasi ketidakpastian awal dari parameter-parameter tersebut, sembari mempertahankan nilai nominal parameter-parameter TRACE di dalam rentang ketidakpastian akhir. Hasil ini berlawanan dengan hasil dari kalibrasi tanpa suku ketidaksesuaian tersebut, sedemikian hingga ketidakpastian akhir terpusatkan di salah satu sisi rentang ketidakpastian awal, dan kadang tidak mengikutsertakan nilai nominal parameter-parameter TRACE. Kecuali untuk beberapa keluaran – temperatur cladding di bagian atas rangkaian fasilitas uji dan *carryover* cairan –, kinerja relatif dari ketidakpastian akhir tidak dipengaruhi oleh syarat batas dari beberapa uji FEBA.

Metodologi yang diajukan di atas berhasil memperbarui ketidakpastian dari parameter-parameter model yang berhubungan dengan simulasi *reflood*. Pada masa yang akan datang, metodologi ini dapat diterapkan untuk parameter-parameter model yang berhubungan dengan simulasi fenomena-fenomena thermo-hidrolika lainnya menggunakan data dari berbagai fasilitas uji efek terpisah. Metodologi ini juga diharapkan dapat memberikan kontribusi dalam melakukan kuantifikasi ketidakpastian secara menyeluruh dalam penilaian keselamatan reaktor air ringan.

KATA KUNCI: thermo-hidrolika sistem, reflood, kode TRAC/RELAP Computational Engine (TRACE), kuantifikasi ketidakpastian (UQ), analisis sensitivitas (GSA), metamodel proses Gauss (GP), Kalibrasi Bayes

CONTENTS

Acknowledgments	iii
Abstract	v
Résumé	vii
Intisari	ix
Contents	xi
List of Figures	xvi
List of Tables	xxvii
List of Algorithms	xxx
1 INTRODUCTION	1
1.1 Computer Simulation and Safety Analysis of Nuclear Power Plant	2
1.1.1 Scientific Computer Simulation	2
1.1.2 Codes and Safety Analysis of Nuclear Power Plant	3
1.1.3 Thermal-Hydraulics (TH) System Codes	4
1.2 Uncertainty Quantification in Nuclear Engineering Thermal-Hydraulics	7
1.2.1 Forward Uncertainty Quantification	8
1.2.2 Inverse (Backward) Uncertainty Quantification	10
1.2.3 OECD/NEA PREMIUM project	12
1.3 Objectives and Scope of the Thesis	15
1.3.1 Statement of the Problem	15
1.3.2 Objectives	16
1.3.3 Scope	18
1.4 Statistical Framework	20
1.4.1 Sensitivity Analysis	20
1.4.2 Statistical Metamodeling	23
1.4.3 Bayesian Calibration	26
1.5 Structure of the Thesis	28
2 REFLOOD SIMULATION USING THE TRACE CODE	31
2.1 The Thermal-Hydraulics System Code TRACE	32
2.2 Phenomenology and Modeling of Bottom Reflood	36
2.3 FEBA Reflood Separate Effect Test Facility	40
2.4 FEBA Model in TRACE	42
2.5 Initial Selection of Input Parameters	43
2.5.1 Selection of Input Parameters	43
2.5.2 Perturbation Factors	47
2.5.3 Prior Uncertainty Quantification	49
2.6 Propagation of the Prior Uncertainties	50
2.7 Chapter Summary	53
3 SENSITIVITY ANALYSIS	55

3.1	Statistical Framework	55
3.2	Describing Variation of Time-Dependent Output	56
3.2.1	Functional Output Representation	57
3.2.2	Curve Registration by Landmarks	59
3.2.3	Functional Principal Component Analysis	60
3.3	Parameters Screening	62
3.3.1	Elementary Effects and One-at-a-Time Design	62
3.3.2	Statistics of Elementary Effects and Sensitivity Measures	64
3.4	Variance Decomposition	65
3.4.1	High-Dimensional Model Representation	66
3.4.2	Sobol' Sensitivity Indices	67
3.5	Implementation	68
3.5.1	The Morris Method	69
3.5.2	The Sobol'-Saltelli Method	69
3.6	Application to TRACE Model of FEBA	72
3.6.1	Simulation Experiment	72
3.6.2	Screening Analysis	75
3.6.3	Sobol' Indices for Conventional QoIs of the Reflood Curve	78
3.6.4	Principal Components of the Reflood Curve	81
3.6.5	Sobol' Indices for QoIs based on Principal Components	84
3.6.6	Discussion	87
3.7	Chapter Summary	91
4	GAUSSIAN PROCESS METAMODELING	93
4.1	Statistical Framework	94
4.2	Gaussian Process Fundamentals	95
4.2.1	From Multivariate Gaussian to Gaussian Process	95
4.2.2	Gaussian Process	98
4.2.3	Covariance Kernel Function	101
4.2.4	Multidimensional Construction	106
4.2.5	Process Variance	108
4.2.6	Mean Function	108
4.3	Gaussian Process Metamodel	109
4.4	Practical Aspects of GP Metamodel Constructions	113
4.4.1	Selection of Design/Training Points	113
4.4.2	Model Fitting/Training	116
4.4.3	Model Validation and Selection	118
4.5	Dealing with Multivariate Output	120
4.5.1	Linear Model of Coregionalization (LMC)	121
4.5.2	Principal Component Analysis	122
4.5.3	Multivariate Gaussian Process Metamodel	124
4.6	Application to the TRACE model of FEBA	126
4.6.1	Simulation Experiment	126

4.6.2	Dimension Reduction by principal component analysis (PCA)	128
4.6.3	GP PC Metamodel Construction	131
4.6.4	GP PC Metamodel Testing	134
4.6.5	Discussion	136
4.7	Chapter Summary	139
5	BAYESIAN CALIBRATION	141
5.1	Statistical Framework	142
5.2	Bayesian Formulation of Calibration Problem	144
5.2.1	Probabilistic Model for the Model Bias Term	144
5.2.2	Probabilistic Model for the Observation error	149
5.2.3	Probabilistic Model for the Simulator	150
5.2.4	Posterior of the Model Parameters	150
5.2.5	Modularization of the Bayesian Framework	152
5.3	MCMC Simulation	155
5.3.1	Motivation	156
5.3.2	Markov Chain	160
5.3.3	Markov Chain Monte Carlo	163
5.3.4	Affine-Invariant Ensemble Sampler (AIES)	168
5.4	Diagnosing MCMC Samples	174
5.4.1	Autocorrelation in Equilibrium and Thinning	176
5.4.2	Initialization Bias and Burn-in	179
5.5	Application to the TRACE Model of FEBA	180
5.5.1	Simulation Experiment	181
5.5.2	MCMC Convergence	194
5.5.3	Calibration Results	197
5.5.4	Calibration Evaluation	201
5.5.5	Discussion	207
5.6	Chapter Summary	215
6	CONCLUSIONS AND FUTURE WORK	217
6.1	Chapter-wise Summary	217
6.2	Achievements and Recommendations	220
6.2.1	Contributions to OECD/NEA PREMIUM Project	221
6.2.2	Implementation and application of GSA methods	222
6.2.3	Development and validation of a TRACE meta-model	223
6.2.4	Bayesian calibration of the TRACE reflood model parameters against various relevant experimental data	224
A	TRACE CODE GOVERNING EQUATIONS	227
A.1	Mass Balance Equations	227
A.2	Momentum Balance Equations	228
A.3	Energy Balance Equations	229
A.4	Heat Conduction Equations	231
A.5	Closure and Flow Regimes	231

B	ADDITIONAL RESULTS	233
B.1	Prior Uncertainty Propagation of the FEBA Tests	233
B.1.1	Clad Temperature Output (TC)	233
B.1.2	Pressure Drop Output (DP)	239
B.1.3	Liquid carryover Output (CO)	241
B.2	Screening Analysis (27-parameter Model)	243
B.3	Convergence of the Sobol' Indices	256
B.4	Sobol Indices (12-parameter Model)	258
B.5	Gaussian Process Metamodel Construction	262
B.6	MCMC Samples from different Calibration Schemes	264
B.7	Forward Uncertainty Propagation of MCMC samples	269
B.7.1	FEBA Test No. 216, clad Temperature Output (TC)	269
B.7.2	FEBA Test No. 214, clad Temperature Output (TC)	272
B.7.3	FEBA Test No. 223, clad Temperature Output (TC)	275
B.7.4	FEBA Test No. 218, clad Temperature Output (TC)	278
B.7.5	FEBA Test No. 220, clad Temperature Output (TC)	281
B.7.6	FEBA Test No. 222, clad Temperature Output (TC)	284
B.7.7	FEBA Test No. 216, Pressure Drop Output (DP)	287
B.7.8	FEBA Test No. 214, Pressure Drop Output (DP)	288
B.7.9	FEBA Test No. 223, Pressure Drop Output (DP)	289
B.7.10	FEBA Test No. 218, Pressure Drop Output (DP)	290
B.7.11	FEBA Test No. 220, Pressure Drop Output (DP)	291
B.7.12	FEBA Test No. 222, Pressure Drop Output (DP)	292
B.7.13	FEBA Test No. 216, Liquid Carryover Output (CO)	293
B.7.14	FEBA Test No. 214, Liquid Carryover Output (CO)	293
B.7.15	FEBA Test No. 223, Liquid Carryover Output (CO)	294
B.7.16	FEBA Test No. 218, Liquid Carryover Output (CO)	294
B.7.17	FEBA Test No. 220, Liquid Carryover Output (CO)	295
B.7.18	FEBA Test No. 222, Liquid Carryover Output (CO)	295
C	COMPUTATIONAL TOOLS	297
C.1	gsa-module	297
C.2	trace-simexp	299
D	SOME USEFUL MATHEMATICAL RESULTS AND RECIPES	301

D.1	The Sobol'-Saltelli Method for Estimating Variance-Based Sensitivity Indices	301
D.2	Multivariate Random Variable (Random Vector)	302
D.3	Gaussian Random Vector (Multivariate Normal Random Variable)	304
D.4	Inverse Transform Sampling	306
D.5	Generating Samples from a Multivariate Normal Distribution	307
D.6	Landmark Registration and Time Warping Function	308
D.7	Karhunen-Loève Theorem	309
D.8	Discrete-State Markov Chain	310
BIBLIOGRAPHY		315
Acronyms and Abbreviations		341
Curriculum Vitae		345

LIST OF FIGURES

Figure 1.1	Nodalization of a nuclear power plant (NPP) in a thermal-hydraulics (TH) system code	5
Figure 1.2	Generic structure of a thermal-hydraulics (TH) system code	6
Figure 1.3	Simplified illustration of a simulator as an input/output model.	7
Figure 1.4	Simplified flowchart of forward uncertainty quantification of a simulator prediction.	10
Figure 1.5	Simplified flowchart of inverse uncertainty quantification of model parameters.	11
Figure 1.6	The structure of thesis.	29
Figure 2.1	Illustration of Heated Channel.	33
Figure 2.2	Some of the observed flow regimes in vertical and horizontal flow	34
Figure 2.3	A hypothetical phase indicator probe inside a channel of a two-phase flow	34
Figure 2.4	Illustration of Flow Averaging	36
Figure 2.5	A typical clad temperature evolution during constant flooding rate reflooding at mid-height assembly.	37
Figure 2.6	Phenomenology of two-phase flow during re-flood according to the TRACE code.	38
Figure 2.7	FEBA experimental facility.	40
Figure 2.8	Nodalization of the FEBA experimental facility in TRACE.	45
Figure 2.9	Nominal TRACE predictions for Flooding Experiments with Blocked Arrays (FEBA) test No. 216 in comparison with the experimental data for three selected outputs.	50
Figure 2.11	Propagation of the 27 input parameters prior uncertainties on FEBA test No. 216 for the pressure drop output (DP).	51
Figure 2.10	Propagation of the 27 input parameters prior uncertainties on FEBA test No. 216 for the clad temperature output (TC).	52
Figure 2.12	Propagation of the 27 input parameters prior uncertainties on FEBA test No. 216 for the liquid carryover output (CO).	53
Figure 3.1	Spline basis functions of order 4	58
Figure 3.2	Variation in functional data set, with and without phase variation.	59

Figure 3.3	Illustration of curve registration.	60
Figure 3.4	Illustration of One-at-a-Time (OAT) design using trajectory and radial schemes.	63
Figure 3.5	Illustration of a typical parameter importance classification based on Morris screening method	66
Figure 3.6	Flowchart for the implemented sensitivity analysis methodology applied to the TRACE model of the FEBA facility	74
Figure 3.7	Trace plots of screening sensitivity measures estimations.	75
Figure 3.8	The effect of influential vs. noninfluential parameters perturbations on different output.	77
Figure 3.9	Sobol' Indices estimates with the maximum mid-height clad temperature as the quantity of interest (QoI).	78
Figure 3.10	Sobol' Indices estimates with the time of quenching at mid-height of the assembly as the QoI	79
Figure 3.11	Evolution of the main-effect indices with the clad temperature at each time step as QoI.	80
Figure 3.12	The proportion of explained variance for each functional principal component (fPC) extracted from selected time-dependent outputs.	81
Figure 3.13	The 1 st fPC of the (registered) mid-height clad temperature transient and the effect of its perturbation on the mean function.	82
Figure 3.14	The 2 nd fPC of the (registered) mid-height clad temperature transient and the effect of its perturbation on the mean function.	82
Figure 3.15	The 1 st fPC of the warping function of the clad temperature transient at the mid-height of the assembly and the effect of its perturbation on the mean function.	83
Figure 3.16	The 1 st fPC of the pressure drop transient at the middle of the assembly and the effect of its perturbation on the mean function.	83
Figure 3.17	The 1 st fPC of the liquid carryover transient and the effect of its perturbation on the mean function.	84
Figure 3.18	Sobol' indices estimates with the 1 st fPC scores of the (registered) mid-height clad temperature transient as the QoI	85
Figure 3.19	Sobol' indices estimates with the 2 nd fPC scores of the (registered) mid-height clad temperature transient as the QoI	85

Figure 3.20	Sobol' indices estimates with the 1 st fPC scores of the warping function for the mid-height clad temperature transient as the QoI	86
Figure 3.21	Sobol' indices estimates with the 1 st fPC of the pressure drop transient at the middle of the assembly as the QoI	86
Figure 3.22	Sobol' indices estimates with the 1 st fPC of the liquid carryover transient as the QoI.	87
Figure 4.1	Illustration of Bayesian perspective of regression and metamodeling.	95
Figure 4.2	Illustration of a bivariate Gaussian distribution	96
Figure 4.3	From multivariate Normal to Gaussian process	97
Figure 4.4	Realizations of a Gaussian process, conditional and unconditional	100
Figure 4.5	Gaussian correlation kernels with three different range parameters.	103
Figure 4.6	Realizations of random function drawn from GP with Gaussian kernel	103
Figure 4.7	Examples of power exponential kernel functions	104
Figure 4.8	Examples of power exponential kernel functions	104
Figure 4.9	Examples of Matérn kernel functions	106
Figure 4.10	Sample paths drawn from GPs with Matérn kernel functions	106
Figure 4.11	Random surface realizations	107
Figure 4.12	Effect of different process variance values on GP realization	108
Figure 4.13	Effect of different mean functions on GP realization having the same covariance kernel	109
Figure 4.14	Grid approach to select training points	114
Figure 4.15	Examples of experimental design for metamodel training	115
Figure 4.16	Principal Component Analysis of a bivariate data set.	123
Figure 4.17	Flowchart of the simulation experiment for constructing a GP principal component (PC) metamodel of the TRACE model of the FEBA facility	126
Figure 4.18	Examples of multivariate clad temperature output in at eight different locations as function of time, presented as "images"	129
Figure 4.19	PCA results for the clad temperature output	130

- Figure 4.20 Convergence of the reconstruction error as function of the number of PC used in the reconstruction of the output space for three different output types. 130
- Figure 4.21 Convergence of PC metamodel with increasing number of training samples with respect to the standardized PCs scores associated with the clad temperature output 131
- Figure 4.22 The effect of training sample size, experimental design, and covariance function on the predictive performance of GP PC metamodel with respect to the clad temperature output 132
- Figure 4.23 Convergence of PC metamodel with increasing number of training samples with respect to the standardized PCs scores associated with the pressure drop output 133
- Figure 4.24 Convergence of GP metamodel with increasing number of training samples with respect to the standardized PCs scores associated with the liquid carryover output 134
- Figure 4.25 Convergence of the predictive performance of the metamodel with respect to the standardized PCs scores for each output type 135
- Figure 4.26 Errors, predicted and observed, due to the dimension reduction procedure and the functional approximation for the three types of output. 136
- Figure 4.27 GP PC metamodel is a global statistical metamodel which gives global accurate prediction *on average* 138
- Figure 5.1 Relationships between elements of the calibration formulation. 143
- Figure 5.2 Illustration of predictions made by computer simulator with and without bias, both with a single uncertain model parameter and a single controllable input x_c . 146
- Figure 5.3 Joint and marginal densities plots for the unnormalized PDF in the example. 158
- Figure 5.4 Sampling from a multivariate density by discretizing the state space in grids. 159
- Figure 5.5 Illustration of iterations in Markov Chain simulation by random walk Metropolis-Hastings algorithm. 165
- Figure 5.6 Illustration of a Markov chain simulation to generate samples from the target density given in the example. 166

- Figure 5.7 Results of samples generated by a Markov chain simulation for the target density given in the example. 167
- Figure 5.8 Convergence issue due to an over-dispersed proposal distribution. 167
- Figure 5.9 Convergence issue due to an under-dispersed proposal distribution. 168
- Figure 5.10 Illustration of a *stretch move* update for a single walker in a 2-dimensional state space. 171
- Figure 5.11 Trace plots of individual walkers and the running mean and standard deviation for x_1 . 172
- Figure 5.12 Results of samples generated by affine-invariant ensemble sampler (AIES) for the target density given in the example. 173
- Figure 5.13 Illustration of autocorrelation functions for three different Markov chains. 175
- Figure 5.14 Information modeling to represent uncertainty propagation results for a QoI Y . 192
- Figure 5.15 Informativeness of two different information sources. 193
- Figure 5.16 Calibration scores of three different sources with the same observed data y_{obs} . 193
- Figure 5.17 Ensemble trace plots for each model parameter of calibration with model bias term. 195
- Figure 5.18 Ensemble average and standard deviation as function of the number of iterations for calibration with model bias term. 196
- Figure 5.19 Univariate and bivariate marginals of the posterior samples for each of the 8 model parameters. Calibration with model bias term. 199
- Figure 5.20 Propagation of the model parameters uncertainty on FEBA test No. 216 for the clad temperature output (TC). The posterior samples are from the calibration scheme w/ Bias, All. 202
- Figure 5.21 Propagation of the model parameters uncertainty on FEBA test No. 216 for the pressure drop output (DP). The posterior samples are from the calibration scheme w/ Bias, All. 203
- Figure 5.22 Propagation of the model parameters uncertainty on FEBA test No. 216 for the liquid carryover output (CO). The posterior samples are from the calibration scheme w/ Bias, All. 204
- Figure 5.23 Calibration score vs. Informativeness for different posterior samples propagated on all the FEBA tests. 206

- Figure 5.24 Uncertainty propagation results for TC1 output (the clad temperature at the top of the assembly) of FEBA test No. 216 with the posterior of the model parameters from 3 different calibration schemes. 210
- Figure 5.25 Uncertainty propagation results for CO output (the clad temperature at the top of the assembly) of FEBA test No. 216 with the posterior of the model parameters from 3 different calibration schemes. 211
- Figure 5.26 Uncertainty propagation results for TC1 output (the clad temperature at the top of the assembly) of FEBA tests No. 216, 220, and 222 with the posterior uncertainties of the model parameters from the calibration scheme w/ Bias, All. 212
- Figure 5.27 Uncertainty propagation results for TC1 output (the clad temperature at the top of the assembly) of FEBA tests No. 216, 220, and 222 with the posterior uncertainties of the model parameters from the calibration scheme w/o Bias. 213
- Figure 5.28 Uncertainty propagation results for TC1 output (the clad temperature at the top of the assembly) of FEBA tests No. 214, 218, and 223 with the posterior uncertainties of the model parameters from the calibration scheme w/ Bias, All. 214
- Figure 5.29 Uncertainty propagation results for TC1 output (the clad temperature at the top of the assembly) of FEBA tests No. 214, 218, and 223 with the posterior uncertainties of the model parameters from the calibration scheme w/o Bias. 214
- Figure B.1 Propagation of the 27 input parameters prior uncertainties on FEBA test no. 214 for the clad temperature output (TC). 233
- Figure B.2 Propagation of the 27 input parameters prior uncertainties on FEBA test no. 216 for the clad temperature output (TC). 234
- Figure B.3 Propagation of the 27 input parameters prior uncertainties on FEBA test no. 223 for the clad temperature output (TC). 235
- Figure B.4 Propagation of the 27 input parameters prior uncertainties on FEBA test no. 218 for the clad temperature output (TC). 236

- Figure B.5 Propagation of the 27 input parameters prior uncertainties on FEBA test no. 220 for the clad temperature output (TC). 237
- Figure B.6 Propagation of the 27 input parameters prior uncertainties on FEBA test no. 222 for the clad temperature output (TC). 238
- Figure B.7 Propagation of the 27 input parameters prior uncertainties on FEBA test no. 214 for the pressure drop output (DP). 239
- Figure B.8 Propagation of the 27 input parameters prior uncertainties on FEBA test no. 216 for the pressure drop output (DP). 239
- Figure B.9 Propagation of the 27 input parameters prior uncertainties on FEBA test no. 223 for the pressure drop output (DP). 239
- Figure B.10 Propagation of the 27 input parameters prior uncertainties on FEBA test no. 218 for the pressure drop output (DP). 240
- Figure B.11 Propagation of the 27 input parameters prior uncertainties on FEBA test no. 220 for the pressure drop output (DP). 240
- Figure B.12 Propagation of the 27 input parameters prior uncertainties on FEBA test no. 222 for the pressure drop output (DP). 240
- Figure B.13 Propagation of the 27 input parameters prior uncertainties on FEBA test nos. 214 & 218 for the liquid carryover output (CO). 241
- Figure B.14 Propagation of the 27 input parameters prior uncertainties on FEBA test nos. 223 & 218 for the liquid carryover output (CO). 241
- Figure B.15 Propagation of the 27 input parameters prior uncertainties on FEBA test nos. 220 & 222 for the liquid carryover output (CO). 242
- Figure B.16 Trace plot of the main-effect sensitivity indices estimations. 256
- Figure B.17 Convergence of the Sobol' main effect indices estimators w.r.t maximum clad temperature. 257
- Figure B.18 Convergence of the Sobol' main effect indices estimators w.r.t first principal component. 257
- Figure B.19 The effect of training sample size, experimental design, and covariance function on the predictive performance of GP PC metamodel with respect to the pressure drop output 262

Figure B.20	The effect of training sample size, experimental design, and covariance function on the predictive performance of GP PC metamodel with respect to the liquid carryover output	263
Figure B.21	Univariate and bivariate marginals of the posterior samples for each of the 8 model parameters. Calibration with respect to the clad temperature output (TC) and with model bias term.	264
Figure B.22	Univariate and bivariate marginals of the posterior samples for each of the 8 model parameters. Calibration with respect to the pressure drop output (DP) and with model bias term.	265
Figure B.23	Univariate and bivariate marginals of the posterior samples for each of the 8 model parameters. Calibration with respect to the liquid carryover output (CO) and with model bias term.	266
Figure B.24	Univariate and bivariate marginals of the posterior samples for each of the 7 model parameters, excluding $df_{fbVIHTC}$ parameter. Calibration with respect to all types of output (TC, DP, and CO) and without model bias term.	267
Figure B.25	Univariate and bivariate marginals of the posterior samples for each of the 8 model parameters. Calibration with respect to all types of output (TC, DP, and CO) and without model bias term.	268
Figure B.26	Propagation of the model parameters uncertainty on FEBA test No. 216 for the clad temperature output (TC). The posterior samples are from the calibration scheme w/ Bias, All.	269
Figure B.27	Propagation of the model parameters uncertainty on FEBA test No. 216 for the clad temperature output (TC). The posterior samples are from the calibration scheme w/ Bias, no df_{fbVIHT} .	270
Figure B.28	Posterior uncertainty propagation of FEBA test No. 216 for the clad temperature output (TC). The posterior samples are from the calibration scheme w/o Bias.	271
Figure B.29	Propagation of the model parameters uncertainty on FEBA test No. 214 for the clad temperature output (TC). The posterior samples are from the calibration scheme w/ Bias, All.	272

Figure B.30	Propagation of the model parameters uncertainty on FEBA test No. 214 for the clad temperature output (TC). The posterior samples are from the calibration scheme w/ Bias, no dffbVIHT. 273	
Figure B.31	Posterior uncertainty propagation of FEBA test No. 214 for the clad temperature output (TC). The posterior samples are from the calibration scheme w/o Bias. 274	
Figure B.32	Propagation of the model parameters uncertainty on FEBA test No. 223 for the clad temperature output (TC). The posterior samples are from the calibration scheme w/ Bias, All. 275	
Figure B.33	Propagation of the model parameters uncertainty on FEBA test No. 223 for the clad temperature output (TC). The posterior samples are from the calibration scheme w/ Bias, no dffbVIHT. 276	
Figure B.34	Propagation of the model parameters uncertainty on FEBA test No. 223 for the clad temperature output (TC). The posterior samples are from the calibration scheme w/o Bias. 277	
Figure B.35	Propagation of the model parameters uncertainty on FEBA test No. 218 for the clad temperature output (TC). The posterior samples are from the calibration scheme w/ Bias, All. 278	
Figure B.36	Propagation of the model parameters uncertainty on FEBA test No. 218 for the clad temperature output (TC). The posterior samples are from the calibration scheme w/ Bias, no dffbVIHT. 279	
Figure B.37	Propagation of the model parameters uncertainty on FEBA test No. 218 for the clad temperature output (TC). The posterior samples are from the calibration scheme w/o Bias. 280	
Figure B.38	Propagation of the model parameters uncertainty on FEBA test No. 220 for the clad temperature output (TC). The posterior samples are from the calibration scheme w/ Bias, All. 281	
Figure B.39	Propagation of the model parameters uncertainty on FEBA test No. 220 for the clad temperature output (TC). The posterior samples are from the calibration scheme w/ Bias, no dffbVIHT. 282	

Figure B.40	Propagation of the model parameters uncertainty on FEBA test No. 220 for the clad temperature output (TC). The posterior samples are from the calibration scheme w/o Bias.	283
Figure B.41	Propagation of the model parameters uncertainty on FEBA test No. 222 for the clad temperature output (TC). The posterior samples are from the calibration scheme w/ Bias, All.	284
Figure B.42	Propagation of the model parameters uncertainty on FEBA test No. 222 for the clad temperature output (TC). The posterior samples are from the calibration scheme w/ Bias, no dffbVIHT.	285
Figure B.43	Propagation of the model parameters uncertainty on FEBA test No. 222 for the clad temperature output (TC). The posterior samples are from the calibration scheme w/o Bias.	286
Figure B.44	Propagation of the model parameters uncertainty on FEBA test No. 216 for the pressure drop output (DP). The posterior samples are from the calibration scheme w/ Bias, All.	287
Figure B.45	Propagation of the model parameters uncertainty on FEBA test No. 216 for the pressure drop output (DP). The posterior samples are from the calibration scheme w/ Bias, no dffbVIHT.	287
Figure B.46	Propagation of the model parameters uncertainty on FEBA test No. 216 for the pressure drop output (DP). The posterior samples are from the calibration scheme w/o Bias.	287
Figure B.47	Propagation of the model parameters uncertainty on FEBA test No. 214 for the pressure drop output (DP). The posterior samples are from the calibration scheme w/ Bias, All.	288
Figure B.48	Propagation of the model parameters uncertainty on FEBA test No. 214 for the pressure drop output (DP). The posterior samples are from the calibration scheme w/ Bias, no dffbVIHT.	288
Figure B.49	Propagation of the model parameters uncertainty on FEBA test No. 214 for the pressure drop output (DP). The posterior samples are from the calibration scheme w/o Bias.	288
Figure B.50	Propagation of the model parameters uncertainty on FEBA test No. 223 for the pressure drop output (DP). The posterior samples are from the calibration scheme w/ Bias, All.	289

Figure B.51	Propagation of the model parameters uncertainty on FEBA test No. 223 for the pressure drop output (DP). The posterior samples are from the calibration scheme w/ Bias, no dffbVIHT.	289
Figure B.52	Propagation of the model parameters uncertainty on FEBA test No. 223 for the pressure drop output (DP). The posterior samples are from the calibration scheme w/o Bias.	289
Figure B.53	Propagation of the model parameters uncertainty on FEBA test No. 218 for the pressure drop output (DP). The posterior samples are from the calibration scheme w/ Bias, All.	290
Figure B.54	Propagation of the model parameters uncertainty on FEBA test No. 218 for the pressure drop output (DP). The posterior samples are from the calibration scheme w/ Bias, no dffbVIHT.	290
Figure B.55	Propagation of the model parameters uncertainty on FEBA test No. 218 for the pressure drop output (DP). The posterior samples are from the calibration scheme w/o Bias.	290
Figure B.56	Propagation of the model parameters uncertainty on FEBA test No. 220 for the pressure drop output (DP). The posterior samples are from the calibration scheme w/ Bias, All.	291
Figure B.57	Propagation of the model parameters uncertainty on FEBA test No. 220 for the pressure drop output (DP). The posterior samples are from the calibration scheme w/ Bias, no dffbVIHT.	291
Figure B.58	Propagation of the model parameters uncertainty on FEBA test No. 220 for the pressure drop output (DP). The posterior samples are from the calibration scheme w/o Bias.	291
Figure B.59	Propagation of the model parameters uncertainty on FEBA test No. 222 for the pressure drop output (DP). The posterior samples are from the calibration scheme w/ Bias, All.	292
Figure B.60	Propagation of the model parameters uncertainty on FEBA test No. 222 for the pressure drop output (DP). The posterior samples are from the calibration scheme w/ Bias, no dffbVIHT.	292
Figure B.61	Propagation of the model parameters uncertainty on FEBA test No. 222 for the pressure drop output (DP). The posterior samples are from the calibration scheme w/o Bias.	292

Figure B.62	Propagation of the model parameters uncertainty on FEBA test No. 216 for the liquid carryover outputs (CO) from three different calibration schemes. 293
Figure B.63	Propagation of the model parameters uncertainty on FEBA test No. 214 for the liquid carryover outputs (CO) from three different calibration schemes. 293
Figure B.64	Propagation of the model parameters uncertainty on FEBA test No. 223 for the liquid carryover outputs (CO) from three different calibration schemes. 294
Figure B.65	Propagation of the model parameters uncertainty on FEBA test No. 218 for the liquid carryover outputs (CO) from three different calibration schemes. 294
Figure B.66	Propagation of the model parameters uncertainty on FEBA test No. 220 for the liquid carryover outputs (CO) from three different calibration schemes. 295
Figure B.67	Propagation of the model parameters uncertainty on FEBA test No. 222 for the liquid carryover outputs (CO) from three different calibration schemes. 295
Figure C.1	Flowchart of <code>gsa-module</code> . 297
Figure C.2	Flowchart of <code>trace-simexp</code> . 299
Figure D.1	Illustration of inverse transform sampling. 307
Figure D.2	Illustration of a 3-State Markov Chain 311
Figure D.3	Illustration of an irreducible 3-State Markov Chain 312
Figure D.4	Examples of periodic and aperiodic chains. 313

LIST OF TABLES

Table 1.1	Number of publications related to different meta-modeling approaches based on Scopus web search as of Feb. 14. 2017. 25
Table 2.1	FEBA test series I experimental conditions 41
Table 2.2	Locations of the thermocouples and the pressure drop measurements in the FEBA experiment. 42
Table 2.3	Geometrical parameters and experimental conditions for the FEBA model in TRACE. 44

Table 2.4	Selected TRACE input parameters (controllable inputs), their perturbation factors and their range of variations. 46	
Table 2.5	Selected TRACE input parameters (model parameters), their perturbation factors and their range of variations 48	
Table 3.1	Monte Carlo estimators to estimate the main-effect indices 72	
Table 3.2	Parameters importance across different outputs, average quantities over the transient. Checkmark signifies a parameter with a Sobol' total-effect indices above 5% and shaded cells signify the final selection of the retained influential parameters. 76	
Table 4.1	Simulation experiment settings for constructing and assessing the GP PC metamodel of the TRACE model of FEBA test No. 216 129	
Table 4.2	Predictive performance of the selected GP PC metamodel on the testing dataset of size 5'000 136	
Table 5.1	Bayesian calibration schemes conducted for the TRACE reflood model parameters against data from FEBA test No. 216. 189	
Table 5.2	Estimated autocorrelation times for the eight model parameters with respect to the ensemble running average and standard deviation, for the calibration scheme w/ Bias, All. 197	
Table 5.3	Summary of calibration results. The three numbers in brackets are the lower 95% credible interval, the median, and the upper 95% credible interval, respectively. 200	
Table A.1	The terms in TRACE two-fluid model mass balance equations 227	
Table A.2	The terms in TRACE two-fluid model momentum balance equations 228	
Table A.3	The terms in TRACE two-fluid model energy balance equations 230	
Table B.1	Parameters importance ranking with respect to average clad temperature output at $z \approx 4.1$ [m] (TC1) 243	
Table B.2	Parameters importance ranking with respect to the average clad temperature output at $z \approx 3.5$ [m] (TC2) 244	
Table B.3	Parameters importance ranking with respect to average clad temperature output at $z \approx 3.0$ [m] (TC3) 245	

Table B.4	Parameters importance ranking with respect to the average clad temperature output at $z \approx 2.4$ [m] (TC4) 246
Table B.5	Parameters importance ranking with respect to the average clad temperature output at $z \approx 1.9$ [m] (TC5) 247
Table B.6	Parameters importance ranking with respect to the average clad temperature output at $z \approx 1.3$ [m] (TC6) 248
Table B.7	Parameters importance ranking with respect to the average clad temperature output at $z \approx 0.8$ [m] (TC7) 249
Table B.8	Parameters importance ranking with respect to the average clad temperature output at $z \approx 0.3$ [m] (TC8) 250
Table B.9	Parameters importance ranking with respect to the average bottom pressure drop output (DP Bot., the segment between $z = 0.0$ [m] and $z = 1.7$ [m]) 251
Table B.10	Parameters importance ranking with respect to the average middle pressure drop output (DP Mid., the segment between $z = 1.7$ [m] and $z = 2.3$ [m]) 252
Table B.11	Parameters importance ranking with respect to the average top pressure drop output (DP Top, the segment between $z = 2.3$ [m] and $z = 4.1$ [m]) 253
Table B.12	Parameters importance ranking with respect to the average total pressure drop output (DP Tot., the segment between $z = 0.0$ [m] and $z = 4.1$ [m]) 254
Table B.13	Parameters importance ranking with respect to the average liquid carryover output (CO) 255
Table B.14	Main-effect and total-effect sensitivity indices for 12-parameter FEBA model with respect to the maximum clad temperature at the mid-height of the assembly as the QoI. 258
Table B.15	Main-effect and total-effect sensitivity indices for 12-parameter FEBA model with respect to the time of quenching at the mid-height of the assembly as the QoI. 259
Table B.16	Main-effect and total-effect sensitivity indices for 12-parameter FEBA model with respect to the 1 st fPC scores of the registered clad temperature transient at the mid-height of the assembly as the QoI. 259

Table B.17	Main-effect and total-effect sensitivity indices for 12-parameter FEBA model with respect to the 2 nd fPC scores of the registered clad temperature transient at the mid-height of the assembly as the QoI. 260
Table B.18	Main-effect and total-effect sensitivity indices for 12-parameter FEBA model with respect to the 1 st fPC scores of the warping function for the clad temperature transient at the mid-height of the assembly as the QoI. 260
Table B.19	Main-effect and total-effect sensitivity indices for 12-parameter FEBA model with respect to the 1 st fPC scores of the pressure drop transient at the middle of the assembly as the QoI. 261
Table B.20	Main-effect and total-effect sensitivity indices for 12-parameter FEBA model with respect to the 1 st fPC scores of the liquid carryover transient as the QoI. 261

LIST OF ALGORITHMS

Figure 1	Brute Force Monte Carlo (MC) for estimating $\mathbb{V}_d[\mathbb{E}_{\sim d}[y x_d]]$ 70
Figure 2	Metropolis-Hastings Algorithm 164
Figure 3	Affine-Invariance Ensemble Sampler (Stretch-Move) 172
Figure 4	Inverse Transform Sampling 306

QUANTIFYING UNCERTAINTY OF COMPUTER MODEL: FORWARD AND BACKWARD

“All models are wrong but some are useful” – George Box

It is perhaps convenient to use the quote by Box – at least half of it – as an excuse if a modeling exercise goes awry. But engineers are constructive bunch, as they are pragmatic, so they often focus on the second part of the statement and try to do better. Some would argue that to make a model useful is to make a model less wrong, a very difficult task. Some others would start by making an effort assessing whether the “wrong” model is useful, can ever be useful, or can be made useful without any direct improvement to the model. The two views are not contradictory, although the latter is arguably more humble. This thesis is an effort in line with the latter view.

By many measures, TH system codes for simulating system behavior of a nuclear power plant (NPP) are an achievement. Their development, by the best and the brightest, includes decades of verification and validation (V&V) and validation activities supported by numerous experimental facilities, small and large scales. Many of the current understanding of physical phenomena in NPP transient were established during that period. Yet, their predictions can still be off when compared against experimental data. The efforts to minimize this difference by developing high-fidelity physical models coupled with high-resolution numerical algorithms are always on-going and are indispensable for moving forward.

At the same time, simulations are being continuously used to make decisions, from optimal system design to safety margin evaluation for reactor licensing. For robust decision-making, it is important to acknowledge and determine the uncertainties associated with the predictions. Thus, independently of the efforts to improve the code, uncertainty quantification of the code predictions is an important part of the code development; it is the main topic of this thesis.

This opening chapter introduces briefly the importance and challenges of using system codes for simulating the TH behavior of NPPs in the context of their safety analysis. Section 1.1 starts with basic definitions of relevant terms used throughout the thesis before moving on to a brief introduction to safety analysis and TH system code. Uncertainty analysis of TH system codes is first discussed in Section 1.2, outlining the background and the context of the doctoral research.

Section 1.3 then describes the statement of the problem, the objectives, and the scope of the research. This thesis proposed a methodology comprised of sequential steps to analyze a computer model (i.e.,

TH system code) with the overall goal of quantifying the uncertainty associated with the model parameters based on experimental data. It consolidates and adapts recent developments in the applied statistics literature. In this context, Section 1.4 provides a broad, but by no means exhaustive, overview of the research landscape on each proposed steps. Finally, Section 1.5 concludes the chapter by outlining the structure of the thesis.

1.1 COMPUTER SIMULATION AND SAFETY ANALYSIS OF NUCLEAR POWER PLANT

1.1.1 *Scientific Computer Simulation*

Scientific computer simulation

The ubiquity of computer simulation in science and engineering has resulted in numerous definitions of the term *scientific computer simulation*, *model*, and *simulation*. To avoid confusion, this thesis adopts a recent definition proposed by Kaizer et al.[1] quoted below:

Scientific Computer Simulation is the imitation of a behavior of a system, entity, phenomenon, or process in the physical universe using limited mathematical concepts, symbols, and relations through the exercise or use of scientific computer model.

model, simulation, scientific simulation, and scientific computer simulation

This definition highlights three main points. First, this definition accentuates the difference between *model* and *simulation*. A model deals with the notion of representation of a system, while a simulation deals with the notion of imitation of a behavior of that system. Secondly, a model is said to be scientific when it represents a real world system as its subject. Finally, the modifier *computer* generally implies that the mathematical models cannot be solved analytically and their solutions require a computer. Because the associated numerical approximations can affect its solution, many computational-related aspects often need to be considered. This thesis only deals with computer simulation.

Perceptual, formal, and procedural models

Beven [2] articulates this definition of a scientific model further through the following distinctions: a perceptual model (i.e., the theoretical description of the physical phenomena), a formal model (i.e., its mathematical description), and a procedural model (i.e., the computer implementation of the formal model). For many physical models of complex system, only the procedural model is able to make a quantitative prediction of the system behavior. These distinctions are useful in acknowledging the level of approximation involved in modeling.

Code

A computer software that implements scientific models down to the solution algorithms is called a *scientific code* or simply a *code* [3]. Many modern implementations of scientific codes, apart from possibly being specific to a scientific domain, are comprehensive platforms.

For instance, in the context of TH system modeling, such codes allow modeling various attributes of the system ranging from its geometry, initial and boundary conditions, and design variables to the settings for discretization scheme and numerical solver.

A *simulation* or a *calculation* [3] using a code can only be made on a particular well-specified system, where all the aforementioned attributes (geometry, initial and boundary conditions, etc.) have been completely fixed or specified. As a result, the terms *computer simulation model* or *simulator* include not only the code itself, but also the particular system of interest being modeled using the code [4].

Simulator

1.1.2 Codes and Safety Analysis of Nuclear Power Plant

Scientific codes play a central role in deterministic safety analysis of NPPs. They provide a *physics-based* evaluation of relevant phenomena taking place in the plant during postulated transients to demonstrate that safety requirements are met [5]. The demonstration is carried out with respect to acceptance criteria, a set of limits and conditions ensuring the integrity of the safety barriers. The criteria are set by regulatory bodies for normal and off-normal operation of the plants.

The physics-based evaluation is achieved through simulation. As noted in [5, 6], there are four disciplines associated with the different physical processes relevant in the safety analysis of the plant behavior: the *neutronics* of the core; the *thermo-mechanics* of the fuel and reactor components; the *radiological* analysis of a possible release; and, the *system thermal-hydraulics* of the plant, the subject of this thesis¹. Each discipline is, in turn, characterized by a distinct set of governing physical equations and that are often solved by a distinct code.

Codes in safety analysis of NPP

The NPP safety is established, among other things, by setting the acceptance criteria in terms of limiting physical quantities relevant for the phenomena involved. The upper tolerance limit of 1'204 [°C] for the peak clad temperature (PCT) is one such criteria for LWRs [7]. Whether the physical quantities respect such limits during postulated scenarios is analyzed using simulations either in a conservative or best-estimate approach [5].

During its early days, reactor safety analysis involved a high-degree of conservatism. Conservatism called for the most pessimistic and penalizing modeling assumptions (including initial and boundary conditions) to ensure conservative results, that is far below their expected values. This approach, was justified by limited modeling capabilities and limited knowledge of the physical process involved. However, it was later found that there are conditions for which conservative assumptions do not lead to conservative (or even physical) predictions.

Conservative analysis

¹ Ref. [6] added one additional key discipline, namely: reliability analysis. It is excluded in the above listing as it is not technically a discipline of *physics*.

An illustration

As an example, consider the analysis for a loss-of-coolant accident (LOCA) of an LWR. Assuming less interfacial shear between the liquid and the gas phases of the coolant (water) reduces mist flow and is a conservative assumption because less heat is transferred to the coolant flow in the upper region of the core, which penalizes the fuel temperature prediction. But this assumption also reduces that the time to refill the core as more liquid is retained in the reactor cooling system. Furthermore, with less shear, there is less resistance in injecting emergency coolant into the core (condition known as the counter-current flow limitation). Both effects are clearly not conservative and put into question the conservatism of the prediction [5].

Best-estimate analysis

Because of this example and many others [5], a more accurate prediction of two-phase flow transient behavior under accident conditions was deemed necessary. As opposed to the conservative approach, *best-estimate* approach calls for (more) physically sound thermal-hydraulics models with more realistic assumptions, which are backed up by experimental data obtained from numerous experimental programs conducted in Separate and Integral Effect Test Facilities. In that context, Best-estimate TH system codes were developed to provide more realistic predictions. The codes were designed to be comprehensive tools capable of simulating realistically a wide range of transients foreseen in LWR operation, and were developed using the current best understanding of flow processes expected to happen during the transients.

1.1.3 *Thermal-Hydraulics (TH) System Codes*

TH system code

A TH system code is a tool to simulate the flow behavior of the reactor coolant during transients. This implies solving time-dependent conservation equations, describing the two-phase fluid flow inside the coolant circuit, coupled with a heat conduction equation, describing the heat transfer between fluid and heated elements (e.g., fuel rods). The simulation of the plant behavior also requires an explicit modeling of the geometry, components, equipments, and systems that are specific to LWRs [6].

Nodalization

The coolant circuit of an LWR is a complex system. The system includes the reactor pressure vessel with hundreds of fuel assemblies; kilometers of interconnecting pipes; scores of valves, pumps, and tanks; as well as numerous special components like steam generators and condensers. The first major simplification made for describing the fluid flow in the coolant circuit is to average the fluid on the surface perpendicular to its flow (i.e., *flow area averaging* – see Chapter 2). This results in a 1-dimensional *nodalization* of the circuit. Through nodalization, an LWR is decomposed into a set of interconnected *nodes* which holds discretized information of fluid flow (see

Fig. 1.1). Due to the 1-dimensional simplification of the flow², a node is only characterized by its *fluid cell* (with attributes of length and free volume) and its *faces* (with attributes of flow area, hydraulic diameter, and orientation).

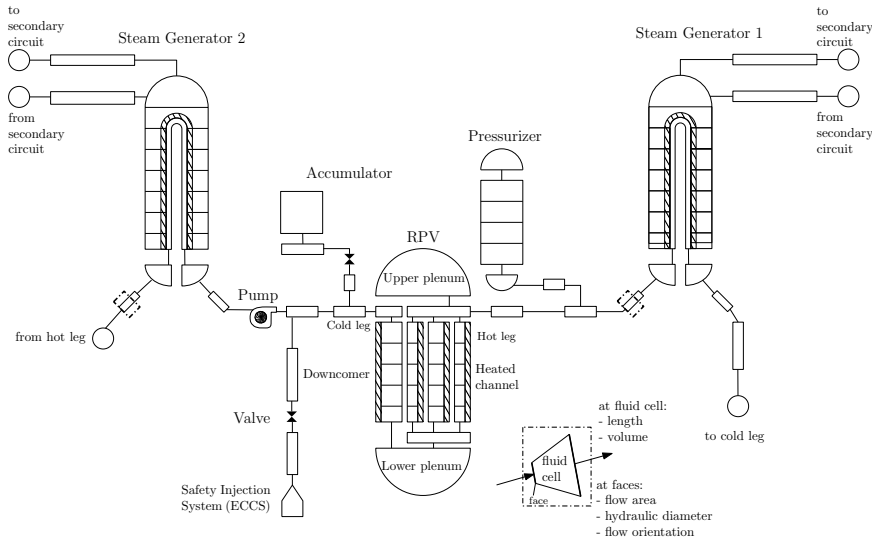


Figure 1.1: Nodalization of an NPP in a thermal-hydraulics (TH) system code. Shaded elements are heated elements, where heat exchange occurs between the element and the fluid.

The typical structure of a system code is illustrated in Fig. 1.2. As shown, a system code constitutes of several building blocks that can be used to model and simulate wide ranges of systems and conditions. It includes a set of conservation equations, closure laws, and equation of states. System codes are complemented with models for special components that perform specific functions (e.g., heated solid structure, pumps, and separators) or actions during transients (e.g., valves, instrumentation, and control systems); and models for special processes and phenomena that are relevant to the LWRs but too complex to be captured implicitly in the (simplified) conservation equations (e.g., critical flow). In fact, the inclusion of models for those components and processes are the defining characteristics of TH system code [6].

Structure of thermal-hydraulics system codes

The core element of a system code is a set of conservation equations describing the dynamics of the state variables of the fluid. The state-of-the-art model widely implemented in TH system codes to describe the dynamics of fluid flow in NPPs (specifically, LWR) is based on the *two-fluid model*. This model separately treats the transport phenomena of the two-phases of fluid flow (gas and liquid) resulting in a set of

Two-fluid model

² Some system codes allow a 3-dimensional modeling for selected components, mainly the reactor pressure vessel where 3-dimensional effects might be of relevance to safety analysis. However, as of today, no system code supports full 3-dimensional modeling of all the components in the coolant circuit.

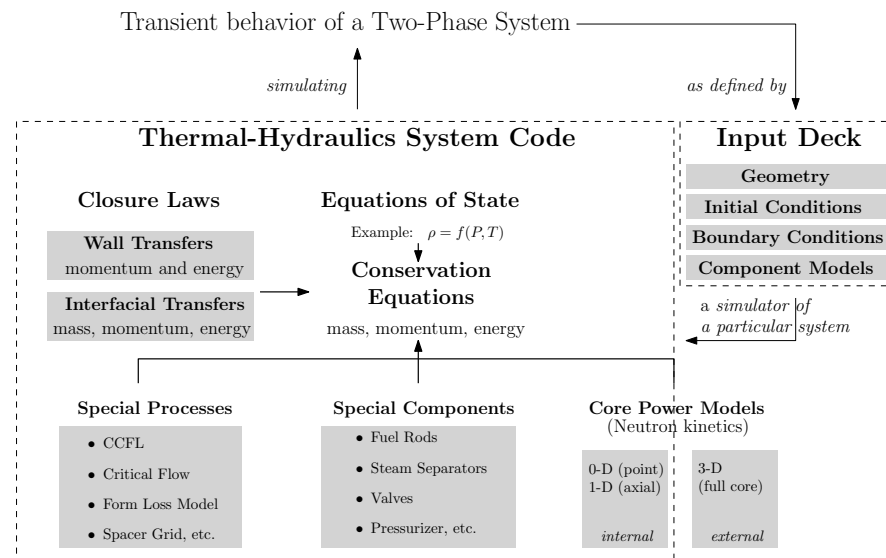


Figure 1.2: Generic structure of a thermal-hydraulics (TH) system code. The code and an input deck define a *simulator* of a system.

six balance equations (mass, momentum, and energy for each of the two phases). The model can capture phenomena where thermal and mechanical non-equilibrium conditions exist between the two phases, giving more realistic picture in a wide range of transients.

The validity of the two-fluid model relies on the proper modeling of the transfer terms between phases and between each phase and the boundary walls. The transfer terms include interfacial drag, interfacial heat transfer, and wall heat transfer. In principle, any two-phase flow pattern exhibits particular phase distributions and interfacial structures. As a result, the mathematical expressions of the transfer terms change with the pattern of the two-phase flow. As the transfer terms represent different physical processes taking place for each flow pattern, they constitute the *physical models* of a system code.

These physical models, so-called *closure laws*, close the set of balance equations for mass, momentum, and energy of the two phases. Based on their origins, closure laws can be classified into three categories: fully empirical, fully mechanistic, and semi-empirical [8]. Fully empirical closure laws are based only on the available representative experimental data by correlating transfer terms of interest with observed flow variables. Given comprehensive experimental data, these models tend to be accurate within the range of experimental conditions (i.e., its validation domain). On the other hand, an extrapolation outside of that range can give dubious results.

A fully mechanistic (i.e., *phenomenological*) approach for developing closure laws lies at the other end of the spectrum. Using this approach, a physical mechanism that governs the phenomena of interest is postulated. Experimental data plays a role only in validating such

*Transfer terms,
physical models*

Closure laws origin

*Fully empirical
approach*

*Fully mechanistic
approach*

a postulated model. If the model cannot be supported by the data then a complete revision might be required. Mechanistic approach to closure laws modeling provides a scientific basis for prediction outside the validation data range (i.e., extrapolation). However, its quality strongly depends on the adequacy of the postulated model and the associated assumptions.

Lastly, the semi-empirical approach combines both approaches, i.e., an initial mechanistic model which is tuned using parameters that are fitted to match experimental data. These parameters then become a measure of the inadequacy of the postulated model in explaining the data due to any unaccounted physical processes.

*Semi-empirical
approach*

Any of these approaches proved to be a difficult effort [9–11] due to various reasons ranging from the lack of knowledge of the underlying physical process (with respect to the fully mechanistic modeling) to limitation in the amount and precision of the measured data (with respect to the fully empirical approach). Simplifying assumptions and extrapolations are made because of these limitations. In the end, closure laws in system codes are of mixed origins and they become a major source of uncertainty³ in the application of TH system codes, especially when used outside their validation domains.

1.2 UNCERTAINTY QUANTIFICATION IN NUCLEAR ENGINEERING THERMAL-HYDRAULICS

Before continuing the discussion of uncertainty analysis of code predictions, this section defines some additional terminologies to avoid later confusion.

The notion of *simulator* introduced in Section 1.1 is depicted in a more generic way, as an input/output model in Fig. 1.3.

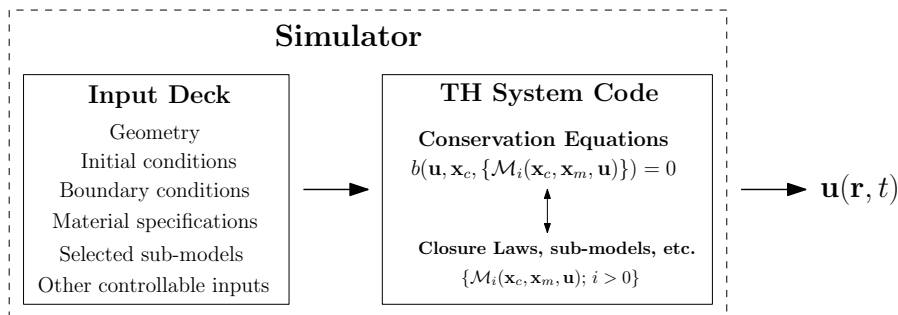


Figure 1.3: Simplified illustration of a simulator as an input/output model.

The input deck defines a specific problem (i.e., system) of interest and can be seen as the input of TH codes. It includes the geometrical configuration (i.e., the nodalization), the material and fluid involved, the initial and boundary conditions, and possibly the settings for the numerical solver. Some of these specifications (such as the bound-

*Controllable inputs
and model
parameters*

³ defined in this thesis as a state of limited knowledge, that is of *epistemic* nature.

ary conditions) are parametrized and constitutes *controllable inputs* denoted by \mathbf{x}_c . The simulator is to be run for a given controllable input value⁴. The conservation equations of the code are closed with additional set of closure laws (and other sub-models) $\mathcal{M}_i(\mathbf{x}_c, \mathbf{x}_m, \mathbf{u})$. These closure laws are, in turn, parametrized by a set of model specific parameters denoted by \mathbf{x}_m which are referred to as the *physical model parameters*. Both the controllable inputs and the physical model parameters are considered by the code as *inputs*.

Specifying the input deck, as far as the user is concerned, completely defines the problem and the code solves the conservation equations (Fig. 1.3) to estimate the physical variables $\mathbf{u}(\mathbf{r}, t)$ (where \mathbf{r} and t denote space and time variables, respectively) associated with the fluid flow and heat structure (e.g., fluid pressure, temperature, wall temperature, etc.). These “raw” outputs are further post-processed to obtain relevant QoIs for the problem at hand (e.g., max. temperature, max. pressure, onset time, etc.).

1.2.1 Forward Uncertainty Quantification

Best-estimate analysis attempts to describe as realistically as possible the behaviors of the physical processes that occur during a plant transient. And yet, neither complete understanding nor enough data is always available to adequately simulate these complex physical phenomena. Simplifying assumptions, approximations, and expert judgments remain to some degree unavoidable for a complete analysis.

*Best-estimate plus
uncertainty*

Hence, best-estimate analysis has to be complemented with uncertainty analysis. The ultimate goal of uncertainty analysis is to associate code prediction with its uncertainty. These combined quantities are then compared with safety limits (e.g., peak clad temperature (PCT)) to check whether the limits still fall outside the uncertainty band of the code prediction.

*Sources of
uncertainty*

There are several known sources of uncertainty that render the prediction on $\mathbf{u}(\mathbf{r}, t)$ and its derived quantities uncertain. The sources of primary interest in the present research are:

1. *Uncertainty associated with the controllable inputs.* In the case of a controlled experiment, controllable inputs are observed and *controlled* for. However, their observations might contain errors due to instrument imprecision or inherent variability. When simulating a real accident scenario in a plant, plant parameters prior to the accident scenario can also be measured and constitute uncertain controllable inputs. In addition, parameters defining the accident scenario, such as the break size in a LOCA, or the availability and performance of safety systems can also be treated as uncertain controllable inputs [12].

⁴ Later on, *controllable* inputs correspond to the parameters whose counterparts in a physical experiment which can be controlled by the experimentalist.

2. *Uncertainty associated with the physical model parameters.* The value of the physical model parameters are often not known a priori. Thus, the uncertainties are epistemic. They can either be estimated using data from a calibration experiment or by expert judgment.
3. *Uncertainty associated with the physical models.* The physical models themselves are still approximations, even with perfectly known model parameters. If derived in a fully mechanistic manner, some important processes might be unaccounted for due to the inherent complexity and lack of knowledge (i.e., the case of *missing physics*). On the contrary, if derived fully empirically, models might be derived separately for different elementary processes, while in the applications of the code multiple such models are used in concert. Despite each being validated, it is fair to question the validity of models used in an ensemble. Any of the two tends to cause a systematic bias on the code prediction, the extend of which is unknown and uncertain. As a result, this source of uncertainty is referred to as model *bias*, *inadequacy*, or *discrepancy*.

In uncertainty analysis, the controllable inputs and physical model parameters are modeled as random variables (\mathcal{X}_c and \mathcal{X}_m , respectively) equipped with probability density functions (PDFs). By transforming the random variable inputs, the simulator output becomes random variable as well

$$\mathbf{U}(\mathbf{r}, t) = f(\mathcal{X}_c, \mathcal{X}_m; \mathbf{r}, t)$$

where f represents the simulator as a mathematical function. The QoI related to the random outputs can be summarized by different integral quantities. For instance, the mean of a QoI given by function g is

$$\mathbb{E}[g] = \int_{\mathcal{X}_c, \mathcal{X}_m} g(f(\mathbf{x}_c, \mathbf{x}_m; \mathbf{r}, t)) p(\mathbf{x}_c, \mathbf{x}_m) d\mathbf{x}_c d\mathbf{x}_m$$

where $p(\mathbf{x}_c, \mathbf{x}_m)$ denotes the joint PDF for the input parameters.

Using Monte Carlo (MC) techniques, samples are generated from the joint input parameters distribution and are used to run the code multiple times. Afterward, the resulting code outputs (raw or post-processed), are summarized to obtain the uncertainty measure of the prediction. In other words, the uncertainties in the controllable inputs and physical model parameters are *propagated forward* through the code to quantify the uncertainty of the predictions as shown in Fig. 1.4. The practice of propagating parametric uncertainty by MC is widely accepted in the nuclear engineering thermal-hydraulics community [13–16].

*Forward uncertainty
quantification*

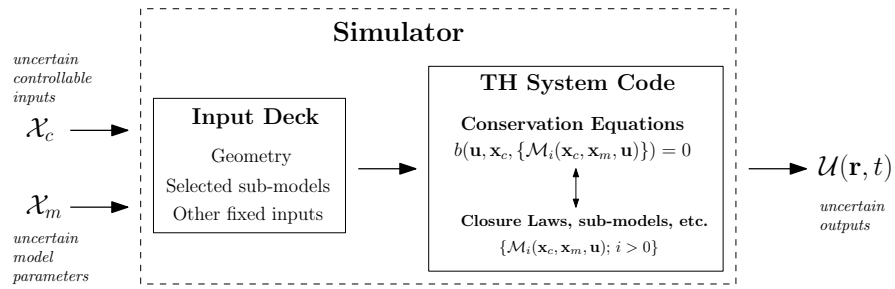


Figure 1.4: Simplified flowchart of forward uncertainty quantification of a simulator prediction. Notice that the simulator has been parametrized by the controllable inputs and physical model parameters, each of which are represented as a random variable.

1.2.2 Inverse (Backward) Uncertainty Quantification

Model parameters

A lot has been said about the origin of the uncertainty associated with the controllable inputs. The physical model parameters, however, are conceptually different. The physical models referred to in this thesis are usually represented either in the form of correlations, phenomenological models, or a mixed between the two (see Section 1.1.3). Therefore, the model parameters do not necessarily have a physical meaning (see Chapter 5) and the source of their uncertainties vary with the type of model. For instance, in an empirical model the model parameters are the curve-fitting parameters and their uncertainties are observable and can be associated with the dispersion of the data.

Separate Effect Test Facilities (SETFs)

However, many physical models, be it empirical or mechanistic, are originally derived from experiments on simple systems that do not, strictly speaking, reflect the flow conditions in an LWR (e.g., heated tube vs. rod bundle, low pressure vs. high pressure, etc.) [8]. Thus, to better represent the flow characteristics in reactor transient, experiments with well-specified conditions are conducted in SETFs, facilities aimed at reproducing a particular safety-relevant phenomena during transient at a particular part of the reactor [6].

Calibration against SETFs

The data are used to assess the physical models. In the assessment, some parameters in the models are adjusted to match the experimental data [9]. Alternatively, additional free parameters can be introduced in the models to serve the same purpose [8]. That is, the parameters are tuning parameters and become measures of the models inadequacy in reproducing the data. Ultimately, optimal values for the parameters are estimated and implemented in the code.

In light of this, it can be argued that the uncertainty associated with the tuning parameters stems from the fact that the calibration was conducted only on limited set of data obtained from selected SETFs. As different SETFs exist for the same phenomena, it is fair to ask if the calibrated value will hold if the calibration were to be conducted on other SETFs data. Additionally, as tuning parameters,

expert-judgment is also often used to estimate the uncertainty. Experts fixed the range of variation of the parameters based on their expectation of the model performance.

To derive the uncertainty associated with the model parameters described above, the problem can be posed as an inverse problem. In this setting, given a set of experimental data $\{\mathbf{D}\}$ taken with known controllable inputs \mathbf{x}_c , the task is to infer the value of the *unobserved* parameters in the physical model used to predict the same quantity as the experimental data. To avoid excessive bias towards the calibration data, it is important here to acknowledge the observation errors of the experimental data and the controllable inputs, and the possible systematic bias of the associated models.

An inverse problem

In a probabilistic setting, a way to make an inference of unobserved parameters based on observed data is through the Bayes' theorem,

Inverse uncertainty quantification

$$p(\mathbf{x}_m | \{\mathbf{D}\}, \mathbf{x}_c) = \frac{p(\{\mathbf{D}\} | \mathbf{x}_m, \mathbf{x}_c) \cdot p(\mathbf{x}_m)}{\int p(\{\mathbf{D}\} | \mathbf{x}_m, \mathbf{x}_c) \cdot p(\mathbf{x}_m) d\mathbf{x}_m}$$

where the left-hand side of the equation is the posterior probability density of the model parameters \mathbf{x}_m conditioned on the observed data $\{\mathbf{D}\}$ and controllable inputs \mathbf{x}_c . The right-hand side constitutes of the likelihood function $p(\{\mathbf{D}\} | \mathbf{x}_m, \mathbf{x}_c)$ (probability of observing data given the parameters), the prior of the model parameters $p(\mathbf{x}_m)$ (the initial state of knowledge regarding the parameters values before observing the data), while the denominator is a normalizing constant such that the posterior is a valid PDF (that is, it integrates to one)⁵. The posterior represents the knowledge one has on the model parameters values conditioned on the data under the modeling assumption. Fig. 1.5 depicts a simplified flowchart of the inverse quantification.

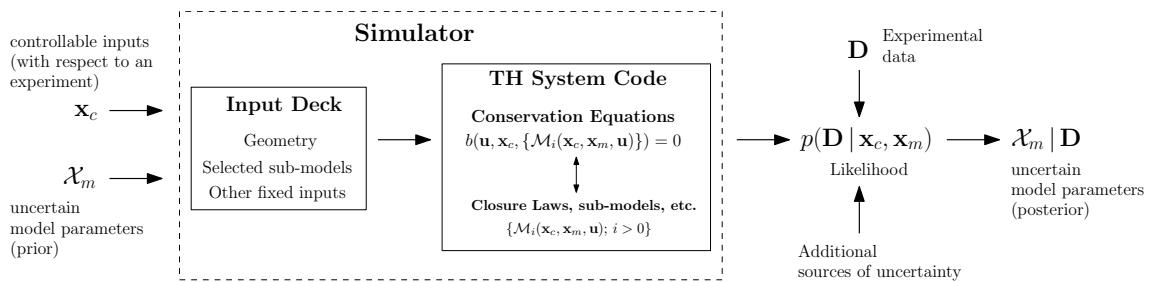


Figure 1.5: Simplified flowchart of inverse quantification for model parameters of a simulator.

The formulation and computation of the posterior above can be seen as a calibration exercise. That is, it seeks to adjust the model parameters such that the predictions of the simulator are consistent with the observed (i.e., calibration) data under the assumed likelihood and the prior. However, instead of obtaining a single estimated

Statistical calibration

⁵ Note that the formulation assumes the controllable inputs \mathbf{x}_c are fully known. If they are considered uncertain, such as due to their inherent variability, then a prior probability can be put on them as well.

value (or values in case of multiple parameters), the resulting posterior is a joint PDF, conditioned on the observed data. In relation to the aforementioned expert-judgment for estimating the parameters uncertainty, the approach uses the experimental data to better inform the prior expectation about the model parameters values. The posterior PDF, in turn, can be used in uncertainty propagation to quantify the uncertainty on the prediction made outside the calibration data.

The importance of characterizing the uncertainty in the physical models parameters was acknowledged by the Working Group on the Analysis and Management of Accidents (WGAMA) of the Organization for Economic Cooperation and Development (OECD)/Nuclear Energy Agency (NEA). This led to the Post-BEMUSE Reflood Models Input Uncertainty Methods (PREMIUM) project. Its main goal is to report the state-of-the-art methodologies to quantify the uncertainty in the physical models parameters. The following will briefly describe the project and highlight the selected main lessons learned from the author's perspective through his participation on behalf of the Paul Scherrer Institut (PSI) [17].

1.2.3 OECD/NEA PREMIUM project

The PREMIUM project was an activity launched by the OECD/NEA with the aim to advance the methods for quantifying the uncertainties associated with the physical model parameters in TH system codes. It was the continuation of the project Best-Estimate Methods – Uncertainty and Sensitivity Evaluation (BEMUSE), which concentrated on the propagation and sensitivity analysis of the input uncertainties in large scale simulation (large break loss-of-coolant accident (LBLOCA)). The main finding of BEMUSE can be found in [18]. The emphasis of the PREMIUM benchmark was placed on the derivation of the model parameters uncertainties and their validation.

The scope of the project was limited to the simulation of the phenomenon of core reflood and quenching under conditions representative of a pressurized water reactor (PWR) large break LOCA. Experimental data from two SETFs was made available for the purpose of uncertainty quantification of the model parameters as well as validation. For the model parameters uncertainty quantification, the data from the FEBA reflood facility was used. The derived uncertainties were then propagated and compared with the experimental data from other experimental runs of FEBA and from another reflood facility (PERICLES). Thus the main goal of the project followed the approach of statistical uncertainty analyses explained above.

Sixteen organizations from 11 different countries participated in the 4-year project (2012–2016) using 6 different TH system codes. Each participant employing a chosen simulation code and methodology had to contribute to the 5 following phases of the benchmark:

1. PHASE 1: Description of the selected simulation code and methodology.
2. PHASE 2: Identification of the uncertain parameters that are most relevant to PWR LOCA reflooding simulations.
3. PHASE 3: Quantification of the uncertainties in the parameters, using available data from the FEBA experiment.
4. PHASE 4: Propagation of the quantified uncertainties as part of a blind benchmark exercise based on data from the PERICLES experiment.
5. PHASE 5: Contribution to the analysis and synthesis of the benchmark results.

During the course of the project, three OECD/NEA reports have been published [19–21]. Details can be found in the reports. The following will describe briefly some of the lessons learned from PREMIUM of relevance to the present study.

PREMIUM provided state-of-the-art (as of 2015) uncertainty quantification methods for TH system codes. It emphasized methodological issues that are yet to be overcome. Some of these issues, such as the identification of important parameters, extrapolation of quantified results, scaling, and nodalization were already raised in the 1994 review studies on uncertainty methods for TH codes sponsored by the European Commission [22]. At the conclusion of PREMIUM, these issues are still considered open problems.

First, there was an apparent lack of consensus among participants (and thus, the community) for a systematic identification of important parameters in TH simulation models. Guidelines were indeed provided, but each participant eventually came up with their own selection criteria and methodology, some still relied solely on graph comparison of outputs when changing one parameter at a time [19].

Although complete exclusion of expert judgment is not feasible (nor advised), it is useful to take benefit from the progress made in the computer experiment community. For instance, the Morris screening method can be useful in the initial parameter identification and importance ranking process by making the analysis more systematic and robust. The method can provide a smooth transition from the more familiar one-at-a-time method adopted by most participants. Furthermore, there was a valid issue raised by a participant regarding the possibility of “complicated” code response from simultaneous parameters perturbation [23]. This can be interpreted as parameter interaction in the literature. In this case, GSA methods can help in the investigation about its presence.

Secondly, there was a slight disagreement between participants regarding the use of *calibrated parameter*. This notion stemmed from the use of a Bayesian method (the so-called CIRCÉ [20, 24], a method

*Identification of
important
parameters*

*Calibrated parameter
and best-estimate
code*

based on maximum likelihood approach under linear assumption coupled with normal prior for the parameter) to update the prior distribution of the model parameter such that the resulting posterior distribution yields the closest agreement with the experimental data. In the application of CIRCÉ, the nominal value of the model parameter was allowed to shift following the updated central measure of the posterior. Such practice of calibration was questioned because the best-estimate code used was, in fact, already calibrated on the basis of larger experimental databases over decades of V&V activities. In other words, the calibration over a very limited set of the FEBA tests would undermine the built-in (calibrated) models already in place.

That is a valid point of contention. The results of applying CIRCÉ were mixed. On one hand the experimental data from the FEBA experiment allowed CIRCÉ to reduce the initial uncertainty on the parameters and simultaneously improved the nominal case predictions. On the other hand, when the updated parameters were used in the uncertainty propagation of another FEBA test, the narrow uncertainty band on the predictions failed to cover some part of the experimental data. Moreover, when the same updated parameters were used in the blind uncertainty propagation for another facility there results were poor: poor nominal case prediction and too narrow uncertainty without covering the experimental data [17, 21].

*Overfitting and
extrapolation*

This indicated a symptom of *overfitting* in which the uncertain parameters were calibrated strongly on one data set and thus became very sensitive to a change of data set. The narrow uncertainty obtained indicates that the calibration procedure converged to a “wrong” values and thus was not applicable to extrapolation.

While the Bayesian approach makes sense for parameter calibration, updating its value in light of new data, its application might require accounting for additional sources of uncertainty. It also makes sense to acknowledge the extensive V&V activities that serve as the basis of TH system codes; observing one additional data set should not render previous results invalid right away. Thus it remains an open question how to compromise between learning from new data and preserving what has been learned before.

*The use of
metamodel*

Finally, there was a natural reluctance among the participants to embrace more recent methods requiring less assumptions (e.g., normal prior of the parameters, linearity between outputs and parameters, etc.) but requiring more code runs. The use of metamodel can help in alleviating such computational restriction⁶, insofar that the error incurred by the use of metamodel can be accepted. Strictly speaking, the community is not unfamiliar with the use of metamodel (a fast approximating function as a substitute of running the code) for uncertainty analysis, especially in the context of uncertainty propa-

⁶ It is not, however, cost-free as will be explained in more detail in Chapter 4.

gation⁷. However, in the context of inverse uncertainty quantification, none of the participant took the benefit of using metamodel and relaxed some of the assumptions in calibration.

1.3 OBJECTIVES AND SCOPE OF THE THESIS

With the larger context provided above, this section presents briefly and specifically the statement of the problem, followed by the objectives as well as the scope of the present doctoral research.

1.3.1 *Statement of the Problem*

System code development is an effort to consolidate correlations and mechanistic models to create a phenomenological-based simulation code that can provide best-estimate results. This consolidated effort results in a code that can simulate a wide range of transients foreseen in NPP operation in a best-estimate manner. Alas, to come up with a consistent set of closure laws is a great challenge for code developers.

The closure laws required to close the two-fluid model pose particularly difficult challenges [11]. For instance, to have a correlation of heat transfer between the wall and the fluid, temperature data from the wall, and the liquid and gas phases are needed. But measuring temperature of the individual phases in an arbitrary interfacial topology has its own technical difficulties to the extent that no such data is available to be implemented in the closure laws. Additionally, the experiments to obtain hydrodynamic closure laws (e.g., interfacial friction factor, wall friction factor) were generally carried out in adiabatic conditions. As a result, this excludes the coupling of any heat transfer phenomena between the phases and the wall in such correlations.

Furthermore, during the development of a code, programming considerations also came into the picture. For robustness, simplification is often required and continuity is enforced. Transitional flow regime for which data is not available is often modeled to be the average of the two known bounding regimes. Different code developments, which used different assumptions and experimental databases, come up with different set of closure laws with their own parametrization (see for instance [10] for TRAC code and [26] for CATHARE code). Several authors have expressed their concerns about the uncertainty stemming from the closure laws [6, 11, 27].

⁷ It was initially used for the estimation of PCT probability distribution from uncertainty propagation in the context of safety margin evaluation of LBLOCA scenario [25].

As an example of the above point, consider that in the TRACE code, after some derivations the interfacial drag coefficient closure law in the inverted slug flow regime $C_{i,IS}$ is given by,

$$C_{i,IS} = \hat{x}_{m,SET} \times \frac{1}{24} \frac{\rho_g (1 - \alpha)}{La \alpha^{1.8}} ; \hat{x}_{m,SET} = 0.75$$

where ρ_g is the density of the gas phase; La is the Laplace number; α is the void fraction; and $\hat{x}_{m,SET}$ is a fitting parameter.

There are several remarks that can be made about the closure law given above. First, the second term in the right-hand side was derived from experimental data but based on several simplifying assumptions. In the inverted slug regime, saturated liquid core breaks up into ligaments. These ligaments are *assumed* to take form as prolate ellipsoid. The drag coefficient is then taken from the experimental database of coefficient for distorted droplet. Then to take into account the multi-particle effect, the coefficient is divided by the void fraction α raised to the power of 1.8 (this, in turn, was taken from experimental data of inertial regime). Lastly, the first term of the equation, $\hat{x}_{m,SET} = 0.75$ was added *to calibrate against* the experimental data from the FLECHT-SEASET reflood experimental facility. This first term, although clearly *non-physical* in nature, is nevertheless an important tuning parameter of the model. Its uncertainty should be considered in uncertainty analysis, especially when reflood is expected to occur. Yet, no statement regarding the associated uncertainty is given. Similar adjustment on several other closure laws exists [28].

As illustrated above, it is clear that models in thermal-hydraulics system code are, to a certain extent, limited. Various experimental programs were carried out to better understand the important phenomena, and to validate (and, as noted, to calibrate) the models. Series of experiments, carried out in SETFs with well-specified boundary conditions were aimed to reproduce limited part of the transient in a selected component following a postulated scenario. For example, in the case of reflooding, several facilities existed and data was gathered (FEBA, PERICLES, etc.). But, there has not been an orchestrated effort to incorporate the accumulated data into the calibration process of the physical models, in a systematic way, while acknowledging the multiple sources of uncertainty in the process.

1.3.2 Objectives

The purpose of this research is to quantify the uncertainty of physical model parameters implemented in a TH system code. The physical models of interest describe the phasic interactions in a complex multiphase flow during a reactor transient, namely heat, mass, and momentum exchanges between vapor, liquid and structures. These models are parametrized by physical or empirical parameters, the

values of which are uncertain. This results in uncertain code predictions of important safety quantities, such as the evolution of the fuel clad temperature during a postulated reactor transient.

Adopting a probabilistic framework to conform to the statistical uncertainty propagation widely adopted in the field of nuclear engineering, the uncertainties in the parameters are represented as probability density functions (PDFs) or their approximations. The derivation of these functions is posed as an inverse statistical problem following a Bayesian framework as the parameters themselves are not directly observable. Although subjectivity cannot be removed completely from the analysis, the research aims to develop a methodology to incorporate the available, albeit indirect, experimental data to inform in a more objective and transparent manner the uncertainties associated with the model parameters. This is done in three steps by consolidating and adapting recent developments in the applied statistics literature to:

1. *Analyze and better understand* the inputs/outputs relationship in a computer simulation with uncertain inputs. Sensitivity analysis (SA), in particular global sensitivity analysis (GSA), methods can be used to assist identifying which of the model parameters can be calibrated using the available data. An uncertainty analysis often starts with a large list of input parameters that may and may not be relevant (i.e., influential) to the simulation at hand. A *screening* method can be used to remove the least influential parameters from the list. Afterward, a variance decomposition method is employed to quantitatively analyze the contribution of the remaining influential parameters uncertainty on the prediction uncertainty. Multiple types of data can be measured during experiments in a test facility (e.g., clad temperature, pressure drop, etc.), it might be worthwhile to consider each one of them. Finally, for each of the different types, the analysis is conducted on various derived QoIs, some of which explicitly consider the output as function. By doing so, it is hoped that interesting model behavior with respect to the input parameters perturbation can be revealed.

Section 1.4.1 provides an overview of the wide range of sensitivity analysis (SA) methods in the applied literature, while Chapter 3 presents the details of the selected SA methods and their applications to a TRACE model.

2. *Approximate* the inputs/outputs relationship of a complex computer simulation for a faster evaluation. The step is required as the statistical calibration method adopted in this thesis is computationally expensive (requiring numerous code runs in the order of hundreds of thousands and beyond). This approximation is done through a Gaussian process (GP) metamodel resulting

in a statistical metamodel. The highly multivariate nature of the outputs (time- and space-dependent) is dealt by a dimension reduction technique. Build upon the results of the previous step, only parameters that are identified to be influential are included in the construction of the metamodel.

Section 1.4.2 introduces a broad overview of metamodeling in the literature and Chapter 4 presents the details and the application of Gaussian process (GP) metamodel to a TRACE model.

3. *Calibrate* the physical model parameters against various relevant experimental data. The word *calibrate* carries a disparaging interpretation related to *tweaking*. However, using a Bayesian statistical framework, the aim of calibration is extended to simultaneously quantify the uncertainty of the parameter estimation. The framework includes various sources of uncertainty, which can be modeled probabilistically, including the model bias term. At the end, the parameters of interest will either be given in the form of posterior PDFs conditioned on the data or samples generated from such distributions to conform with the practice of statistical uncertainty propagation widely adopted in the field of nuclear engineering.

Section 1.4.3 provides the practice of Bayesian calibration of computer model from the literature. Chapter 5 the details the formulation of a Bayesian calibration problem for model parameters, the ways to solving it, and an application of it to a TRACE model.

Finally, as the calibration is only conducted using experimental data in a limited set of experimental conditions, it is important to validate the proposed methods by demonstrating the applicability of the results to the simulation of the phenomena in the same facility but in different experimental conditions. That is to propagate the posterior uncertainty of the parameters and to compare the results against experimental data not used in the calibration step.

1.3.3 Scope

Although the proposed set of strategies in this PhD research work can be applicable to the analysis and calibration of any physical model of a system code, it is illustrated by its application on the models of particular importance during simulation of reflooding, i.e., the so-called post-Critical-Heat-Flux (post-CHF) flow regimes. There are several reasons for this emphasis as recognized by the BEMUSE and PREMIUM projects (see Section 1.2.3):

- Reflooding is an important part in the simulation of LWRs transient during LOCA. Modeling reflooding determines the appro-

priate representation of the dynamics of heat transfer phenomena during the effort to rewet an uncovered core. Of paramount interest is the estimation of the time at which the rod can be expected to be rewet as well as the maximum temperature reached prior to rewet. Reflood is a transient with highly coupled hydrodynamic-heat-transfer effects and it challenges the assumption made on the implemented closure laws. Indeed, several reflood experimental programs conducted in SETFs exist. Unfortunately, no orchestrated effort was made so far to consolidate the generated data in general and into the TRACE code in particular.

- The models are adequately complex. It is complex that 4 flow regimes are involved in a single phenomena: multiple sub-models, parametrized with numerous inputs, with multivariate outputs (both time- and space-dependent). But as the source of data is from reflooding SETFs, real plant system (and full scale) effects can be excluded and the ensuing analysis can be concentrated on a limited set of models. In fact, as already pointed out, reflooding SETFs were designed to validate and to calibrate reflood models in system codes.
- Multiple data of various types (pressure, temperature, etc.), taken with different experimental conditions (flow rate, system pressure, etc.), are typically available from experiment within the same facility. As calibration in the present research is conducted using one experimental condition, it is important to validate the calibration results against the data with different experimental conditions albeit from the same experimental facility. Moreover, additional data from other reflooding SETFs are available. This is important for validating the proposed method further and expanding it to calibration against data from multiple facilities.

As such, while it is important to acknowledge that reflood simulation and the associated relevant model (or models) are only parts of a large and complex TH system code, they can provide a representative and relevant illustration on the particulars of analyzing and calibrating the code using experimental data from SETF in general; providing a suitable testing ground for the proposed methods.

The methods and practices of sensitivity analysis, approximation, and calibration of computer model need not be statistical. This thesis, however, focuses on the statistical approach for each of the aforementioned steps. The main reasoning for this choice are twofold: First, statistical methods tend to require fewer assumptions regarding the model complexity. While they may be more computationally expensive than their non-statistical counterparts, they are also easier to set up, with minimal intrusion to the code itself, and subject to less severe dependence on the number of input parameters. Secondly, the ultimate results of the model parameters calibration (i.e., their quan-

tified uncertainties) should be represented in terms of probability. As mentioned previously, this is to conform with the widely accepted practice of statistical uncertainty propagation in the nuclear engineering community.

TRACE code

As a last note, the thermal-hydraulics (TH) system code considered in this thesis is the TRAC/RELAP Computational Engine (TRACE) code developed by the the United States Nuclear Regulatory Commission (USNRC). The main reason to consider solely this particular code is the fact that TRACE is the thermal-hydraulics system code used for the safety analysis of the Swiss NPPs conducted within the Steady-state and Transient Analysis Research for Swiss Reactors (STARS) program [29] at the Paul Scherrer Institut (PSI).

1.4 STATISTICAL FRAMEWORK FOR COMPUTER MODEL SENSITIVITY ANALYSIS, APPROXIMATION, AND CALIBRATION

The set of strategies for sensitivity analysis, model approximation, and calibration presented above constitutes a consolidated statistical framework that will be used in this thesis for quantifying the uncertainty in model parameters of a TH system code. This section presents a broad, and by no means exhaustive, literature review of the strategies used in this thesis. For each strategy, the review first reiterates its main motivation followed by a generic classification and the steps involved before briefly summarizing its applications in nuclear engineering TH, both in the past and more recent times. As will be outlined in Section 1.5, three main chapters of the thesis will be dedicated for each of the proposed strategies detailing the selected methods further and presenting their applications on a TRACE model.

1.4.1 Sensitivity Analysis (SA)

An essential part of model development and assessment is to properly describe and understand the impact of model input parameter variations on the model predictions. SA is an important methodological step in that context [3]. SA is the process of investigating the role of input parameters in determining the model output [30] variation and it seeks to quantify the importance of each model input parameter on the output.

Classifications

Various classifications exist in the literature to categorize SA techniques [30–34]. In the review by Ionescu-Bujor and Cacuci [32, 33], SA techniques are classified with respect to their scope (local vs. global) and to their framework (deterministic vs. statistical). In the review of SA methods by Iooss and Lemaître [30], and the works by Saltelli et al. [34] and by Santner et al. [35], the statistical framework is implicitly assumed, and the classification is based on the parameter space of interest (local vs. global).

Local analysis is based on calculating the effect on the model output of small perturbations around nominal parameter values. Often, the perturbation is done one parameter at a time thus approximating the first-order partial derivative of the model output with respect to the perturbed parameter. The derivative can be computed through efficient adjoint formulation [36, 37] capable of handling numerous input parameters.

Local sensitivity analysis

Besides being numerically efficient, sensitivity coefficients obtained from local deterministic sensitivity analysis have the advantage of being intuitive in their interpretation, irrespective of the method employed [38]. The intuitiveness stems from the equivalence to the derivative of the output with respect to each parameter [32] around a specifically defined point (i.e., nominal parameter values). Thus the coefficients can be readily compared over different modeled systems, independently of the range of parameters variations.

Global analysis, on the other hand, seeks to explore the input parameter space across its range of variation and then quantify the input parameter importance based on a characterization of the resulting output response (hyper-)surface. In the global deterministic framework [32, 37], the characterization is aimed at the identification of the critical points of the system (e.g., maxima, minima, saddle points, etc.). In statistical global methods [34, 39, 40], the characterization is aimed at measuring the dispersion of the output based on variance [41, 42], correlation [43], or elementary effects [44].

Global sensitivity analysis

Due to the different characterizations, the global statistical framework can potentially give spurious results not comparable to the results from the local method as there is no unique definition of sensitivity coefficient provided by different global methods [38]. In some cases, different methods can give different and inconsistent parameter importance ranking [34, 39]. Furthermore, the result of the analysis can be highly dependent to the assumed input parameters probability distributions or their range of variations [33, 37].

Yet, despite the aforementioned shortcomings, the global statistical framework has three particular attractive features relevant to the present study. First, the statistical method for sensitivity analysis is non-intrusive in the sense that minimal or no modification to the original code is required. In other words, the code can be taken as a black box and the analysis is focused on the input/output relationship [34]. This is the case especially in comparison to adjoint-based sensitivity [45, 46], which is a highly efficient and accurate method applicable to a large number of parameters, provided that the code is designed/modified for adjoint analysis.

Global statistical sensitivity analysis

Second, no a priori knowledge on the model structure (linearity, additivity, etc.) is required. This is essential in many cases because depending on the model complexity and for large parameter variations, the linearity or additivity assumption might not hold.

Last, the choice of a statistical framework for sensitivity analysis fits the Monte Carlo (MC)-based uncertainty propagation method widely adopted in nuclear reactor evaluation models [15, 16, 25, 47]. The method prescribes that the uncertain model input parameters (modeled as random variables) should be simultaneously and randomly perturbed across their range of variations. Multiple randomly generated input values are then propagated through the code to quantify the dispersion of the prediction (e.g., peak clad temperature) which serves as a measure of the prediction reliability. Statistical global sensitivity analysis thus complements the propagation step by addressing the follow-up question on the identification of the most important parameters in driving the prediction uncertainty.

*Choosing model
output as a quantity
of interest*

Saltelli et al. [40] emphasized that an analysis using computer simulation should be focused on a specific question the simulation is required to answer as opposed to the analysis of each and every individual model output. This is done through judicious choice of representative quantities of interest (QoIs) that properly substantiate the problem at hand. In particular, computer code output often comes in a form of time series. In such case, Saltelli et al. [34, 39] proposed to derive the relevant QoI from time-dependent output using a predefined scalar function such as the maximum, the minimum, the average, etc. that fits the initial question.

*Function as model
output*

However, in some cases, the whole course of a transient is of primary interest such as in assessing the ability of a model to reproduce the overall dynamics of the simulated system. If the attention is focused on the overall change in shape of the time-dependent output (a shift in the Y-axis, a delay, a distortion, etc.), the descriptions provided by the aforementioned scalar functions might be incomplete and overlook important features of the variation. To tackle this problem, Campbell et al. [48] proposed to represent the functional (time-dependent) output in a certain basis function expansion and to carry out the sensitivity analysis on the coefficients of the expansion. In accordance to such approach, functional data analysis (FDA) popularized by Ramsay and Silverman [49] is useful to reduce the high dimensionality of time-dependent output.

*Developments in
nuclear engineering
application*

Despite these recent developments, there are very few publications on the application of global sensitivity analysis to nuclear thermal-hydraulics evaluation models specifically dealing with time-dependent output. Notable recent examples related to sensitivity analysis for a time-dependent TH problem were the work done by Ionescu-Bujor et al. [50] for reflooding experiment of degraded fuel rods, utilizing adjoint sensitivity method; by Auder et al. [51] for pressurized thermal shock analysis, utilizing statistical methods with emphasis on meta-modeling; and by Prošek and Leskovar [52] for LBLOCA analysis, utilizing Fast Fourier Transform-Based method (FFTBM) and local sensitivity analysis.

1.4.2 Statistical Metamodeling

Many tasks involving computer simulations can be boiled down to making predictions. A *computer experiment*, an experiment using computer simulations, evaluates the output based on different inputs to achieve various objectives. In the aforementioned sensitivity analysis, the objective is to identify the influential inputs that drives the variation in the outputs of the computer simulator. In the forward uncertainty quantification, MC simulation are used to propagate the uncertainty of the inputs to quantify the uncertainty of the simulator prediction using the notion of probability; while in its inverse counterpart, the goal is to identify a region of the input parameter space that is consistent with both the observed data and the assumed prior uncertainty of the inputs. The latter objective, in turn, is related to *optimization* where the goal is to identify particular value of inputs that maximize a certain objective function as computed by the simulator.

Computer experiment

The objectives above are arguably distinct, but they share a common characteristic of involving analyses of outputs from numerous simulator runs. An increasingly more realistic and complex computer simulator, however, often translates into a long running simulation and may have to be evaluated a large number of times due to the complexity of the relationship between high-dimensional inputs and high-dimensional outputs (e.g., non-linearities, interactions). The high computational cost hinders the analysis and the effort to achieve the aforementioned objectives of computer experiment.

Complex simulator, expensive simulator

As a result, having a fast approximating model of a complex simulator is beneficial in conducting a computer experiment and its value was acknowledged by Sacks et al. in their seminal paper [53] and formalized further in several textbooks [35, 54]. The approximating model, while simpler and much faster to evaluate than the *original* simulator, is designed to capture the dominant features of the inputs/outputs relationship of the original complex simulator [55]. Capturing the dominant features allows the approximating model to be used in lieu of the original simulator in the experiment. This approximating model in the literature is referred to as *metamodel*, *surrogate model*, *response surface model*, *proxy model*, or *emulator*.

An approximating model

Nowadays, any of the terms above are used interchangeably and all are used to substitute the original simulator to reduce the computational cost of conducting computer experiments [55–57]. Subtle differences do exist. Thus, it is worthwhile to consider a broad classification of surrogate models and the approaches to their derivations (i.e., surrogate modeling or metamodeling) according to the literature. Surrogate models according to their derivations can be broadly classified in two categories: the data-driven *response surface surrogates* and the mechanistic *reduced-order models* [57].

Classification

Reduced-order model

The reduced order models are perhaps more familiar in the scientific community where a complex physical model is being simplified by putting more stringent assumptions or reducing the numerical resolution while trying to preserve the most important physical processes present in the more complex model. The point neutron kinetics model is an example of a reduced-order model, substituting the more complex 3-dimensional nodal code. A TH system code is also a reduced-order model of the more expensive multi-phase computational fluid dynamics code. When applicable, reduced-order models can be useful as first approximations and didactic tools to build intuition.

*Response surface
surrogates*

The response surface surrogates, on the other hand, make no pretense of preserving the underlying physical process modeled in a complex simulator. It seeks to *emulate* the relationship (i.e., *mapping*) between inputs and outputs of the simulator. The term *metamodel* is used throughout the thesis and exclusively refer to this particular type of surrogate model. The workflow of constructing a response surface surrogate consists of three steps. The first step is to gather the data, that is by running the simulator at limited and selected points across the input parameter space of interest and evaluate its outputs. The selection of such points are known as the *design of experiment* [35, 58]. The second step is to choose an approximating function that emulates well the relationship between the inputs and outputs and *train* this function based on the data. Training a surrogate model involves *fitting* the parameters associated with the selected approximating function. The function is chosen such that it is simpler and faster to evaluate at arbitrary inputs, relative to the original simulator. Finally, a validation step is conducted to assess the quality of the resulting metamodel. These are a typical workflow of constructing a metamodel, though variation exists [57].

*Gaussian process
metamodel*

The surrogate model introduced in the papers of Sacks et al. [53, 59] were GP metamodel. The metamodel was constructed as a tool to interpolate between observed data, that is, between the inputs and outputs of actual simulator runs. Once constructed the output at any arbitrary input point can be predicted faster using the metamodel. This idea was borrowed from a spatial interpolation tool in geostatistics (where the inputs were spatial coordinates) developed by Krige dating back to the 1950s [60] and formalized by Matheron in the 1960s [61]. GP metamodel is arguably the most popular approach to metamodeling and it enjoys a renewed interest due to its application for machine learning [62].

A GP metamodel is a statistical metamodel. It is based on the extension of multivariate Gaussian distribution to a continuous multidimensional input parameter space. Under the Bayesian interpretation, the metamodel assumes a prior probability distribution over functions (i.e., a probability distribution of which each realization is a

function) to initially describe an unknown complex function that underlies the simulator. The observed data (i.e., design of experiment plus the corresponding outputs) is then used to update the prior and learn more about the true underlying function. Though the simulator itself might be deterministic, the limited size of the observed data renders prediction at arbitrary non-observed input *uncertain*. This measure of uncertainty makes a GP metamodel an attractive choice to be incorporated into a model calibration framework where multiple sources of uncertainty are considered. This research adopts GP for constructing a metamodel of a TH system code model as detailed in Chapter 4.

GP metamodel is by no means the only method to construct a data-driven metamodel, though it can be considered as the most popular choice in the literature (Table 1.1). *Response Surface Method* (RSM), originally developed as a technique in the design and analysis of physical experiments [63], has a long history of being adapted to the design and analysis of computer experiments [64–67]. It is mostly based on either linear or quadratic regression (with interaction terms) (see for instance [68], and more recent reviews [56, 69]). In recent times, other methods such as the ones based on artificial neural network [70] and polynomial chaos expansion (PCE) [71, 72] have also gained traction. For comparison, Table 1.1 shows the search hits from Scopus, an online bibliographic database [73], for the different selected metamodeling approaches. Note that the list is not at all exhaustive.

Other metamodeling approaches

Table 1.1: Number of publications related to different metamodeling approaches based on Scopus web search as of Feb. 14. 2017.

METAMODELING APPROACH	SEARCH KEYWORD	ⁱ NUMBER OF PUBLICATIONS	SINCE
Gaussian Process / Kriging	("Gaussian Process OR kriging")	1838	1992
Artificial Neural Network	"neural network"	997	1993
Response Surface Method	"response surface"	947	1977
Polynomial Chaos Expansion	"polynomial chaos"	208	2004

ⁱ (...) AND ("surrogate" OR "metamodel")

Metamodel applications have a long history in nuclear engineering analyses due to the complexity of the simulators and the long-understood importance of quantifying the uncertainty of the predictions. Hence, historically, metamodels (specifically, the response surface method) have been applied for quantifying the prediction uncertainty forward through MC sampling as well as for statistical sensitivity analysis [74, 75]. The range of applications varied from quan-

Developments in nuclear engineering application

tifying the reactor safety margin [13] for a LBLOCA scenario, propagating the uncertainty of fuel rods failure in the core [76] during the same scenario, to the uncertainty and sensitivity analyses of severe accident progressions [77]. In recent times, more advanced metamodels (e.g., GP metamodel) have been applied to more diverse engineering analysis; from the design optimization problem of fuel assembly [78] and spacer grid [79] to the calibration of physical models in TH system code [80] and fuel performance [81] code.

1.4.3 Bayesian Calibration

*Model calibration,
goal and approach*

The objective of model calibration is to increase the agreement between simulation predictions and measurement data by adjusting some of the simulator inputs [3, 82]. Traditionally, calibration is closely related to an optimization problem of an objective function measuring the error between simulation predictions and measurement data (e.g., root-mean-square-error). However, statistical approach to calibration using a Bayesian framework has become a popular practice in the scientific simulation community. Instead of minimizing a measure of error, Bayesian framework treats the uncertainty of the inputs probabilistically and update their prior probability distributions based on the available, albeit uncertain, measurement data. The framework offers flexibility in modeling various sources of uncertainty [83, 84].

*Bayesian framework,
Kennedy and
O'Hagan approach*

Bayesian framework for the calibration of computer simulation model was popularized by the work of Kennedy and O'Hagan [83]. The main goal of the framework is similar to any calibration framework, that is to learn the appropriate values of model parameters and their uncertainties by taking into account different sources of uncertainty based on the observed data. The distinct idea is to acknowledge that a systematic bias between a physics-based simulator and reality might exist and is often not known a priori. If the bias is not modeled properly, the calibration process might overfit the model parameters. That is, the calibrated model parameters will be overly sensitive to the calibration data and thus not applicable for prediction. As such, Kennedy and O'Hagan proposed to model the unknown bias term probabilistically by putting a prior probability distribution on the bias term to be updated simultaneously with that of the model parameters, by using the observed data. The proposed prior distribution is a GP. Due to its popularity, the term *KOH approach* became synonymous to this particular approach of computer model calibration [3, 85, 86]. It is adapted here to deal with the particular problem posed in the PREMIUM benchmark that itself represents a typical problem in nuclear engineering TH analysis, as detailed in Chapter 5.

Bayesian framework for model calibration consists of two main aspects [87]: a *formulation* of a posterior distribution for the model pa-

rameters of interest and the *computation* involving the posterior distribution. Regarding the first aspect, the KOH approach for model calibration, in essence, prescribes a probabilistic model for the data-generating process of the experimental data, incorporating the prediction by the simulator, the uncertain model bias term, and the uncertain model parameters into it. The formulation will eventually results in a posterior probability distribution of the model parameters as presented in Section 1.2.2. Extension and modification to the KOH approach includes dealing with high-dimensional output [88–90], multiple types of output [91], different choices of the model bias term [86], and various simplifications [92–94].

Steps in Bayesian framework

Bayesian framework, formulation

Regarding the choice for the prior distribution, there is a tendency in the modern literature [87, 95] to move away from the notion of *Bayesianism*, which emphasizes specifying the correct prior (if not the *true* prior altogether) so as to guarantee that the resulting posterior will always be true relative to that prior. In a more modern practice of Bayesian statistics, the use of prior is seen in a more pragmatic light, i.e., as a starting assumption that can be changed if not appropriate [96, 97]. Gelman et al. [87] advocates to check if the resulting parameters posterior distributions and the prediction using them make sense and are useful, rather than to check whether the posterior parameter distribution is *true*.

On prior distribution

The second aspect of the Bayesian framework is related to the computations involving the posterior distribution of the model parameters. The formulated posterior distribution in practical problems is multidimensional. Numerically integrating the posterior distribution to summarize a given QoI (see Section 1.2.1) might not be efficient. Traditionally, maximum a posteriori estimates⁸ is often used to describe each of the posterior model parameter values using a single number. It simply requires the maximization of the posterior density function (i.e., its mode). An extension of this, giving description of the shape of the posterior, is done by the so-called *Laplace's approximation* or the *normal approximation* [87, 95, 98, 99]. In this approximation, the distribution of the posterior is approximated as a normal distribution. The mean corresponds to the maximum of the posterior density function, while the variance of the distribution is approximated as the function of the second derivative of the posterior around the mean (i.e., its curvature). Multidimensionality of the posterior distribution is taken into account by using the Hessian matrix. The approximation works well if the posterior distribution is approximately normal (i.e., unimodal, bell-shaped, and linearly correlated).

Bayesian framework, computation

Semi analytical approach

To deal with more generic formulations of the posterior distribution, modern approach to Bayesian computation involves random simulation to directly generate samples from the posterior. This can be seen as an extension of the simulation method for estimating inte-

MCMC, classical samplers

⁸ or maximum likelihood estimates, if the prior is non-informative

gral quantities (the Monte Carlo (MC) simulation), for the case of a complicated sampling distribution not easily sampled from. The idea of MC simulation dates back to the 1940 for solving the problems in neutron transport [100] and statistical mechanics [101]. In the latter, generation of a *Markov chain* by MC simulation paved the way to a generic simulation method applicable to generate samples from any kind of probability distribution; thus the origin of MCMC simulation. Later on, its usage for data analysis in general and Bayesian data analysis in particular were revived in the 1970s and the early 1980s by the generalization of the simulation algorithm by Hastings [102] (i.e., the Metropolis-Hastings (MH) algorithm) and by the invention of a computationally efficient sampler and its application for image reconstruction by Geman and Geman [103] (i.e., the Gibbs sampler).

*MCMC, modern
samplers*

Nowadays, MCMC sampler is the backbone of Bayesian computation [87, 104, 105]. Loosely speaking, its improvement and modern implementations can be broadly classified into three different families: adaptive MH samplers (e.g., [106]), Hamiltonian MC samplers [107], and ensemble samplers [108]. Adaptive MH sampler deals with the adaptation of the algorithm to achieve faster convergence. Hamiltonian MC sampler simulates the movement of a particle in the parameter space as described by the posterior distribution according to the Hamiltonian dynamic. Finally, ensemble sampler uses multiple particles that move together in the input parameter space each of them moving according to the position of the others. Ensemble sampler has a particularly simple implementation, is easily parallelized, and requires minimal tuning [109]. Chapter 5 describes in more detail the basics of MH and ensemble samplers.

*Developments in
nuclear engineering
application*

Unlike the forward uncertainty propagation and statistical sensitivity analysis, the use of Bayesian model calibration is relatively new in nuclear engineering applications [110]. A notable early example was the previously mentioned CIRCÉ [24] for the calibration of model parameters in a TH system code. A recent demonstration on the applicability of the method can be found in Ref. [111]. More recent examples are in-line with the KOH approach but provide extensions to it: for dealing with high-dimensional output of the same type (i.e., time- and space-dependent) with [112] or without [80, 81] explicit treatment of the model bias, and with [113] or without [114] the use of metamodels.

1.5 STRUCTURE OF THE THESIS

This doctoral thesis is organized into six chapters. The description and the application on a thermal-hydraulics simulation of the statistical approaches for sensitivity analysis, statistical metamodeling, and the Bayesian calibration, preceded by a brief review of the TH system code TRACE, the selected phenomenon of interest, and the associated

physical models, constitute the main chapters of the present thesis (see Fig. 1.6). They are bookended by an introductory chapter (this chapter) and a concluding chapter.

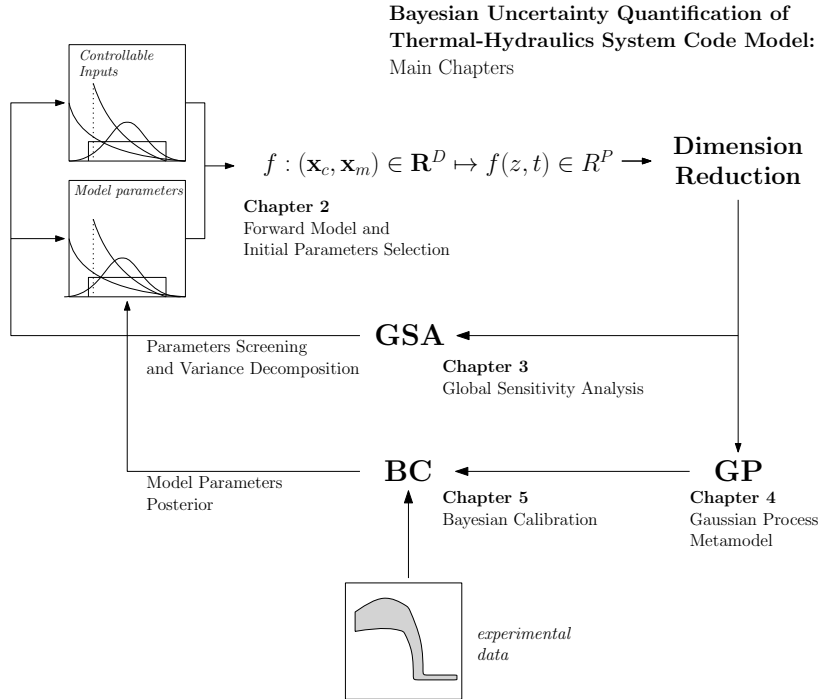


Figure 1.6: The structure of the thesis and its main chapters.

CHAPTER 2 gives an overview of the system thermal-hydraulics code TRACE with an emphasis on its reflood phenomenon modeling and simulation. The chapter also introduces the reflood experiment at the FEBA facility that serves as the experimental basis of this work followed by its modeling in TRACE. This model becomes the running case study in the three subsequent chapters to which the proposed methods are applied. The chapter includes the selection of the initial parameters relevant for reflood simulations and the propagation of their prior uncertainties on the code predictions.

CHAPTER 3 introduces the GSA methods adopted in this thesis with three key underlying ideas. The first idea is to reduce the dimensionality of the input parameters space through parameter screening, while the second is to reduce the dimensionality of the code output space. As the output of the simulation is time-dependent, dimension reduction is carried out while trying to preserve the interpretability of the results. The third and final idea is to investigate, quantitatively, the effect of variation of parameters on the overall time-dependent output variation through variance decomposition. The presented methods are then applied to the TRACE model of FEBA and the results are discussed.

CHAPTER 4 presents an approach to construct a fast surrogate model that approximates the inputs/outputs relationship of a computation-

ally expensive simulator. The theoretical minimum of the method is introduced, before adapting the method for dealing with highly multivariate output via dimension reduction. Afterward, the application of the method to the TRACE model of FEBA is presented and discussed. In the end, a metamodel of the TRACE model is constructed and validated in anticipation of the high cost of the calibration approach presented in the following chapter.

CHAPTER 5 describes the Bayesian calibration and is the last of the main chapters of the thesis. The description of the methods is split into two parts, following the convention in the Bayesian data analysis, the formulation part and the computation part. The formulation of the Bayesian statistical calibration problem (i.e., the posterior) as well as its simplification (the so-called *modularization*) are first introduced. The resulting posterior is potentially complex, i.e., a high-dimensional PDF with highly varying ranges in each dimension. Consequently, the computational part is focused on a simulation-based approach called MCMC to directly generate representative samples useful for downstream analysis (e.g., forward propagation). After that, as in the two previous chapters, the application of the method to the TRACE model of FEBA is presented and the results are discussed. Included in the discussion is the validation of the method based on additional experimental data from FEBA that were not used in the calibration.

CHAPTER 6 brings the thesis to an end. The main findings and accomplishments of the thesis are summarized through chapter-wise summary. Recommendations of future work are then presented.

Four parts of appendices are included in the back of the thesis. They include the governing equations of the TRACE code, additional results of the thesis not presented in the main chapters, the computational tools developed and used in the context of this thesis, and some useful mathematical results and recipes.

Reflood is the last phase of the four canonical phases in the mitigation of LBLOCA in LWRs [115]. An LBLOCA transient starts with a rapid depressurization of the primary coolant circuit (*blowdown*). It is then followed by an initial rejection of emergency coolant water injection into the reactor core due to massive steam flowing out of the boiling core (*bypass*). After an eventually successful injection of the emergency coolant water through the downcomer and into the lower plenum of the reactor pressure vessel (RPV) (*refill*), the *reflood* phase takes place. It refers to the phase of the transient in which the emergency coolant water flows slowly upward through the dried reactor core, quenching the fuel elements along the way, preventing them from being further damaged due to overheating.

This chapter introduces the phenomenology of reflood and its modeling in the thermal-hydraulics system code TRACE. Section 2.1 first presents a quick overview of the TRACE code, including a major simplification taken in the code to describe a complex two-phase flow phenomena in the reactor coolant circuit during accident scenarios. Section 2.2 then describes reflood in LWRs: its importance, phenomenology, and modeling. It introduces several important terminologies used throughout the thesis. The description is specific to the TRACE code and is by no means an exhaustive account on the subject.

The present study is based on the data from a SETF for reflood experiment. The Flooding Experiments with Blocked Arrays (FEBA) facility and the relevant test runs for this study are described in Section 2.3. The modeling aspects of the facility in TRACE is then detailed in Section 2.4.

Section 2.5 deals with the problem of selecting the initial set of input parameters perceived to be influential for the simulation. Afterward, prior uncertainties in the form of PDFs are assigned to the selected input parameters. Those two steps provide the starting point for the sensitivity and uncertainty analyses on the TRACE model of FEBA presented in the upcoming chapters. Section 2.6 presents the propagation of the specified prior uncertainties on the TRACE model to assess the initial level of uncertainty in the predictions before any experimental data is used to update the prior uncertainties. Section 2.7 finally concludes and summarizes the chapter.

2.1 THE THERMAL-HYDRAULICS (TH) SYSTEM CODE TRACE

*Thermal-hydraulics
system code*

A thermal-hydraulics (TH) system code is a computer code used to analyze the TH behavior of NPPs [116]. Its current usage ranges from safety analysis and licensing process of current reactor designs to qualification of a new reactor designs [117, 118]. To that end, the code is designed to be a comprehensive tool capable of simulating wide range of operating conditions, from normal operations, anticipated transients, to accident scenarios foreseen in the operation of NPPs.

*Nuclear reactor
system*

A nuclear reactor system is a complex system of numerous interconnected components, each serving distinct purposes, built with multiple engineered safety features. During a transient, the system might exhibit complex behavior with physical phenomena interacting at vastly different time scales (10^{-1} [s] in a power excursion due to control rod ejection, 10^5 [s] for decay heat removal after successful reactor shutdown) and length scales (10^{-3} [mm] for boiling at sub-channel level, 10^3 [m] for coolant flow in the primary/secondary circuit). Additionally, the engineered safety features are designed for some equipments (such as control rods, valves, pump) to perform safety-related actions. Such equipments, in turn, are controlled by a complex dynamical control system. As such, system code has to take into account these different aspects to properly simulate the thermal-hydraulics behavior of NPPs.

*Component-based
codes*

Indeed, component-based codes, such as TRACE, approach the problem by representing each prominent component in a nuclear reactor system separately. On top of a two-phase fluid dynamics equation solver, system code includes models for steam separator, pump, heat exchanger, valve, pressurizer, and neutron-kinetics as well as comprehensive models for control system to mimic the signal monitoring and component actuation systems in an NPP. System thermal-hydraulics thus distinguishes itself by considering explicitly the geometry, materials, boundary conditions, various interconnecting components, and control systems that constitute an NPP [6].

*Basic
thermal-hydraulics,
two-phase flow in a
heated channel*

However, it can be argued that the modeling of two-phase flow inside a heated channel remains a central part in nuclear thermal-hydraulics analysis which puts an emphasis in the correct prediction of clad temperature evolution during different postulated scenarios (Fig. 2.1). This is also supported by the fact that the majority of operating nuclear power plants is of LWR type, where two-phase flow can be expected to occur during its operation, both in normal operating and accident conditions for boiling water reactors (BWRs) and in accident condition for PWRs.

Flow regimes

The problem of modeling properly the two-phase flow in a heated channel, though much more limited in scope, is by no means trivial. This is due to the fact that in two-phase flow, the morphological configurations of the flow (i.e., *flow regimes*) can vary widely depending

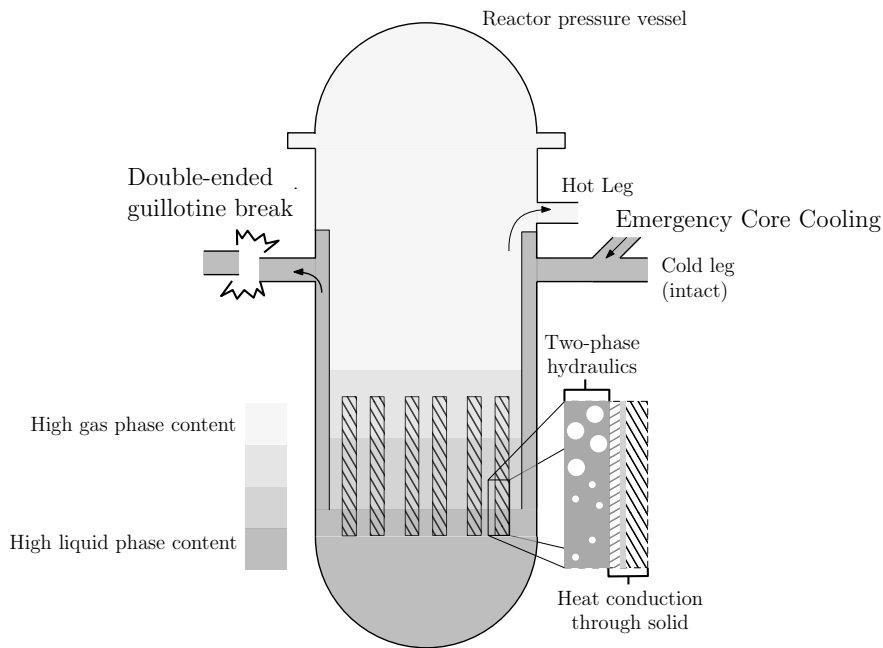


Figure 2.1: Thermal-hydraulics system analysis encompasses many aspects of nuclear reactor system analysis, but the core of the problem for predicting the clad temperature evolution – especially during an accident condition – is to model properly and realistically the coolant flow in a heated channel in steady or transient conditions. Here it is shown a simplified picture of an LBLOCA in a PWR where phase change occurs along the heated channel.

on many flow parameters such as differences in the respective phase density and velocity, as well as in the flow orientation. Different flow regimes implies different interfacial surface structure between the two phases (see Fig. 2.2), which in turn affects the mass, momentum, and energy coupling terms (*transfer relation*) between the phases. At the same time, the interfacial surface and its deformation in an arbitrary flow configuration are not known *a priori* and becomes part of the problem to be simultaneously solved.

The most rigorous approach in describing two-phase flow is by using local instantaneous formulation where a set of partial differential equations describing the conservation of mass, momentum, and energy, is formulated for each phase. The two phases are, in turn, separated by zero-thickness interfacial surfaces. The resulting set of equations fully describes the flow at any given location and at any given time. In addition, the solution of this formulation also respects the *topological constraint* of the flow. The constraint states that only one phase can exist at any given time and location in the flow [119]. This is illustrated in the Fig. 2.3 where a hypothetical probe is put within a two-phase flow and a signal of indicator function $M(\mathbf{r}, t)$ is recorded.

*Local instantaneous
formulation,
topological
constraint*

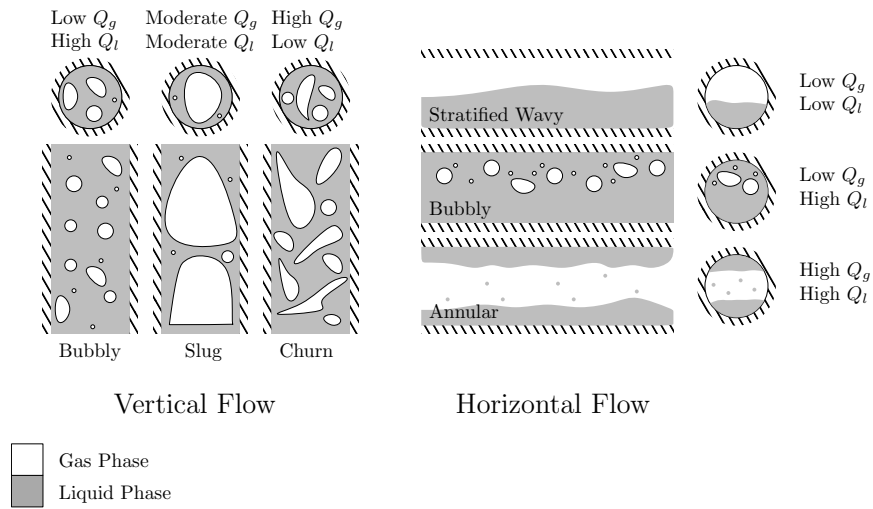


Figure 2.2: Some of the observed flow regimes in vertical and horizontal flow with different superficial liquid velocity, $Q_l = V_l/A$, and superficial gas velocity, $Q_g = V_g/A$, where A is the flow area. The flows of both phases are co-current.

The indicator function $M(\mathbf{r}, t)$ is defined as,

$$M(\mathbf{r}, t) = \begin{cases} 1; & \text{if probe tip is in the gas phase} \\ 0; & \text{otherwise} \end{cases} \quad (2.1)$$

where \mathbf{r} is position; and t is the time. The indicator function defined here is equivalent to the local instantaneous void fraction, which can be interpreted as the probability that the gas phase is present at a given point in space at a given moment [28].

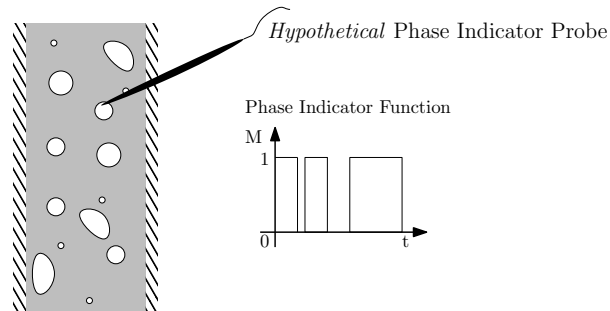


Figure 2.3: Illustration of a hypothetical phase indicator probe inside a channel of a two-phase flow, recording the evolution of the indicator function (Eq. (2.1)) at a given point.

Resolving the motion of interfacial surface

As mentioned, the interfacial surface structure of the flow determines the coupling terms between the two phases. However, this surface and its deformation along the flow are not known *a priori*. As such, the solution of the local instantaneous formulation of two-phase flow requires to solve the motion of interfacial surface. As the time and length scales of the interfacial structure in a two-phase flow of

an arbitrary morphological configuration can vary wildly, the problem of resolving the motion of interfacial surface becomes intractable. Though advances have been made in the area of Computational Fluid Dynamics (CFD) in this regard, the problem remains intractable for the purpose of thermal-hydraulics system analysis¹.

To simplify the intractability problem of resolving the motion of interfacial surface motion in two-phase flow inside a channel, time- and area-average is carried out on the flow. Averaging can be seen as a filtering operation to remove the local temporal and spatial fluctuations (short scale variation) in the flow. The length and duration which define short scale variation are problem specific (that is, at least qualitatively, not longer than the length and time scales of the flow configuration of interest). The volume over which averaging is carried out is referred to either as a *control volume*, a *cell*, or a *node*. It is further assumed that the flow is one-dimensional, in which the flow area changes slowly along the principal direction of the flow. Under these assumptions, a control volume simply corresponds to a cross-sectional slice of the channel and the averaging is based on the flow area instead [119].

Averaging the indicator function both in time and in (a sliced) area gives the *void fraction*,

$$\langle \bar{\alpha} \rangle = \frac{1}{A \Delta t} \int_A \int_t^{t+\Delta t} M(\mathbf{r}, t) \, d\mathbf{r} \, dt \quad (2.2)$$

Following the above formulation, void fraction can be interpreted as the fraction of the control volume occupied by the gas phase [28].

Averaging the flow state variables in time and area and using them to formulate a set of mass, momentum, and energy balance equations describing the fluid dynamics in 1-dimension yield the so-called *two-fluid model* [121]. The model is the state of the art formulation for describing the dynamics of two-phase flow in system codes (including, for example, CATHARE, RELAP5, and TRACE). This model separately treats the transport phenomena of the two phases of fluid resulting in six balance equations which are able to capture phenomena where thermal and mechanical non-equilibrium conditions exist between the two phases, conditions to be expected in a wide range of NPP transients.

Averaging greatly simplifies the description of the complicated interfacial structure between phases in a two-phase flow. This simplification, at the same time, incurs a loss of information regarding energy, momentum, and mass transfers at the local level (between the phases and between each phase and the channel wall, Fig. 2.4). These transfer terms will have to be modeled separately for each distinct flow regime of interest through closure laws [8].

TRACE is the best-estimate system TH code developed by the the United States Nuclear Regulatory Commission (USNRC) as a tool

*Time- and volume
(area)-averaging*

*Time- and
Volume-Averaged
formulation,
two-fluid model*

¹ See [120] for a recent review on the topic.

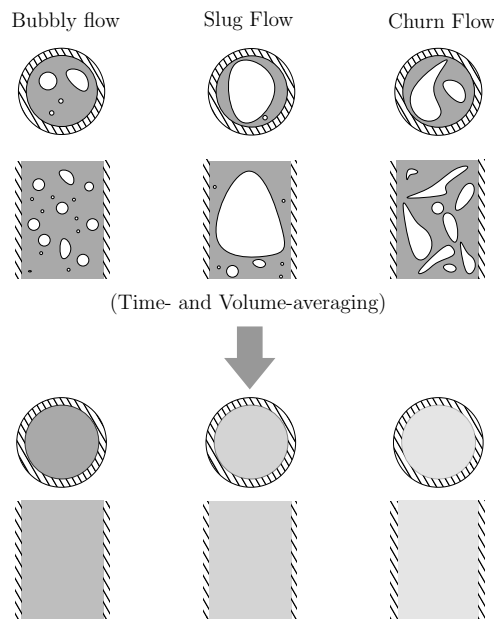


Figure 2.4: Time and volume average carried out on the two-phase flow inside a channel results in a tractable form of fluid dynamics equation, but incur loss of information at the local level, especially when it comes to the interfacial structure between phases and between each phase and the channel wall.

TRACE code

for LWR transient analysis during normal and accident scenarios. Its development is an on-going effort to modernize into a single software package all previous USNRC TH codes that were developed separately for specific reactor types and applications. This ultimately would make the code more versatile for end users and more efficient to maintain for the developer. Appendix A summarizes the final formulation of THE governing equations (time and volume-averaged) in TRACE, of which the complete derivation can be found in [28].

2.2 PHENOMENOLOGY AND MODELING OF BOTTOM REFLOOD

*Reflood phase in
LBLOCA*

As mentioned in the opening of this chapter, reflood phase is the last phase of the four canonical phases in the mitigation of LBLOCA in LWRs, in which emergency coolant water flows slowly upward through the reactor core, quenching the fuel elements along the way. The phase is expected to occur after the refill phase, in which a successful injection of the water through the downcomer and lower plenum of the RPV.

Quenching

Quenching (or *rewetting*) refers to the phenomenon in which a sustainable contact between the liquid phase of the coolant and the hot surfaces of the fuel is re-established. Prior to the quenching, the excessively high surface temperature prevents a stable contact between the liquid phase and the surface, degrading the heat transfer between the two. The maximum temperature for which the liquid might make

a stable contact with the surface is referred to as the *quenching temperature*. In consequence, although the bulk of the flow through the core is liquid, the inability for the liquid to make contact with the surface keeps it at a very high temperature [115, 122].

In BWR, reflood might also occurs by spraying the core from the top resulting in the *top reflood*; while in both BWR and PWR, injection of water downward through the downcomer and upward through the core is termed *bottom reflood*. There are different physical processes associated with the two, such as the fact that in the top reflood there is steam flow from the bottom of the channel pushing back the liquid injection. This thesis is only concerned with the bottom reflood. As the process sets a limiting ability for the emergency coolant to bring about efficient cooling to the fuel elements in the LBLOCA transient, a proper modeling of the physical processes associated with the reflood phase is an important for the safety analysis of LWRs.

A typical mid-height clad temperature evolution in a channel undergoing a bottom reflood (so-called *reflood curve*) can be seen in Fig. 2.5 under a constant coolant injection rate and a constant power boundary condition.

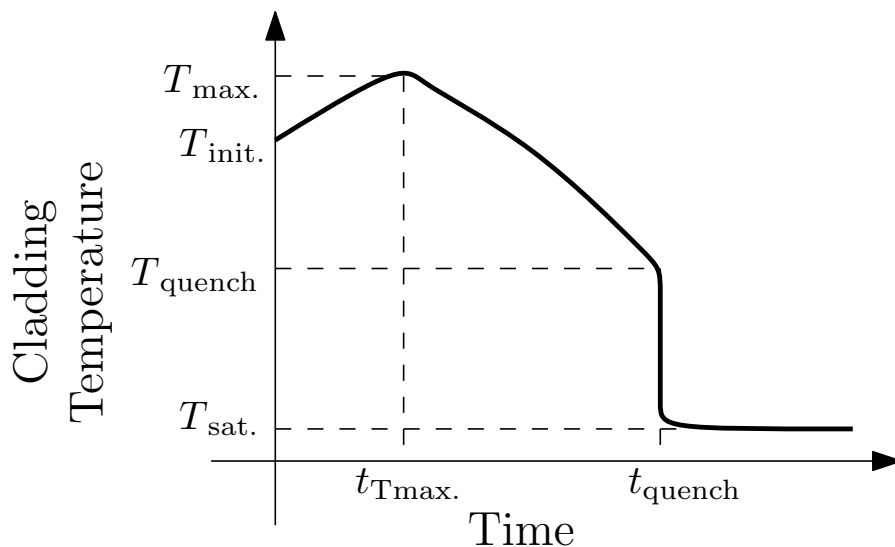


Figure 2.5: A typical clad temperature evolution during constant flooding rate reflooding at mid-height assembly (adapted from [122]). The labels on the both axes are typical QoIs of reflood transient, where the abbreviations max., init., and sat. refer to the *maximum*, *initial*, and *saturation*, respectively.

At the start of the transient (clad temperature at $T_{\text{init.}}$) the channel consists purely of steam. Keeping the power constant increases the clad temperature up until mixture of steam and liquid (droplets) arrives at the location, improving the heat transfer mechanism, and decreasing the temperature ($T_{\text{max.}}$ at $t_{T_{\text{max.}}}$). As the channel keeps undergoing reflood from the bottom, more droplets are available at the location to keep decreasing the clad temperature. Moreover, as the

Top and bottom
reflood

Reflood curve

quenching happens below this particular location, large axial temperature gradient in the clad is present and is further accelerating the heat conduction from the un-quenched part to the quenched part of the clad. Finally, when the temperature of the clad reaches the quenching temperature (T_{quench} at t_{quench}), quenching occurs and stable contact between liquid and the clad can be established. From that point onward, the clad temperature is in equilibrium with the liquid at saturation.

The phenomenological view of the process, adopted by TRACE code [28], is shown in Fig. 2.6a along with the corresponding part in the reflow curve of Fig. 2.6b. The post-critical heat flux (CHF) flow regimes (regimes (2)–(5) in the figure) are in-between two pre-CHF flow regimes, namely nucleate boiling at the bottom and steam convection at the top.

Reflow, phenomenology

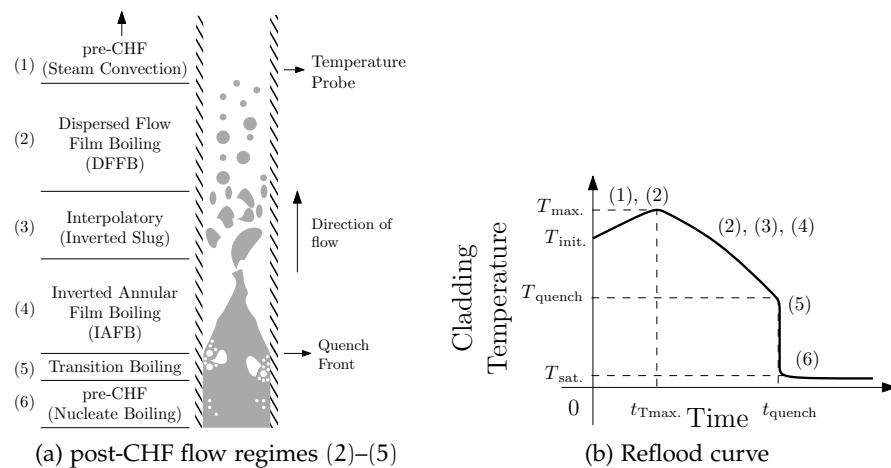


Figure 2.6: Phenomenology of two-phase flow during reflow according to the TRACE code and the corresponding parts of the transient in the reflow curve.

Consider a case of injecting subcooled liquid water with a constant feed rate (i.e., *flooding rate*) into a dry heated channel. At the given location of the temperature probe at the start of the transient, steam convection (regime (2) in Fig. 2.6a) is the dominant heat transfer mechanism and the clad temperature keeps increasing. In TRACE, the steam convection process belongs to the pre-CHF package [28].

As the bottom part of the channel is quenched (the point of quenching on the surface is referred to as *quench front*) three flow regimes can be observed. Far from the quench front, liquid droplets are dispersed and carried away by the bulk steam flow. The flow regime, called dispersed flow film boiling (DFFB), provides an improved heat transfer mechanism from the wall to the fluid as compared to the pure steam convection through direct radiation to the droplets, convection to the droplets and convection to the steam. The droplets provide additional heat sink from the bulk steam flow. The presence of the droplets in the flow also further enhance the turbulence of the steam

Dispersed flow film boiling (DFFB)

flow improving the convection from the wall to the steam flow [122]. The improvement to heat transfer brought by these mechanisms allows the clad temperature to reach a maximum and decrease (i.e., the temperature reversal).

As the quench front progresses upward, not too far from the front a more efficient cooling is provided from morphologically less regular entrained liquid, called ligaments or slugs (regime (3) in Fig. 2.6a). Due to this efficient cooling, the clad temperature keeps decreasing. The slug flow regime is inverted in the sense that the slugs are of the liquid phase. In TRACE these slugs are modeled as prolate ellipsoids. The flow regime itself represents an interpolatory region between the previous DFFB flow regime and the subsequent inverted annular film boiling (IAFB) flow regime [28].

Inverted Slug

Closer to the front, the bulk of the subcooled liquid flow starts to appear in front of the surface. However, a thin vapor film still separates the liquid from the wall and thus prevents an ideal heat transfer to occur. In this so-called IAFB flow regime (regime (4) in Fig. 2.6a), the bulk of the coolant flow in the center of the channel is liquid (i.e., the liquid core). The heat transfer mechanism from the wall is through convection to the vapor film and direct radiation to the liquid core [28, 122].

*Inverted annular
film boiling (IAFB)*

Finally, as quenching becomes imminent and the clad temperature reaches the quenching temperature, the flow regime switches to the unstable *transition boiling*, which literally means the transition between dry wall and wet wall regimes. In TRACE the heat transfer is evaluated based on the look-up table CHF at the particular flow conditions. It results in a very large heat transfer coefficient (HTC) between the wall and the fluid and causes the rapid drop (i.e., quenching) of the temperature (regime (5) in Fig. 2.6a). After quenching, the clad surface is in full contact with the liquid. The clad temperature is in equilibrium with the bulk flow of saturated liquid and the flow regime involves different phenomena, namely nucleate boiling (regime (6) in Fig. 2.6a). In TRACE, as the steam convection, the nucleate boiling process belongs to the pre-CHF package [28].

Transition boiling

2.3 FEBA REFLOOD SEPARATE EFFECT TEST FACILITY

A series of FEBA experiments was conducted in the 1980s at the Karlsruhe Institute of Technology (KIT)² to improve the knowledge of heat transfer mechanism during reflooding, taking into account the effects of spacer grids and flow blockage due to fuel rod ballooning. The data from the facility was also intended to validate the TH models and codes available at the time.

The facility consisted of a test section with a full height 5×5 bundle of PWR fuel rod simulator (Fig. 2.7a) enclosed in a rectangular stainless steel housing (Fig. 2.7b). An approximate cosine power profile was mapped over the height of the fuel rod simulators (Fig. 2.7c). Seven spacer grids were used to provide mechanical support of the fuel rod simulators (Fig. 2.7d).

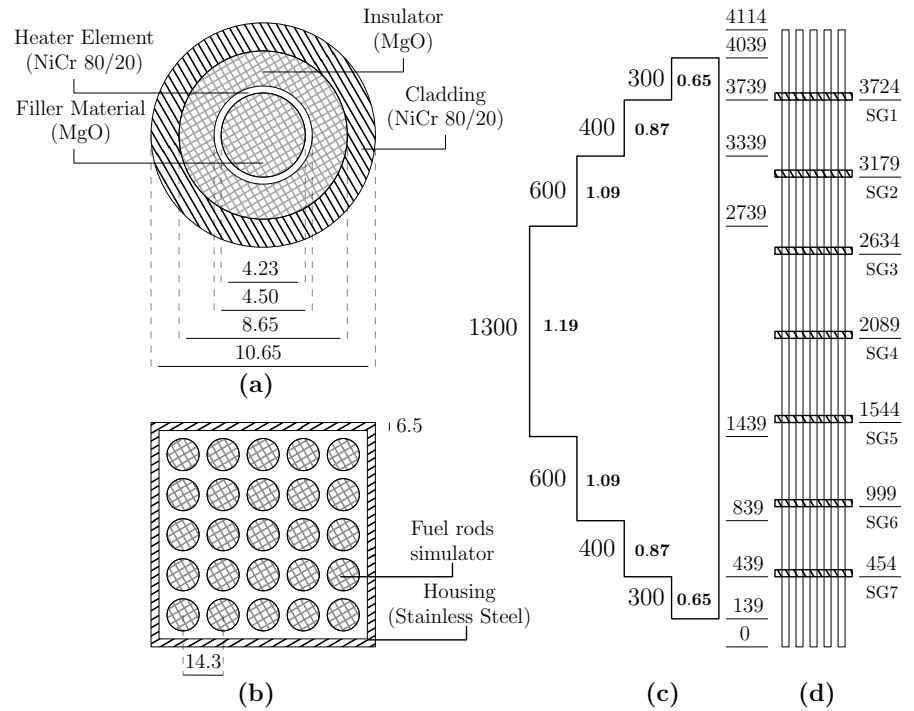


Figure 2.7: (a) The cross section of a fuel rod simulator used in FEBA separate-effect test facility; (b) the cross section of the test section including the rectangular housing; (c) the approximate cosine power profile, numbers written inside the box are the relative power P/P_{avg} ; (d) the location of spacer grids in the test section. All dimensions are in units of millimeters [mm].

During the initialization phase of the experiment, the test section was heated up at low nominal power (200 [kW]) to achieve a specified initial heater rod temperature, with no liquid present in the test section. The transient phase of the experiment was initiated by ramp-

² formerly Kernforschungszentrum Karlsruhe (KfK)

ing up the power according to 120% (ANS³) decay heat power curve while simultaneously injecting subcooled liquid from the bottom of the test section. Several temperature measurements at the outer surface of the heater rods, hereinafter referred to as the clad temperature, were taken at different axial locations during the course of each transient test.

Eight different test series were performed in the FEBA facility. The first two test series (I and II) used two different numbers of spacer grids, seven and six, respectively. The middle spacer grid was removed in test series II to investigate the effect of spacer grids in a reflood transient. The other test series used different flow area blockage sizes at midheight of the test section to investigate the effect of rod ballooning of different sizes. In each test series, combinations of two different inlet liquid velocities and three different system backpressure were imposed.

The present thesis analyzed the experimental data sets from test series I. This particular test series was used as the base experimental setup with all seven spacer grids mounted and no flow area blockage. Different experimental runs corresponding to different experimental conditions of test series I are given in Table 2.1.

Table 2.1: FEBA test series I experimental conditions.

Test No.	System Pressure [bar]	Flooding Rate [cm · s ⁻¹]	Duration of Test [s]
216	4.12	3.81	600
214	4.11	5.77	400
223	2.21	3.82	900
218	2.08	5.81	550
220	6.18	3.85	400
222	6.18	5.78	300

Three types of time series measurement were recorded in the experiment. These included thermocouples to measure the clad temperature (referred to as TC) at eight different axial locations, pressure probes to measure the pressure drop (referred to as DP) at four different axial segments of the test assembly, and a collecting tank measuring the mass of water carried over at the end of the test section (i.e., the liquid carryover, referred to as CO). It should be noted that the collecting tank for measuring the liquid carryover was saturated at 10 [kg] and thus no measurement above that value is available. The axial locations of the thermocouples and the axial segments at which the pressure drop were measured are summarized in Table 2.2. Note

³ American National Standard

that the thermocouple ID was inverted, the increasing number indicated a decreasing elevation. That is, TC1 at the top and TC8 at the bottom of the assembly.

Table 2.2: Locations of the thermocouples and the pressure drop measurements in the FEBA experiment.

Types of Measurement	ID	Axial Locations (or segments) [m]
TC	TC1	4.1
	TC2	3.6
	TC3	3.0
	TC4	2.4
	TC5	1.9
	TC6	1.3
	TC7	0.8
	TC8	0.3
DP	Bottom (Bot.)	0.0 – 1.7
	Middle (Mid.)	1.7 – 2.5
	Top (Top)	2.3 – 4.1
	Total (Tot.)	0.0 – 4.1

The facility specification and the test data are compiled in a series of reports that are available at the KIT library website [123]. The specifications and the data provide a valuable source of information for the TRACE code assessment since the FEBA experiment is not part of the original validation matrix of the code.

2.4 FEBA MODEL IN TRACE

The FEBA facility was modeled using the TH system code TRACE. The TRACE code used was a prototypical version developed, with the support of USNRC, for propagation of uncertainties. The development was a branch from the reference code version v5.0 Patch 3 [28]. The model was developed based on specifications provided within the context of the PREMIUM benchmark [21, 124], following whenever possible the modeling best-practices guidelines for the TRACE code to minimize user effect [125].

The model comprised the following TRACE components:

- A 1-dimensional VESSEL component to model the bundle test section.

- A PIPE component to model the upper plenum of the test section.
- A FILL component to set the inlet flow and inlet temperature boundary conditions.
- A BREAK component to model the outlet pressure boundary condition.
- Two HTSTR components to model the heater rods simulator and non-powered test section housing.
- A POWER component to impose the electrical power boundary condition.

The VESSEL component was nodalized into 28 hydraulic nodes of varying sizes between 60 and 315 [mm]. Both HTSTR components were nodalized into the same number for the coarse axial conduction nodes. However, since a large axial temperature gradient was expected in a reflood transient, the fine-mesh reflood flag in TRACE was enabled. As a result, each of the coarse conduction nodes was divided uniformly in five, yielding a total of 142 axial conduction nodes.

Model nodalization

The main geometrical parameters and experimental conditions used to develop the TRACE input model are summarized in Table 2.3, and the nodalization of the model is illustrated in Fig. 2.8.

2.5 INITIAL SELECTION OF INPUT PARAMETERS

This section presents the selection process of the initial set of uncertain input parameters of the FEBA simulation in TRACE. Afterward, the assignment of the initial (prior) uncertainties of these parameters are presented. This part is closely related to PSI participation in the PREMIUM benchmark thus several reference are made to activities related to that benchmark [126].

2.5.1 Selection of Input Parameters

The selection process for the uncertain input parameters to consider differs depending on the type of parameter. Each of the selected parameters can broadly fall into one of the two following categories:

- Input parameters that are not specific to the TRACE code (e.g., initial and boundary conditions, material thermo-physical properties). This category of parameters is often referred to as the *controllable inputs* of the simulation.
- Input parameters that are specific to TRACE code (e.g., implementation of the two-phase momentum and heat transfer package for reflood condition). This category is often referred to as the *model parameters* of the simulation.

Table 2.3: Geometrical parameters and experimental conditions for the FEBA model in TRACE.

Parameter	Unit	Value
Test section total length	[m]	4.114
Total heated length	[m]	3.9
Flow area	[m ²]	3.901×10^{-3}
Hydraulic diameter	[mm]	13.45
Rectangular housing width	[mm]	78.55
Rectangular housing thickness	[mm]	6.5
Number of rods	[—]	25
Rod outer diameter	[mm]	10.75
Pitch-to-Diameter ratio	[—]	1.33
Number of spacer grids	[—]	7
Spacer grid flow obstruction	[%]	20
Spacer grid axial locations	[m]	0.454, 0.999, 1.544, 2.089, 2.634, 3.179, 3.724
Number of hydraulic nodes	[—]	28 (varying length)
Number of axial nodes	[—]	28 (coarse) 142 (fine)
Inlet liquid temperature	[K]	312
Inlet flow velocity	[cm · s ⁻¹]	see Table 2.1
System backpressure	[bar]	see Table 2.1

*Controllable inputs
selection*

The selection of parameters belonging to the controllable inputs category simply corresponds to the parameters recommended by the benchmark organizers and employed by most participants [19]. The 13 selected parameters of this category are listed in Table 2.4.

On the other hand, the selection of the model parameters specific to the TRACE code is challenging due to the fact that TRACE is a relatively recent code (in comparison with codes like RELAP5, ATHLET, or CATHARE). In essence, the code has been developed from different variants of the TRAC codes for different reactor types (TRAC-BF1, TRAC-P) to result in a single consolidated code applicable to both PWR and BWR. Contributing to that difficulty is the fact that TRACE is currently undergoing significant developments and improvements, including modifications to the two-phase closure models for momentum and heat transfers. Consequently, the tasks of selecting the model

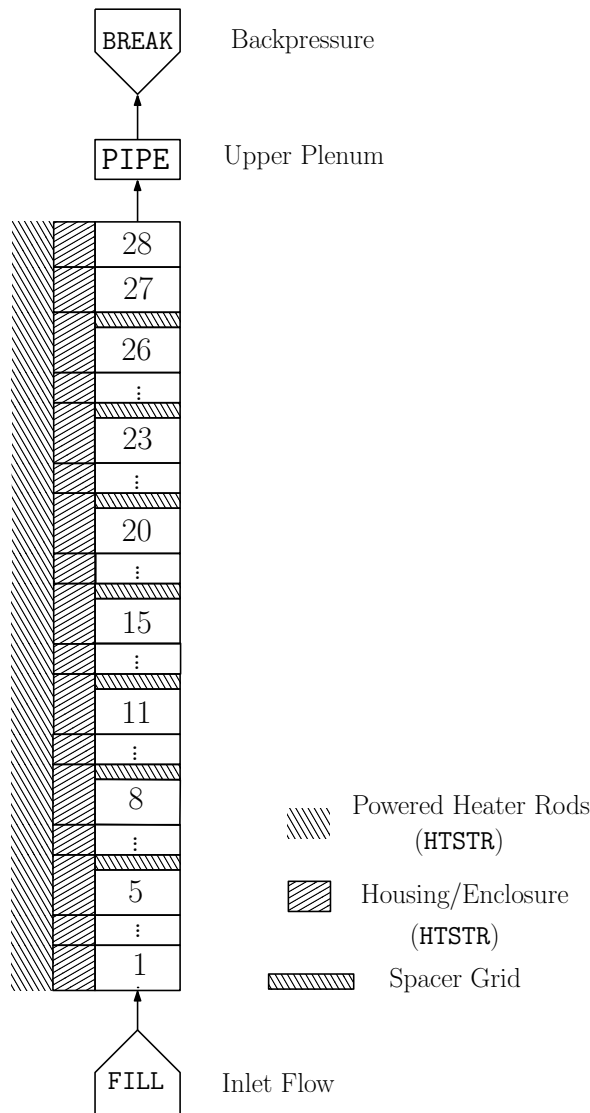


Figure 2.8: Nodalization of the FEBA experimental facility in TRACE.

parameters and later their prior uncertainties are more difficult than for a more established codes.

To overcome this issue, the following principles have been followed to select the model parameters:

Model parameters selection

1. The selection has been focused on the physical models in the post-CHF package of the TRACE code (including the reflood models). Specifically, these are models for the IAFB and DFFB flow regimes [28].
2. Models related to spacer grid are also included as they are known to have a significant impact on reflooding [127].
3. Parameters related to the minimum film boiling temperature and transition boiling should be selected, since they have (by model construction) an impact on the time of quenching.

Table 2.4: Selected TRACE input parameters (controllable inputs), their perturbation factors and their range of variations.

No.	Parameter ID	Description	Distribution	Range of Variation	Nominal Value	Mode of Perturbation
1	breakP	Outlet pressure	Uniform	[0.90, 1.10]	1.0	Multiplicative
2	fillT	Inlet water temperature	Uniform	[-5.00, +5.00]	0.0 [K]	Additive
3	fillV	Inlet water velocity	Uniform	[0.90, 1.10]	1.0	Multiplicative
4	pwr	Heater rod power	Uniform	[0.90, 1.05]	1.0	Multiplicative
5	nick	Conductivity (Nichrome)	Uniform	[0.95, 1.05]	1.0	Multiplicative
6	nicCP	Specific heat (Nichrome)	Uniform	[0.95, 1.05]	1.0	Multiplicative
7	nicEM	Emissivity (Nichrome)	Uniform	[0.90, 1.00]	0.95	Substitutive
8	mgok	Conductivity (MgO)	Uniform	[0.80, 1.20]	1.0	Multiplicative
9	mgocP	Specific heat (MgO)	Uniform	[0.80, 1.20]	1.0	Multiplicative
10	vesEps	Wall roughness	Uniform	$[6.10 \times 10^{-7}, 2.44 \times 10^{-6}]$	1.5×10^{-6} [m]	Substitutive
11	ssk	Conductivity (stainless steel)	Uniform	[0.95, 1.05]	1.0	Multiplicative
12	ssCP	Specific heat (stainless steel)	Uniform	[0.95, 1.05]	1.0	Multiplicative
13	sSEM	Emissivity (stainless steel)	Uniform	[0.56, 0.94]	0.84	Substitutive

Additionally, as a common principle, the selected models and their parameters are perturbed by means of perturbation factors (detailed below) at the highest-possible level of the structure of these models. Different codes share similarity in representing major flow regimes (high-level) but might differ in the constituents models (i.e., sub-models) for each flow regime (lower-level). Focusing on the higher-level implementation of the models allows, to some extent, to use reference uncertainty information obtained from codes other than TRACE.

In accordance with the first selection principle above, a set of 10 high-level parameters has been selected (five for each flow regime). Specifically, for each flow regime: the wall-fluid HTC, the liquid-interface HTC, the vapor-interface HTC, the wall-fluid drag coefficient, and the interfacial drag coefficient.

Following the second principle, two additional parameters have been selected: the spacer grid pressure loss coefficient model from Yao, Loftus, and Hochreiter as well as the grid convective heat transfer enhancement model from Yao, Hochreiter, and Leech (see [28] pp. 425–429 and [128]). These perturbations on the parameters are applied to all seven spacer grids at once.

Lastly, from the third principle, the quench temperature parameter in TRACE and wall-fluid HTC for transition boiling (see [28] pp. 293–299) have been added to the list of uncertain input parameters.

In the end, 14 parameters are selected and are summarized in Table 2.5, yielding a total number of 27 uncertain input parameters.

2.5.2 Perturbation Factors

The nominal values of the selected input parameters of the TRACE FEBA model are varied by means of perturbation factors. These perturbation factors are modeled as random variables following a predefined PDF detailed in the next section, from which a set of samples of input parameters values can be generated.

For a given sampled perturbation factor, one of three modes of perturbation is possible: *additive*, *multiplicative*, and *substitutive*. In the additive mode, the sampled perturbation factor is added to the nominal parameter value of the TRACE model. In the multiplicative mode, the sampled perturbation factor is multiplied by the nominal parameter value. Finally, in the substitutive mode, the sampled perturbation factor directly substitutes for the nominal parameter value. The mode of perturbation for each selected input parameter are listed in last column of Table 2.4 and Table 2.5.

A tool is developed in the Python programming language to assist in automatically pre-processing, executing, and post-processing numerous TRACE simulations of the FEBA model based on a set of sampled input parameters values. The tool, `trace-simexp`, is detailed in Appendix C.2.

Perturbation factor

Modes of perturbation

`trace-simexp`

Table 2.5: Selected TRACE input parameters (model parameters), their perturbation factors and their range of variations

No.	Parameter ID	Description	Distribution	Range of Variation	Nominal Value	Mode of Perturbation
14	gridk	Spacer grid Δp coefficient	Uniform	[0.25, 1.75]	1.0	Multiplicative
15	gridHT	Spacer grid HTC enhancement	Log-Uniform	[0.50, 2.00]	1.0	Multiplicative
16	iafbWHT	Wall HTC (IAFB)	Log-Uniform	[0.50, 2.00]	1.0	Multiplicative
17	dffbwHT	Wall HTC (DFFB)	Log-Uniform	[0.50, 2.00]	0.0	Multiplicative
18	iafbLIHT	Liquid-interface HTC (IAFB)	Log-Uniform	[0.25, 4.00]	1.0	Multiplicative
19	iafbVIHT	Vapor-interface HTC (IAFB)	Log-Uniform	[0.25, 4.00]	1.0	Multiplicative
20	dffblIHT	Liquid-interface HTC (DFFB)	Log-Uniform	[0.25, 4.00]	1.0	Multiplicative
21	dffbvIHT	Vapor-interface HTC (DFFB)	Log-Uniform	[0.25, 4.00]	1.0	Multiplicative
22	iafbIntD	Interfacial drag (IAFB)	Log-Uniform	[0.25, 4.00]	1.0	Multiplicative
23	dffbIntDr	Interfacial drag (DFFB)	Log-Uniform	[0.25, 4.00]	1.0	Multiplicative
24	iafbwallDr	Wall drag (IAFB)	Log-Uniform	[0.50, 2.00]	1.0	Multiplicative
25	dffbwallDr	Wall drag (DFFB)	Log-Uniform	[0.50, 2.00]	1.0	Multiplicative
26	transHTCwallSV	Wall HTC (Transition boiling)	Log-uniform	[0.50, 2.00]	1.0	Multiplicative
27	tQuench	Quenching temperature [K]	Uniform	[-50.0, +50.0]	0.0 [K]	Additive

2.5.3 *Prior Uncertainty Quantification*

The uncertainties associated with the controllable inputs were taken directly from the recommended value of the PREMIUM benchmark and the list can be found in Table 2.4. As for the model parameters, the uncertainty ranges has been determined following the available literature on uncertainties in physical models for LBLOCA.

The main sources of information consisted of Ref.[129] and Ref. [130], which included uncertainty information for the closure models of the system codes TRAC-PF1/MOD1 and ATHLET-Mod2.1 (Cycle B), respectively. Furthermore, prior experience and knowledge of the closure model uncertainties of the CATHARE2 code (V1.3L_1, Rev.5) have been used [126, 131]. Ref. [131] accounts for an analysis by IPSN⁴ of the uncertainty quantification method “Méthode Déterministe Réaliste” for the PWR LBLOCA which was proposed by EDF⁵ and was evaluated in 2000. The high-level implementation of the perturbation factors for the uncertainty analysis in the post-CHF closure models, allowed information from different codes to be extracted for the initial estimate. This approach was deemed adequate in the context of the determination of the prior PDFs.

To simplify the quantification of the prior uncertainties further, the PDFs of the multiplication factors were assumed to follow symmetric bounded uniform and log-uniform distributions with the nominal parameter value equals to the median value. For the log-uniform distribution the form $[2^{-n}, 2^n]$ was assumed, where n is an integer. All model parameters that were a priori deemed to be important were assumed to follow log-uniform distributions.

The ranges of the parameters (i.e., their minimum and maximum), were chosen to cover range of similar parameters available in Refs. [129, 130]. Though this at times resulted in the selection of large bounds, they were deemed acceptable following a verification study against the nominal predictions. The verification heavily relied on engineering judgment via visual inspection of the width of the prediction uncertainty bands to decide if such bands were indeed reasonable⁶. The approach is admittedly imprecise, but is intentional as it avoids underestimating the prior uncertainty range of influential model parameters.

Table 2.5 lists the results of the prior uncertainty quantification of the selected model parameters. Note that all 27 input parameters considered are a priori independent.

⁴ Institut de protection et de sûreté nucléaire (IPSN)

⁵ Electricité de France

⁶ The loosely defined notion of “reasonable” in this case is similar to the notion of “behavioral vs. non-behavioral” prediction in hydrology modeling [2].

2.6 PROPAGATION OF THE PRIOR UNCERTAINTIES

The quantified prior uncertainties of the input parameters are propagated through the TRACE model of FEBA to assess (and verify) the prior level of prediction uncertainties. Independent samples are generated from the prior PDFs of the 27 selected input parameters (Tables 2.4-2.5) and the TRACE model of FEBA is run using the sampled parameters values.

Figs. 2.9a, 2.9b, and 2.9c show the nominal TRACE predictions (i.e., the prediction with the nominal values of the input parameters) in comparison with the experimental data for FEBA test No. 216 for three selected outputs of different types: the clad temperature TC at the mid-height assembly, the pressure drop DP of the middle axial segment, and the liquid carryover CO (up to the saturation of the collecting tank at 10 [kg]), respectively.

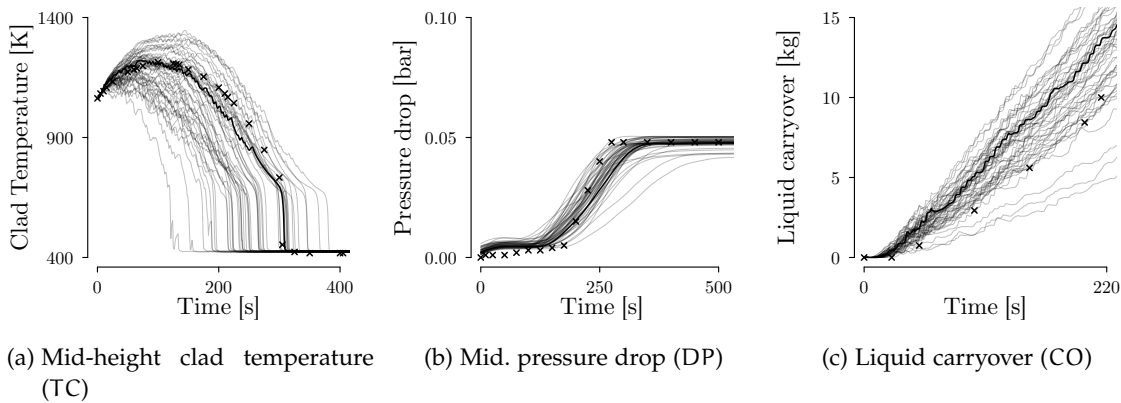


Figure 2.9: Nominal TRACE predictions (thick lines) for FEBA test No. 216 in comparison with the experimental data (crosses) for three selected outputs. The thin lines in each panel indicate the predictions from 50 selected realizations of the uncertainty propagation of the input parameters.

The comparison between the nominal TRACE predictions and the corresponding experimental data for the clad temperature and the pressure drop are satisfactory. TRACE seems to capture all the important features of the transient in FEBA test No. 216. That is, TRACE predicts well the behavior of the reflow curve during the transient; while for the DP output TRACE predicts well the behavior of channel flooding. Note that in Fig. 2.9b the transient between the two equilibrium values indicates the flooding of the channel between axial level 1.7 and 2.3 [m] from an initial pure steam flow (low pressure drop) to mixture (rising) and eventually a pure liquid flow (higher pressure drop). Note also that Fig. 2.9a is the prediction at the axial level 1.9 [m], a level within the axial segments of the pressure drop. On the other hand, there is a strong apparent bias (over-prediction) of the TRACE predictions with respect to the liquid carryover.

The thin lines plotted in each panel of Fig. 2.9 indicate the predictions from 50 selected realizations of the uncertainty propagation. Particularly with respect to the clad temperature output, the predictions exhibit large variations both in terms of amplitude (vertical) and phase (horizontal). Specifically for the latter, the timing of important events like the time of quenching varies significantly across realizations. Moreover, for all the three outputs shown in the figure, the experimental data seems to be within the parametrization of TRACE according to the assumed prior uncertainties for the parameters.

Figs. 2.10, 2.11, and 2.12 show the complete results of the uncertainty propagation for the three types of output for FEBA test No. 216 based on 1'000 samples of input parameters values. The prediction uncertainty bands plotted in each panel of the figures refer to the pointwise symmetric 95% probability. That is, they are constructed based on the intervals between the 2.5-th and 97.5-th percentiles of each output type at each time step. Similar plots showing the results of all the 6 FEBA tests are given in Appendix B.1.

Fig. 2.10 shows the uncertainty propagation results for the clad temperature TC output at all eight axial levels. As observed, for each axial level, the experimental data is well enveloped within the wide prediction uncertainty bands. The uncertainty band becomes wider starting from the start of the transient up to the time of quenching. Furthermore, the uncertainty bands also become wider for the TC predictions moving from the bottom to the top of the assembly. Lastly, as observed, the nominal TRACE predictions tends to have larger discrepancy with the experimental data above the mid-height assembly. Although the time of quenching at all axial levels are well predicted, the TC predictions above the mid-height assembly are underestimated during the transient up to the time of quenching.

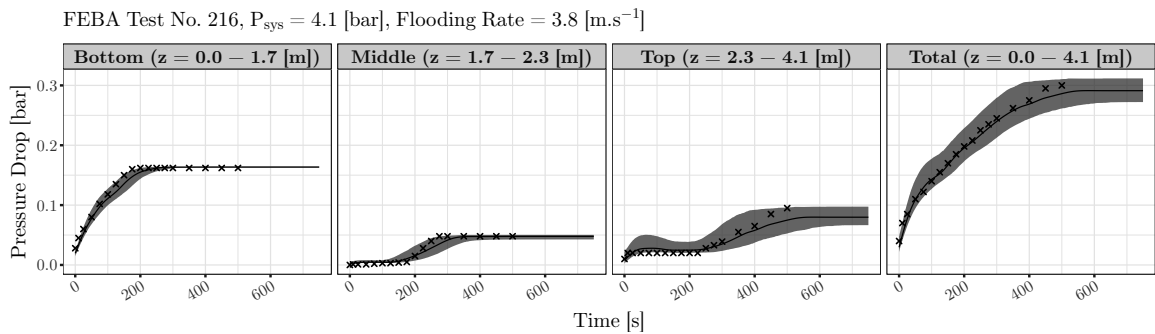


Figure 2.11: Propagation of the 27 input parameters prior uncertainties on FEBA test No. 216 for the pressure drop output (DP). The uncertainty bound corresponds to the symmetric (95%) probability; solid lines and crosses indicate the simulation with the nominal parameters values and the experimental data, respectively.

Fig. 2.11 shows the uncertainty propagation results for the pressure drop DP output at all four axial segments. The plots in the figure

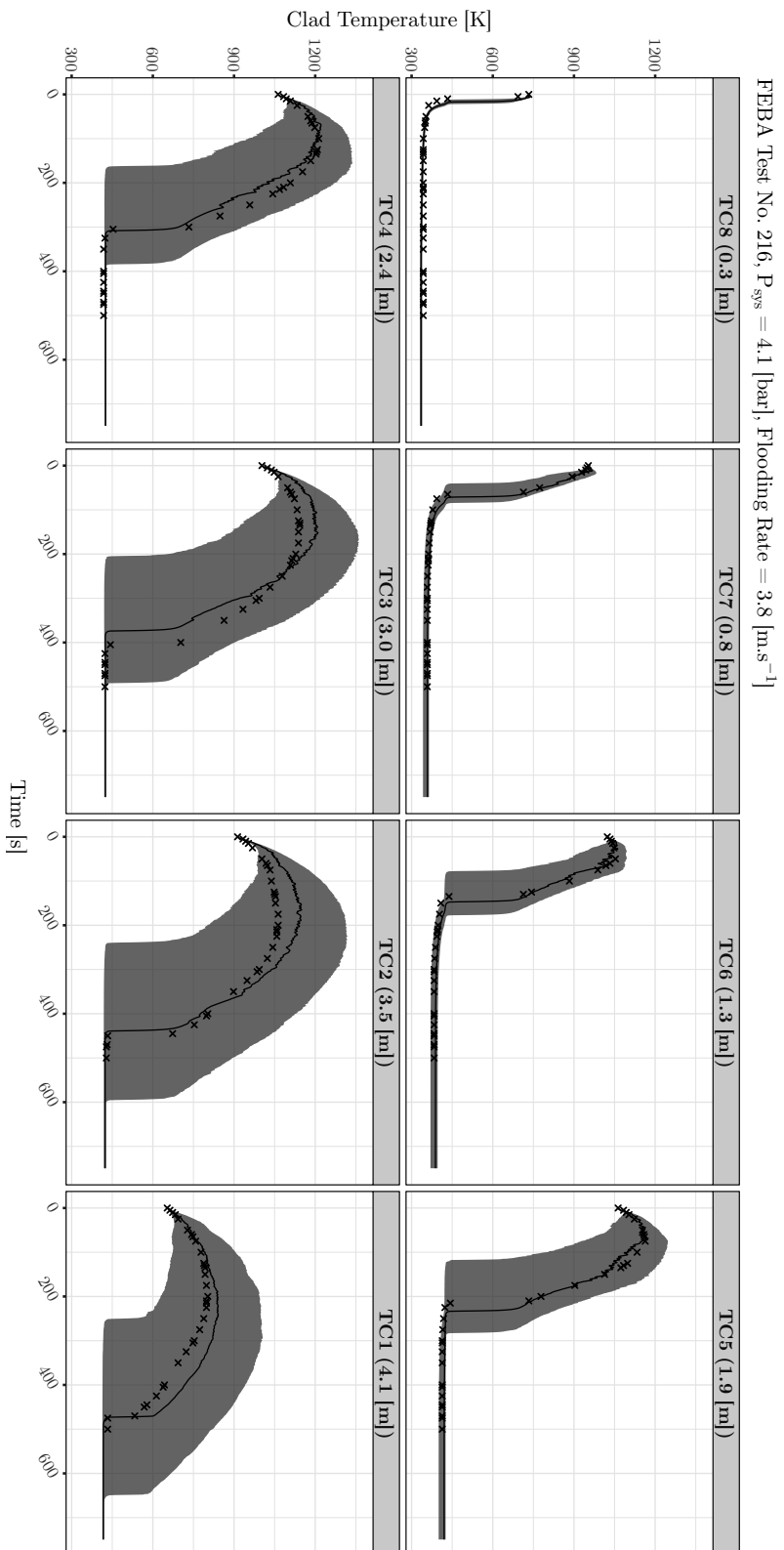


Figure 2.10: Propagation of the 27 input parameters prior uncertainties on FEBA test No. 216 for the clad temperature output (TC). The uncertainty bounds correspond to the symmetric (95%) probability; solid lines and crosses indicate the simulation with the nominal parameters values and the experimental data, respectively.

shows the same pointwise symmetric 95% probability of the prediction uncertainty as before. As observed, there is no major discrepancy between the TRACE predictions and the experimental data and the uncertainty bands cover the experimental data well, especially during the transient (i.e., the ramp between two equilibrium values).

Finally, Fig. 2.11 shows the uncertainty propagation results for the liquid carryover CO output. As mentioned, although there is a large discrepancy between the nominal TRACE prediction and the experimental data, the prediction uncertainty covers the experimental data. In particular, the propagation of the prior input parameters uncertainties results in large band that is skewed toward the lower values of the predictions uncertainties.

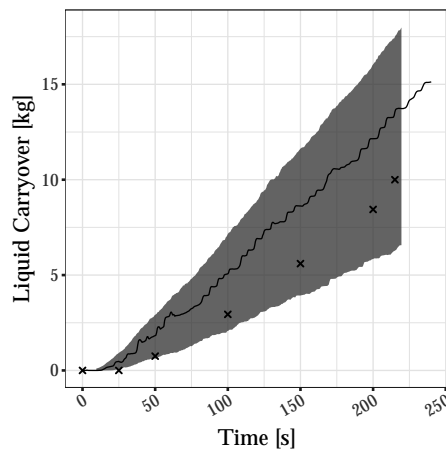


Figure 2.12: Propagation of the 27 input parameters prior uncertainties on FEBA test No. 216 for the liquid carryover output (CO). The bound corresponds to the symmetric (95%) probability; solid lines and crosses indicate the simulation with the nominal parameters values and the experimental data, respectively.

2.7 CHAPTER SUMMARY

The physical model of the TH system code of interest in the present doctoral research has been presented in this chapter. The reflood phenomena and its modeling with the TRACE code were presented.

The FEBA SETF for reflood experiment was described and modeled using the TRACE code. The simulation of the selected reflood experiment using the TRACE model with nominal parameters values gave no indication of major deficiency with respect to important outputs.

A set of 27 initial input parameters, each of which either belongs to the controllable input or model parameter category, have been selected. The justification for the selection was given along with the specification of the prior uncertainties associated with the parameters. The specification of the uncertainties was admittedly imprecise, but deemed adequate for the prior uncertainties. These priors were

then propagated through the TRACE model of FEBA. As expected, the prediction uncertainty bands for all types of output were found to be very wide but at the same time covering all the experimental data points.

The set of methods presented in the next three chapters builds upon the results of this chapter. The experimental data and the TRACE model become the basis for the applications of the methods proposed in this thesis. In Chapter 3, the importance of each selected input parameter is verified in a quantitative manner via SA. The assessment will serve as the basis for parameter screening to reduce the size of the problem. In Chapter 4, a fast approximation of the TRACE model of FEBA is developed to alleviate the computational burden of evaluating the TRACE model numerous times. Finally, in Chapter 5, the selected model parameters are calibrated against the experimental data of FEBA test No. 216, which results in an a posteriori quantification of the parameters uncertainties.

SENSITIVITY ANALYSIS: UNDERSTANDING MODEL INPUT/OUTPUT RELATIONSHIP UNDER UNCERTAINTY

As mentioned in the introduction, describing and understanding properly the impact of model parameters variations on a model prediction are an essential part of the model development and assessment. This chapter presents the application of global and statistical SA to analyze the FEBA model in TRACE in order to investigate the effects of the input parameter variations.

After first introducing in Section 3.1 the notational convention used in this chapter, the proposed methodology is presented. The methodology leverages various developments in global sensitivity analysis (GSA) and functional data analysis (FDA) methods and follows three key underlying ideas.

The first idea, presented in Section 3.2, is to reduce the dimensionality of the output space while preserving the interpretability of the results by utilizing techniques derived from FDA [49]. Section 3.3 introduces the second idea, which is to reduce the dimensionality of the input parameter space through screening analysis using the Morris method [44, 132]. The third and final idea is to investigate, quantitatively and in more detail, the effect of variation of parameters on the overall time-dependent output variation. This is done through variance-based SA using the Sobol'-Saltelli method [41, 133], which is presented in Section 3.4.

The methods are then applied to analyze the FEBA model in TRACE to understand better its inputs/outputs relationship under the assumed uncertainty on its input parameters. The results are presented and discussed in Section 3.6. Finally, Section 3.7 closes the chapter with a summary.

3.1 STATISTICAL FRAMEWORK

The methodology for SA presented in this work belongs to the category of statistical framework, a term attributed to Cacuci and Ionescu-Bujor [32] or simply the global method, following the terminology from Saltelli et al. [39]. Within this framework, sensitivity measures of a parametrized model are obtained by post-processing the collection of model outputs obtained from multiple model evaluations at different points in the input parameter space according to a certain experimental design. As such, the model itself can be considered as a black box, and the input parameters are modeled as random variables

equipped with a joint PDF. The specification of a joint PDF allows for the generation of an experimental design.

Consider the following mathematical model used as a template for the rest of the present chapter:

$$y(t) = f(t; \mathbf{x}), t \in [t_a, t_b] \quad (3.1)$$

where $y(t)$ is the scalar output at time t from a deterministic function f ; and \mathbf{x} is the input parameter vector in D -dimension, i.e., $\mathbf{x} = (x_1, x_2, \dots, x_d, \dots, x_D)$. It is customary to assume, for generality, that the input parameters are normalized between $[0, 1]$ (i.e., $\mathbf{x} \in [0, 1]^D$).

Let \mathbf{DM} be an experimental design matrix of size $N \times D$, where N is the number of samples. Each row in the matrix represents a point in the D -dimensional input parameter space. The model is then evaluated at each of these N points by using a simulation code that results in a matrix of discrete-time outputs of size $N \times \frac{t_b - t_a}{\Delta t}$ where Δt is the time-step size,

$$\mathbf{Y} = \begin{pmatrix} y_1(t_a) & \cdots & y_1(t_i) & \cdots & y_1(t_b) \\ \vdots & & \vdots & & \vdots \\ y_n(t_a) & \cdots & y_n(t_i) & \cdots & y_n(t_b) \\ \vdots & & \vdots & & \vdots \\ y_N(t_a) & \cdots & y_N(t_i) & \cdots & y_N(t_b) \end{pmatrix} \quad (3.2)$$

where $y_n(t_i) = y(t_i; \mathbf{x}_n)$ is the model output at time t_i evaluated using \mathbf{x}_n , the input parameter vector at the n -th row of \mathbf{DM} .

Based on this general description of a time-dependent model output, the next three sections will outline the main components of the proposed SA methodology.

3.2 DESCRIBING VARIATION OF TIME-DEPENDENT OUTPUT

Ramsay and Silverman [49] popularized FDA, which refers to statistical analysis of data that are functions. The main assumption of FDA, as opposed to a more conventional multivariate analysis, is that data present sufficient smoothness, defined by existence of derivatives up to a certain order. Another distinguishing feature of FDA, as opposed to time-series analysis or spatial statistics, is the availability of numerous replications of such data (i.e., set of functions) produced by the same or similar underlying process. The role of FDA in this work is to describe the overall variation within the data set \mathbf{Y} using a reduced set of scalars and functions obtained from principal component analysis (PCA). The functions remain the same for all of the data set, while the scalars vary as function of the sample. Key here is that the required number of scalars is much smaller than the size of \mathbf{Y} and, in turn, can be used as the QoI for SA.

3.2.1 Functional Output Representation

The assumption of continuity within a practically discrete data set (such as the numerical code output of Eq. (3.2)) is made explicit through a functional representation. The recommended representation is through a linear combination of basis functions [49]. This thesis adopts the B-spline basis function [134] expansion because of its flexibility [135, 136] and the availability of its implementation in open numerical libraries [137].

Within this framework, a function can be written using basis function expansion as,

$$y_i(t) = \sum_{k=1}^K c_{ik} \cdot \phi_k(t); \quad i = 1, 2, \dots, N \quad (3.3)$$

where K indicates the number of basis functions; $\phi_k(t)$ is the k -th basis function; and c_{ik} ; $k = 1, \dots, K$ are the basis coefficients for curve i . These coefficients are fitted to the data set to construct curve i , with or without smoothing.

A B-spline is constructed using piecewise polynomial (*spline*) connected at selected points in the domain called *knots*. Let $\tau = \{\tau_k; k = 0, 1, \dots, L\}$ be a sequence of knots, i.e., an ordered set of non-decreasing numbers that divide the function domain into L sub-intervals. Within each sub-interval, a piecewise polynomial of degree p is defined. At the interconnection (knots), adjacent polynomials are continuous with their derivatives up to $p - 1$ matching up. In other words, the spline is $p - 1$ differentiable at the knots.

The basis functions in the B-spline system are determined by the degree of the polynomial p and the knot sequence $\tau = \{\tau_k; k = 0, 1, \dots, L\}$. However, per definition, there is no data point available on the left (right) of the leftmost (rightmost) knot. As such, there is no differentiability (smoothness) condition to uphold at the boundaries and the resulting system of equations are underdetermined. To resolve this issue, the endpoints can be repeated p times and augmented into both ends of the knot sequence such that

$$\tau_{-p} = \dots = \tau_0 \leq \tau_1 \leq \dots \leq \tau_L = \dots = \tau_{L+p} \quad (3.4)$$

This procedure will only reduce the order of continuity of the p outer left and p outer right basis functions of the domain such that the system of equations becomes fully determined. As a result, the augmented knot sequence becomes

$$\tau = \{\tau_k; k = -p, -p + 1, \dots, 0, 1, \dots, L + p\} \quad (3.5)$$

The B-spline basis functions of degree p can then be defined recursively on the augmented knot sequence using de Boor - Cox formula as follows,

$$\begin{aligned}
 B_k^0(t) &= \begin{cases} 1, & \tau_k \leq t < \tau_{k+1} \\ 0, & \text{otherwise} \end{cases} \\
 B_k^p(t) &= \alpha_k^p(t) B_k^{p-1} + [1 - \alpha_{k+1}^p(t)] B_{k+1}^{p-1} \\
 \alpha_k^p(t) &= \begin{cases} \frac{t - \tau_k}{\tau_{k+p} - \tau_k}, & \tau_{k+p} \neq \tau_k \\ 0, & \text{otherwise} \end{cases}
 \end{aligned} \tag{3.6}$$

where $B_k^p(t)$ denotes the k -th B-spline of degree p . The degree p and the number of interior knots $L - 1$ (i.e., excluding the end points), determine the number of spline basis functions according to $K = p + L$. In other words, the B-spline basis functions are $\{B_k^p(t); k = -p, -p + 1, \dots, L - 1\}$.

Fig. 3.1 illustrates all the 14 spline basis functions of degree 3 over 10 uniform interior knots (or 11 sub-intervals) defined in $[0, 1]$. The resulting augmented knot sequence is $\tau = \{0, 0, 0, 0, \frac{1}{11}, \dots, \frac{10}{11}, 1, 1, 1, 1\}$. The three leftmost and three rightmost basis functions are less smooth at the two boundaries than the other eight basis functions in the center. From leftmost to the right (rightmost to the left) the three basis functions are non-, once-, and twice-differentiable at the left (right) boundaries, respectively.

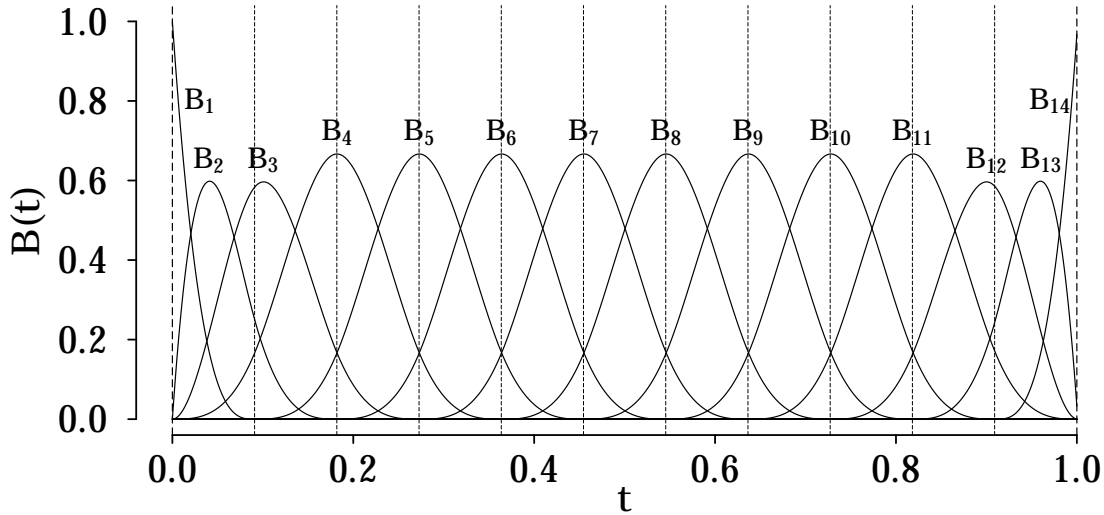


Figure 3.1: The fourteen B-spline basis functions of degree 3 (cubic) with 10 uniform interior knots (shown in light dashed vertical lines) that divides the domain into 11 sub-intervals.

Returning to the formulation of Eq. (3.3), a function y_i is then represented by B-spline basis functions of a given order $B_k^p(t)$ and a knot sequence τ as follows,

$$y_i(t) = \sum_{k=-p}^{L-1} c_{ik} \cdot B_k^p(t); \quad i = 1, 2, \dots, N \quad (3.7)$$

3.2.2 Curve Registration by Landmarks

Essential to the idea of summary statistics of a data set is the measure of central tendency (e.g., the mean), which characterizes a *typical* realization. The measure of dispersion such as the variance, in turn, can be defined relative to the mean. Two types of variations are often simultaneously present in a functional data set: the variation in magnitude (*vertical* variation) and the variation in phase (*horizontal* variation). The simultaneous presence of these two types of variations makes the definition of a mean function difficult [138].

variation in a functional data set: amplitude & phase

Fig. 3.2 illustrates this point. For a functional data set that does not contain strong phase variations, a simple cross-sectional mean (average values across realizations taken at every argument values) does indeed represent a typical realization (Fig. 3.2a). On the other hand, with a strong phase variation (often mixed with amplitude variation), the cross-sectional mean fails to produce a typical realization (Fig. 3.2b). In this case, according to Kneip ([138]) a more proper *structural* mean of the data set can be defined instead by first separating the phase and amplitude variations.

Cross-sectional vs. structural mean

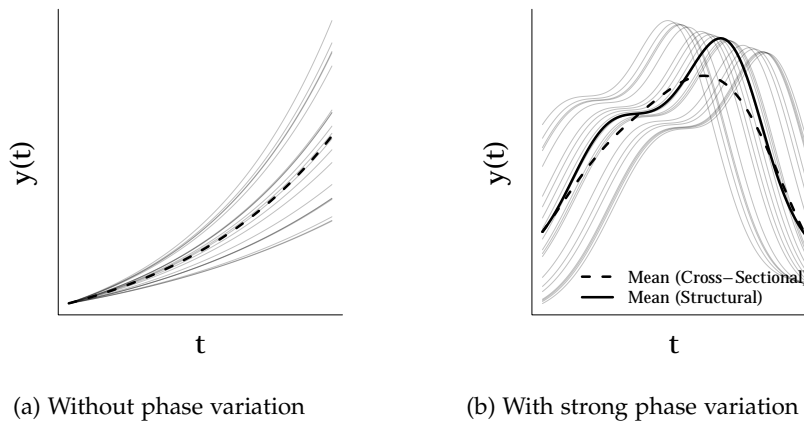


Figure 3.2: Two examples of functional data sets. (Left) without pronounced phase variations, cross-sectional mean can reflect a typical realization. (Right) with pronounced phase variations, cross-sectional mean differs from the notion of typical realization. *Structural* mean derived after registration better represents a typical realization. The scales in the axes are arbitrary.

*landmark
registration*

In order to obtain a meaningful structural mean in presence of strong amplitude and phase variations, the two types of variation are first separated through a *registration* procedure [139, 140]. The procedure transforms the time argument using a warping function to reduce the phase variation in the data set. Specifically, the *landmark* based registration can be employed when the main features of the function of interest (i.e., reflood curve) are readily identifiable. In a functional data set with phase variation, this particular type of registration forces important events in a curve (its *landmarks*) to occur at the same time relative to a set of reference values.

This is illustrated in Fig. 3.3. The left panel shows a functional data set exhibiting phase variation, which is reduced by aligning its landmark, here the time of maximum of each realization (shown as vertical solid lines) to a reference value (shown as the vertical dashed line). The structural mean (solid line curve) is simply computed as the cross-sectional mean of the registered curves shown on the right panel. The structural mean properly represents the inflection and the maximum points of a typical realization indeed, and therefore is more representative of the curves in the original data set.

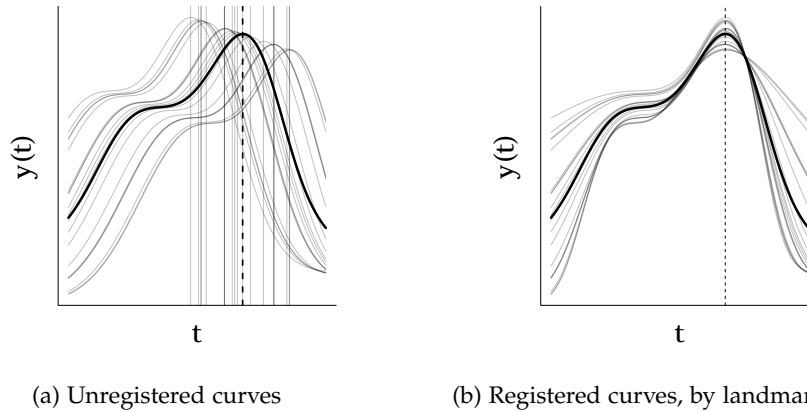


Figure 3.3: Illustration of curve registration. Shown in the left, curves whose landmarks (solid vertical lines) are to be aligned with respect to a reference value (dashed vertical line). Shown in the right, the registered curves. The structural mean is shown as a thick solid line curve. The scales in the axes are arbitrary.

More details on the properties of the warping functions used to transform the time argument for registration can be found in Appendix D.6.

3.2.3 Functional Principal Component Analysis

Separation of phase variation from magnitude variation by registration procedure allows for the definition of a proper mean function.

With respect to the mean, the notion of functional variation can then be defined. In the following discussion, it is assumed that the functional data set does not contain phase variation. That is, the functions are fully registered. If this is not the case then the previous step of registration might be required to first obtain a proper mean function.

The covariance function of a set of function realizations $\{y_n(t); n = 1, 2, \dots, N; t \in [t_a, t_b]\}$ from a random process \mathcal{Y} is defined as

Covariance function

$$v(t_1, t_2) \equiv \frac{1}{N} \sum_{n=1}^N (y_n(t_1) - \bar{y}(t_1)) \cdot (y_n(t_2) - \bar{y}(t_2)) \quad (3.8)$$

where $\bar{y}(t)$ is the proper mean function.

To extract more meaningful information from the covariance function, the function can be projected onto lower-dimensional space using an orthogonal decomposition. This projection can be done through the functional principal component analysis (fPCA) (also known as the Karhunen-Loève transform (KLT), see Appendix D.7 for the underlying theorem):

Functional principal component analysis (fPCA)

$$v(t_1, t_2) = \sum_{j=1}^{+\infty} \rho_j \cdot \xi_j(t_1) \cdot \xi_j(t_2) \quad (3.9)$$

where ρ_j is a series of ordered eigenvalues of decreasing values; $\xi_j(t)$ is the corresponding series of orthogonal eigenfunctions (or the fPC).

The transformation of the covariance function into pairs of eigenvalues and eigenfunctions also allows each element of the original data set $\{y_n(t)\}_{n=1}^N$ to be represented as a series that is optimal in the root-mean-square-error (RMSE) sense:

$$y_n(t) = \bar{y}(t) + \sum_{j=1}^{+\infty} \theta_{j,n} \cdot \xi_j(t); \quad n = 1, 2, \dots, N \quad (3.10)$$

where the fPC score $\theta_{j,n}$ associated with each realized function is defined by the orthogonality condition

$$\theta_{j,n} = \int_{t_a}^{t_b} [y_n(t) - \bar{y}(t)] \cdot \xi_j(t) dt \quad (3.11)$$

Eqs. (3.10) and (3.11) imply that across realizations in the samples, $\{y_n(t)\}$ can be represented linearly using a common mean function and sums of deviation terms from the mean. The deviation terms consist of a set of common eigenfunctions and a set of fPC scores. As such, the random character of each realization is left to the score associated with each component and each realization. Put differently, the eigenfunctions described the (common) modes of variations, while the scores quantify the strength of a particular mode [141]. These scores will be used as the QoI in the subsequent global SA. A way to compute the fPC and the associated scores can be found in Ref. [142].

3.3 PARAMETERS SCREENING

Screening methods are used to rank the importance of the model parameters using a relatively small number of model evaluations [39]. However, they tend to simply give qualitative measures. That is, meaningful information resides in the rank itself but not in the exact importance of the parameters with respect to the output. Screening is particularly valuable in the early phase of a SA to identify the noninfluential parameters of a model, which then could be safely excluded from further detailed analysis. This step is important to reduce the size of the problem especially if more expensive methods are to be applied at the subsequent steps. In this work, attention was paid to a particular screening method proposed by Morris [44] with an extension proposed by Campolongo et al. [132].

3.3.1 Elementary Effects and One-at-a-Time Design

Elementary effect

Consider a mathematical model $f : \mathbf{x} \in [0, 1]^D \mapsto \mathbf{y} = f(\mathbf{x}) \in \mathbb{R}$, where $\mathbf{x} = (x_1, x_2, \dots, x_D)$ is a vector of input parameters. The elementary effect of the d -th parameter on f is defined as

$$EE_d = \frac{f(x_1, \dots, x_d + \Delta, \dots, x_D) - f(x_1, \dots, x_d, \dots, x_D)}{\Delta} \quad (3.12)$$

or more concisely,

$$EE_d = \frac{f(\mathbf{x} + \Delta \cdot \mathbf{e}_d) - f(\mathbf{x})}{\Delta} \quad (3.13)$$

where \mathbf{e}_d is the d -th basis vector of the input parameter space; and Δ , the grid jump, is chosen such that $\mathbf{x} + \Delta \cdot \mathbf{e}_d$ is still in the specified domain of the parameter space, i.e., $[0, 1]^D$; Δ is a value in $\{\frac{1}{p-1}, \dots, 1 - \frac{1}{p-1}\}$, where p is the number of (discretization) levels that partition the model parameter space into a uniform grid of points where the model can be evaluated. For a given p , the grid constructs a finite distribution of $p^{D-1}[p - \Delta(p - 1)]$ elementary effects for each input parameter.

One-at-a-time
(OAT) experimental
design

The elementary effect distributions for each of the input parameters, evaluated across discretized input parameter space, provide useful information on the importance of a parameter on the output. Unfortunately, an exhaustive evaluation of all elementary effects for a given discretization levels suffers from a curse of dimensionality especially for numerous parameters or for reasonably fine discretization level¹. Consequently, a class of design of experiment that only change one parameter at a time (one-at-a-time (OAT)) are devised to estimate the statistics of the distributions.

Trajectory OAT
design

The key idea behind the original Morris method is in initiating the model evaluations from various *nominal* points, \mathbf{x} , randomly se-

¹ for $p = 8$ and $D = 20$ the total number of evaluations for exhaustive computation of all EEs is $\approx 6 \times 10^{17}$ for each parameter

lected over the grid and then gradually advancing one grid jump, perturbing one parameter at a time, making a perturbed point as the base point for the next perturbation. The order of perturbation (i.e., which dimension to perturb first) and the direction of the perturbation (i.e., whether it is added or subtracted) are also randomized between replications. As such, different replication generates different starting nominal point as well as different order and sign (but with the same size) of perturbation. The OAT experimental design complemented with this requirement is known as the trajectory design [143]. Fig. 3.4a illustrates a trajectory design with 4 replications in a two-dimensional input parameter space discretized in 6 levels .

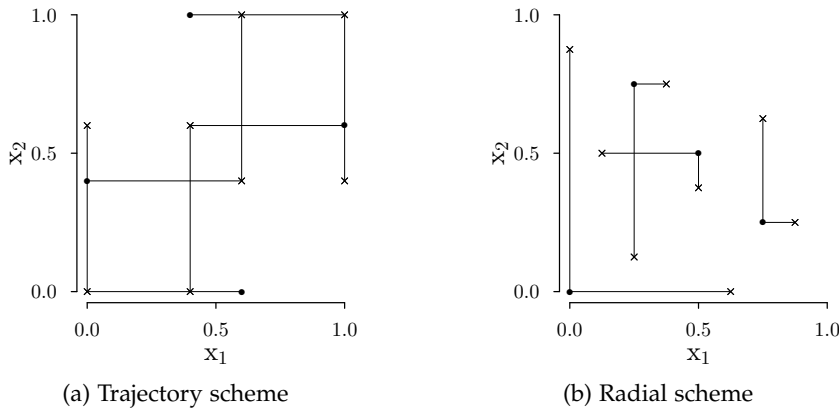


Figure 3.4: Illustration of One-at-a-Time (OAT) design constructed using trajectory scheme (left) and radial scheme (right) each with 4 replications. The trajectory design is discretized in 6 levels, while the number of levels is irrelevant for radial design. Filled circles are the nominal or the starting point (for the trajectory) or the base (for the radial) points and crosses are the perturbed levels.

To remove the requirement to specify a method-specific parameter p (the number of levels), Campolongo et al.[132] proposed to use a radial scheme coupled with Sobol' quasirandom sequence. In a single replication of this particular OAT design, each parameter is perturbed relative to a *base/nominal* point which is not required to be located on a predetermined grid. The size and sign of the perturbation is also allowed to vary from parameter to parameter in different replication. As such, radial design implicitly incorporates several additional possible sources of variation in the method that can potentially bias the estimation of elementary effects. Because the size of parameter perturbations varies, the definition of the elementary effects is slightly changed to

$$EE_d = \frac{f(\mathbf{x} + \Delta x_d \cdot \mathbf{e}_d) - f(\mathbf{x})}{\Delta x_d} \tag{3.14}$$

Radial OAT design

Elementary effect for radial design

where now each parameter at each design replication has its corresponding perturbation size $\Delta x_d \in [-1, 1]$ such that $x_d + \Delta x_d \in [0, 1]$.

An illustration of a radial design in a two-dimensional input parameter space with 4 nominal/base points is shown in Fig. 3.4b.

3.3.2 Statistics of Elementary Effects and Sensitivity Measures

Consider now that an N_R number of elementary effects associated with the d -th parameter have been sampled from the finite distribution of EE_d , using an OAT design with N_R replications, either based on the trajectory or radial design. The statistical summary of the sampled EE_d based on a given number of an OAT design replications can be calculated. The first is the arithmetic mean defined as,

*Mean of the
(sampled)
elementary effects*

$$\mu_d = \frac{1}{N_R} \sum_{r=1}^{N_R} EE_d^r \quad (3.15)$$

where EE_d^r is the elementary effect of the d -th parameter of the r -th replication. The mean gives the global influence of the d -th parameter on the chosen output f .

The second statistical summary of interest is the standard deviation of the (sampled) elementary effects for input parameter x_d ,

*Standard deviation
of the (sampled)
elementary effects*

$$\sigma_d = \sqrt{\frac{1}{N_R} \sum_{r=1}^{N_R} (EE_d^r - \mu_d)^2} \quad (3.16)$$

The standard deviation gives an indication of the presence of non-linearity or interactions between the d -th input parameter and the others.

In cases where f is a non-monotonic function, the sign of EE_d may change according to the change of the output, and cancellation effects on the estimation of μ_d might occur. To circumvent this issue, Campolongo et al. [132] proposed to take the absolute values of the sampled elementary effects. It is defined as,

*Mean of the
(sampled) absolute
elementary effects*

$$\mu_d^* = \frac{1}{N_R} \sum_{r=1}^{N_R} |EE_d^r| \quad (3.17)$$

Note that although the overall sign of the output perturbation is lost by using this measure, its use is justified if the input parameters are to be ranked based on a single importance measure.

The aforementioned statistical summaries, when evaluated over a large number of replications N_R , can provide global sensitivity measures of the importance of each input parameter. As indicated by Morris [44], there are three possible categories of parameter importance based on those statistics:

*Input parameter
importance
classification*

1. Parameters with noninfluential effects on the model output, i.e., the parameters that have relatively small values of both μ_d (or μ_d^*) and σ_d .
2. Parameters with linear or additive effects, i.e., the parameters that have relatively large value of μ_d (or μ_d^*) and relatively small value of σ_d . The small value of σ_d and the large value of μ_d (or μ_d^*) indicate that the variation of elementary effects across replications is small while the magnitude of the effect itself is consistently large for the perturbations in the parameter space.
3. Parameters with nonlinear or interaction effects, i.e., the parameters that have a relatively small value of μ_d (or μ_d^*) and a relatively large value of σ_d . Opposite to the previous case, a small value of μ_d (or μ_d^*) indicates that the aggregate effect of perturbation is relatively small (or in the case of μ_d , can be close to zero) while a large value of σ_d indicates that the variation of the effect is large; the effect can be large or negligibly small depending on the values of the other parameters at which the model is evaluated. Such large variation is a symptom of nonlinear effects or parameter interaction.

This classification makes parameter importance ranking and, in turn, screening of noninfluential parameters possible. However, the procedure is done rather qualitatively, and this is illustrated in Fig. 3.5, which depicts a typical parameter classification derived from a visual inspection of the elementary effects statistics on the σ vs. μ^* plane. The notions of influential and noninfluential parameters are based on the relative locations of those statistics in the plane. Typically, the noninfluential ones are clustered closer to the origin (relative to the more influential ones) with a pronounced boundary such as the situation depicted in Fig. 3.5. Admittedly, if these statistics are spread somewhat uniformly across the plane, the distinction would be more ambiguous and problematic². Furthermore, for a parameter with a large value of both μ^* and σ , the method cannot distinguish between nonlinearity effect from parameter interaction effect on the output.

3.4 VARIANCE DECOMPOSITION

Variance-based methods for GSA use variance as the basis to define a measure of input parameter influence on the overall output variation [33]. In a statistical framework of sensitivity and uncertainty analysis, this choice is natural because variance (or standard deviation) is often used as a measure of dispersion in the model prediction [34]. The

² In such a case, more advanced classification approaches such as the ones based on clustering techniques might be helpful to identify a finer structure of the parameters importance.

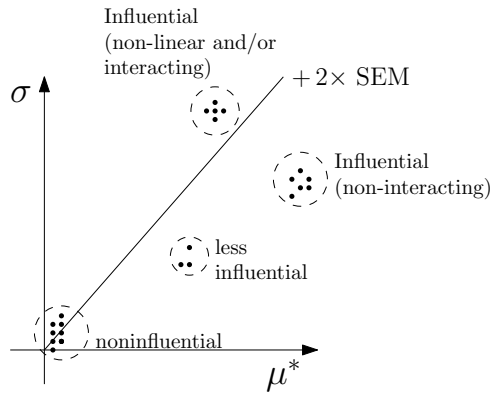


Figure 3.5: Illustration of a typical parameter importance classification obtained from the Morris screening method. The importance of each parameter relative to the other ones is defined with respect to its location on the $\sigma - \mu^*$ plane. Each dot represents a parameter, and the line corresponds to twice the standard error of the mean (SEM) indicating the relative magnitude of the standard deviation to the mean.

dispersion, in turn, can measure the level precision of the prediction when the input parameters are considered uncertain.

This section first presents a method to decompose the model output variance into the contributions from the individual variances of the inputs. Then, two sensitivity measures based on the decomposition are introduced and a method for their estimations is presented.

3.4.1 High-Dimensional Model Representation

High-dimensional model representation (HDMR)

Consider once more a mathematical model $f : \mathbf{x} \in [0, 1]^D \mapsto y = f(\mathbf{x}) \in \mathbb{R}$. The high-dimensional model representation (HDMR) of $f(\mathbf{x})$ is a linear combination of functions with increasing dimensionality up to the dimension of \mathbf{x} [144],

$$f(\mathbf{x}) = f_o + \sum_{d_1=1}^D f_d(x_{d_1}) + \sum_{1 \leq d_1 < d_2 \leq D} f_{d_1, d_2}(x_{d_1}, x_{d_2}) + \dots + f_{1, 2, \dots, D}(x_1, x_2, \dots, x_D) \quad (3.18)$$

where f_o is a constant. The representation in Eq. (3.18) is unique if the following condition [41]:

$$\int_0^1 f_{d_1, d_2, \dots, d_i}(x_{d_1}, x_{d_2}, \dots, x_{d_i}) dx_{d_m} = 0; \text{ for } m = 1, 2, \dots, i \quad (3.19)$$

is established for all $i \in 1, \dots, D$ and any corresponding ordered combination of dimensions $1 \leq d_1 < d_2 < \dots < d_i \leq D$ of the input parameter space.

Assume now that \mathcal{X} is a random vector of independent and uniform random variables over a unit hypercube $\{\Omega = \mathbf{x} \mid 0 \leq x_i \leq 1; i = 1, \dots, D\}$ such that $y = f(\mathbf{x})$ becomes

$$y = f(\mathcal{X}) \quad (3.20)$$

where y is a random variable, resulting from the transformation of the random vector \mathcal{X} by the function f . Using Eq. (3.19) to express each term in Eq. (3.18), it follows that

$$\begin{aligned} f_o &= \mathbb{E}[y] \\ f_{d_1}(x_{d_1}) &= \mathbb{E}_{\sim d_1}[y|\mathcal{X}_{d_1}] - \mathbb{E}[y] \\ f_{d_1, d_2}(x_{d_1}, x_{d_2}) &= \mathbb{E}_{\sim d_1, d_2}[y|\mathcal{X}_{d_1}, \mathcal{X}_{d_2}] \\ &\quad - \mathbb{E}_{\sim d_1}[y|\mathcal{X}_{d_1}] - \mathbb{E}_{\sim d_2}[y|\mathcal{X}_{d_2}] - \mathbb{E}[y] \end{aligned} \quad (3.21)$$

The same follows for higher-order terms in the decomposition. In Eq. (3.21), $\mathbb{E}_{\sim o}[\circ|\circ]$ is a conditional expectation operator, where the subscript symbol $\sim o$ means that integration on the parameter space is carried out over all parameters except the one(s) in the subscript. For instance, $\mathbb{E}_{\sim 1}[y|\mathcal{X}_1]$ refers to the conditional mean of y given \mathcal{X}_1 , and the integration is carried out for all possible values of parameters in \mathbf{x} except x_1 . Note that because \mathcal{X}_1 is a random variable, the expectation conditioned on it is also a random variable.

Assuming that f is square integrable, applying the variance operator on y results in

$$\begin{aligned} \mathbb{V}[y] &= \sum_{d_1=1}^D \mathbb{V}[f_{d_1}(x_{d_1})] + \sum_{1 \leq d_1 < d_2 \leq D} \mathbb{V}[f_{d_1, d_2}(x_{d_1}, x_{d_2})] + \dots \\ &\quad + \mathbb{V}[f_{1, 2, \dots, D}(x_1, x_2, \dots, x_D)] \end{aligned} \quad (3.22)$$

3.4.2 Sobol' Sensitivity Indices

Division by $\mathbb{V}[y]$ aptly normalizes Eq. (3.22):

$$1 = \sum_{d_1=1}^D S_{d_1} + \sum_{1 \leq d_1 < d_2 \leq D} S_{d_1, d_2} + \dots + S_{1, 2, \dots, D} \quad (3.23)$$

where Sobol' main-effect sensitivity index S_d is defined as,

Main-effect index

$$S_d = \frac{\mathbb{V}_d[\mathbb{E}_{\sim d}[y|\mathcal{X}_d]]}{\mathbb{V}[y]} \quad (3.24)$$

The numerator is the variance of the conditional expectation, and the index is a global sensitivity measure interpreted as the amount of variance reduction in the model output if the parameter \mathcal{X}_d is fixed (i.e., its variance is reduced to zero). The main-effect sensitivity index is also known in the literature as the *first-order* sensitivity index as it

captures only the effect of a single parameter variation on the model output considering *no interaction* with other parameters [133].

Total-effect index A closely related sensitivity index proposed by Homma and Saltelli [145] is the Sobol' total-effect index defined as,

$$\begin{aligned} ST_d &= \frac{\mathbb{E}_{\sim d}[\mathbb{V}_d[y|\mathcal{X}_{\sim d}]]}{\mathbb{V}[y]} = \frac{\mathbb{V}[y] - \mathbb{V}_{\sim d}[\mathbb{E}_d[y|\mathcal{X}_{\sim d}]]}{\mathbb{V}[y]} \\ &= 1 - \frac{\mathbb{V}_{\sim d}[\mathbb{E}_d[y|\mathcal{X}_{\sim d}]]}{\mathbb{V}[y]} \end{aligned} \quad (3.25)$$

The index, also a global sensitivity measure, can be interpreted as the amount of variance left in the output if the values of all input parameters, *except* \mathcal{X}_d , can be fixed. In other words, the total-effect index measures the contribution to the output variance of parameter \mathcal{X}_d , including all variance caused by its interactions, of any order, with any other parameters.

*Parameter
prioritization
objective*

These two sensitivity measures can serve the objectives of GSA for model assessment as proposed by Saltelli et al [34, 39]. The main-effect index is relevant to parameter prioritization in the context of identifying the most influential parameter since fixing a parameter with the highest index value would, *on average*, lead to the greatest reduction in the output variation.

*Parameter screening
objective*

The total-effect index, on the other hand, is relevant to parameter fixing (or *screening*) in the context of identifying the least influential set of parameters since fixing any parameter that has a very small total-effect index value would not lead to significant reduction in the output variation. The use of the total-effect index to identify which parameter can be fixed or excluded is similar to that of the elementary effect statistics of the Morris method, albeit more exact.

Finally, the difference between the two indices of a given input parameter, i.e., $ST_d - S_d$, is used to quantify the amount of all interactions involving that parameter in the model output.

3.5 IMPLEMENTATION

In this work, an implementation of the Morris method and a Monte Carlo method to estimate the main- and total-effect sensitivity indices has been developed using the Python [146] programming language, to allow for well-controlled parametric and convergence studies. The implementation follows a black box approach of SA. It deals with the generation of design of experiment (a set of input values at which the model or code is evaluated) and the post-processing of output to obtain the selected measures of sensitivity presented in the previous sub-sections. In the following, the basic procedures that underlie the implementation of both methods are laid out. More details on the programming aspects of the implementation (the so-called *gsa-module*) can be found in Appendix C.1.

3.5.1 The Morris Method

The implementation of the Morris method (see Section 3.3.1) follows four sequential steps:

1. An OAT design matrix consisting of N_R replications is created by randomly sampling the nominal points as well as the perturbed points for each parameter. A replication in an OAT design consists of one nominal point with D (number of dimensions/parameters) additional perturbed points. In each of the perturbed points, only one parameter change its value relative to the base. Note that in trajectory design, the nominal point only serves as the starting point and a perturbed point becomes the base point for the next perturbation. Different replications yield different nominal points and the associated perturbed points.
2. Each point in the design matrix, included in $[0, 1]^D$, is scaled to the corresponding point in the D -dimensional parameter space of the model parameters through iso-probabilistic transformation (see Appendix D.4).
3. The model is evaluated for each (rescaled) point of the design matrix. The total number of model evaluations for a given design matrix is $N_R \times (D + 1)$.
4. Finally, for a selected QoI the N_R elementary effects EE_d are computed for each input parameter x_d . The statistical summaries μ_d , μ_d^* , and σ_d are computed, and the ranking of the input parameters for the selected QoI is established based on μ_d^* .

The different rankings based on μ_d^* obtained from various relevant QoIs can then be used and compared to consistently identify and screen out noninfluential parameters (low μ_d^*) from the relatively influential ones (high μ_d^*) [147].

3.5.2 The Sobol'-Saltelli Method

In principle, the estimation of the Sobol' sensitivity indices defined by Eqs. (3.24) and (3.25) can be directly carried out using MC simulation. The most straightforward, though rather rudimentary, estimation method is to use two nested loops for the computation of the conditional variances and expectations appearing in both indices.

In the estimation of the main-effect index of parameter x_d , for instance, the outer loop samples values of x_d while the inner loop samples values of $x_{\sim d}$ (all parameters except x_d). These samples, in turn, are used to evaluate the model and generate the output realizations.

Algorithm 1 illustrates the procedure to compute the variance of conditional expectation used in main-effect indices estimation of a

*Brute force
Monte Carlo*

parameter x_d . In the inner loop, the arithmetic mean of the model output is taken for a given value of \mathcal{X}_d but over many values of $\mathcal{X}_{\sim d}$. Afterward, in the outer loop, the variance of the model output is taken over many values of \mathcal{X}_d .

Algorithm 1 Brute Force MC for estimating $\mathbb{V}_d[\mathbb{E}_{\sim d}[\mathcal{Y}|\mathcal{X}_d]]$

```

 $\Sigma_i \leftarrow 0$ 
 $\Sigma_{i^2} \leftarrow 0$ 
for  $i = 1$  to  $N$  do
  sample  $x_d^{(i)}$  from  $\mathcal{X}_d$ 
   $\Sigma_j \leftarrow 0$ 
  for  $j = 1$  to  $N$  do
    sample  $x_{\sim d}^{(j)}$  from  $\mathcal{X}_{\sim d}$ 
     $\Sigma_j += f(x_{\sim d}^{(j)}, x_d^{(i)})$ 
  end for
   $\mathbb{E}_{\sim d}[\mathcal{Y}|\mathcal{X}_d]^{(i)} \leftarrow \frac{1}{N} \Sigma_j$ 
   $\Sigma_i += \mathbb{E}_{\sim d}[\mathcal{Y}|\mathcal{X}_d]^{(i)}$ 
   $\Sigma_{i^2} += \mathbb{E}_{\sim d}[\mathcal{Y}|\mathcal{X}_d]^{(i)} \times \mathbb{E}_{\sim d}[\mathcal{Y}|\mathcal{X}_d]^{(i)}$ 
end for
 $\mathbb{V}_d[\mathbb{E}_{\sim d}[\mathcal{Y}|\mathcal{X}_d]] \leftarrow \frac{1}{N} (\Sigma_{i^2} - \Sigma_i^2/N)$ 

```

Algorithm 1 can easily become prohibitively expensive as the nested structure requires N^2 model evaluations *per input dimension* for one of the sensitivity indices (i.e., the main- or total-effect index), while N (the size of MC samples) are typically in the range of $10^2 - 10^4$ for a reliable estimate.

Sobol' [41] and Saltelli [133] proposed an alternative approach that circumvent the nested structure of MC simulation to estimate the indices. The formulation starts by expressing the expectation and variance operators in their integral form and ends with different possible MC estimators for both sensitivity indices. A detailed derivation of the integral form and the origin of the estimator can be found in Appendix D.1.

An implementation of the Sobol'-Saltelli method is also part of `gsa-module python3` package (see Appendix C.1 for detail). For N number of MC samples and D number of model parameters, the MC simulation procedure to estimate the sensitivity indices follows the sampling and resampling approach adopted in [41, 133, 145] described in the following.

1. Generate two $N \times D$ independent random samples A and B from a uniform independent distribution in D -dimension, $[0, 1]^D$:

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1D} \\ \vdots & \ddots & \vdots \\ a_{N1} & \cdots & a_{ND} \end{pmatrix}; \quad B = \begin{pmatrix} b_{11} & \cdots & b_{1D} \\ \vdots & \ddots & \vdots \\ b_{N1} & \cdots & b_{ND} \end{pmatrix} \quad (3.26)$$

*Sobol'-Saltelli
method*

2. Construct D additional design matrices from A and B where each matrix A_B^d is matrix A with the d -th column substituted by the d -th column of B :

$$\begin{aligned}
 A_B^1 &= \begin{pmatrix} b_{11} & \cdots & a_{1D} \\ \vdots & \ddots & \vdots \\ b_{N1} & \cdots & a_{ND} \end{pmatrix} \\
 A_B^d &= \begin{pmatrix} a_{11} & \cdots & b_{1d} & \cdots & a_{1D} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ a_{N1} & \cdots & b_{Nd} & \cdots & a_{ND} \end{pmatrix} \\
 A_B^D &= \begin{pmatrix} a_{11} & \cdots & b_{1D} \\ \vdots & \ddots & \vdots \\ a_{N1} & \cdots & b_{ND} \end{pmatrix}
 \end{aligned} \tag{3.27}$$

3. Rescale each element in the matrices of samples to the actual values of model parameters according to their actual range of variation through iso-probabilistic transformation (Appendix D.4).
4. Evaluate the model multiple times using input parameters vectors that correspond to each row of A , B , and all A_B^d .
5. Finally, extract the QoIs from all the outputs and recast them as vectors, which will be used to estimate the main- and total-effect indices according to a selected estimator described below.

For the main-effect sensitivity index, two estimators are considered. One is proposed by Saltelli [133], and the other, as an alternative, is proposed by Janon et al [148]. The latter proved to be more efficient, especially for a large variation around a parameter estimate [30, 148].

The general form of main-effect sensitivity index estimator is

$$\widehat{S}_d = \frac{\frac{1}{N} \sum_{n=1}^N f(B)_n \cdot f(A_B^d)_n - \mathbb{E}^2[y]}{\mathbb{V}[y]} \tag{3.28}$$

*Main-effect
sensitivity index
estimator*

where the subscript n corresponds to the row of the sampled model parameters such that $f(B)_n$ and $f(A_B^d)_n$ are the model outputs evaluated using inputs taken from the n -th row of matrix B and matrix A_B^d , respectively. The estimators for the term $\mathbb{E}^2[y]$ and $\mathbb{V}[y]$ differs for the two indices estimators and are given in Table 3.1.

To estimate the total-effect sensitivity indices, the Jansen estimator [149] is recommended in [150]. The estimator reads

$$\widehat{ST}_d = \frac{\frac{1}{2N} \sum_{n=1}^N (f(A)_n - f(A_B^d)_n)^2}{\mathbb{V}[y]} \tag{3.29}$$

*Total-effect
sensitivity index
estimator*

where $\mathbb{V}[y]$ is estimated by the Saltelli et al. estimator in Table 3.1.

Table 3.1: Two MC estimators for the terms in Eq. (D.6) to estimate the main-effect indices (the sum is taken implicitly over all samples N)

ESTIMATOR	$\mathbb{E}^2[y] = \left(\int f dx\right)^2$	$\mathbb{V}[y] = \int f^2 dx - \left(\int f dx\right)^2$
Saltelli [133]	$\frac{1}{N} \sum f(A)_n \cdot f(B)_n$	$\frac{1}{N} \sum f(A)_n^2 - \left(\frac{1}{N} \sum f(A)_n\right)^2$
Janon et al. [148]	$\left(\frac{1}{2N} \sum f(B)_n + f(A_B^d)_n\right)^2$	$\frac{1}{2N} \sum f(B)_n^2 + f(A_B^d)_n^2$ $- \left(\frac{1}{2N} \sum f(B)_n + f(A_B^d)_n\right)^2$

*Computational cost:
brute force Monte
Carlo vs.
Sobol'-Saltelli*

The computational cost associated with the estimation of all the main-effect and total-effect indices using the Sobol'-Saltelli method is $N \times (D + 2)$ code runs, where N is the number of MC samples and D is the number of parameters. Compare this to the cost of brute force Monte Carlo of $2 \times D \times N^2$ code runs to estimate all the main-effect and total-effect sensitivity indices.

*Computational cost:
Morris vs.
Sobol'-Saltelli*

As an additional comparison, the cost for Morris method to compute the statistics of elementary effect is $N_R \times (D + 1)$ code runs, where N_R is the number of OAT design replications. In either methods, the number of samples N (in the case of the Sobol'-Saltelli method) and replications N_R (in the case of the Morris method) determines the precision of the estimates. A larger number of samples (and replications) increases the precision. Note, however, that in practice the typical number of Morris replications is between $10^1 - 10^2$ [151], while the number of MC samples for the Sobol' indices estimation amounts to $10^2 - 10^4$ [41].

3.6 APPLICATION TO TRACE MODEL OF FEBA

The GSA methodology presented above was applied to analyze the simulation of the FEBA experiments using the TRACE model described in Chapter 2. In the following, only the results from analyzing FEBA test No. 216 (with inlet velocity of 3.8×10^{-2} [m.s⁻¹] and system pressure of 4.1 [bar]) are presented.

3.6.1 Simulation Experiment

Screening analysis

The simulation experiment for global sensitivity analysis on the TRACE model of FEBA was carried out in two steps. The first step was aimed at screening out any possible noninfluential parameters with relatively few code runs using several screening methods, i.e., the two variants of the Morris screening method (the radial and the trajectory designs) and the Sobol'-Saltelli method; with the latter only the total-effect indices were estimated, in line with the factor fixing objective.

The second step of the analysis consisted of variance decomposition through the estimation of the Sobol' indices on the reduced number of parameters. This was a more detailed analysis where the contribution of each input parameter variation to a particular output variance was quantified. Since the number of code runs is directly proportional to the number of input parameters, the screening procedure done in the first step allowed us to reduce the size of the problem and to generate a larger sample for a fewer code runs³. A larger sample, in turn, led to a more precise Sobol' indices estimates. The experimental design matrix used to carry out the estimation was generated using a Sobol' quasi-random sequence generator [152].

*Variance
decomposition*

Different types of QoIs were investigated for this simulation experiment. The application of the Morris method to the TRACE model of the FEBA facility to rank the parameter importance was already demonstrated in [153] using the time-averaged temperature as QoI, which is defined as

*Quantities of
Interest (QoIs)*

$$\bar{T} = \frac{\int T(t)dt}{\int dt} \quad (3.30)$$

where the integration of the pointwise time-dependent reflood curve was approximated using the trapezoidal rule over the duration of the transient. The time-averaged temperature was selected as the simplest possible scalar QoI to capture the overall variation of the temperature transient since it was previously shown that a high maximum clad temperature as QoI would not necessarily imply a delayed time of quenching, and vice versa [153]. To further investigate these aspects, the maximum clad temperature and the quench time have also been considered as QoIs in this study.

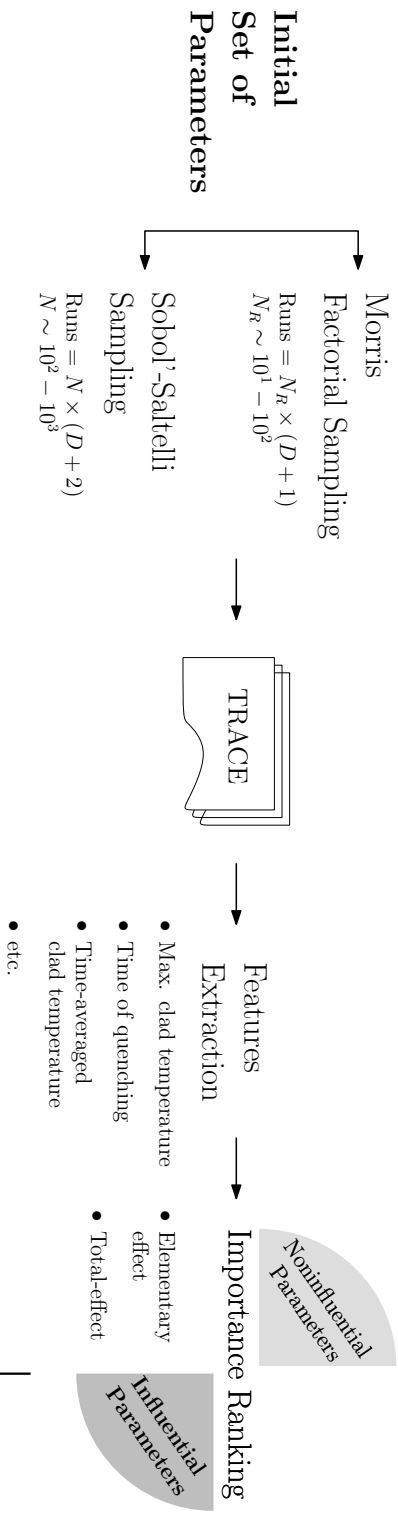
To represent better the notion of functional variation, FDA-based techniques were applied to derive a new set of QoIs. All the steps required in the application of FDA were already demonstrated in the context of the reflood simulation output in [142]. All the required computations related to FDA were done with the R programming language [137] using the `fda` package [154]. The application of FDA resulted in a set of common fPCs and a set of realization-specific fPC scores. The scores were therefore used as the QoI for the Sobol' indices estimation and compared to the indices obtained from the more conventional QoIs for the reflood transient.

*FDA-based
Quantities of
Interest*

Finally, to give a measure of the uncertainty in all indices estimated by the Sobol'-Saltelli method, the 95% percentile confidence interval (CI) were constructed using the bootstrap technique (see [155] and for detail). The flowchart of the simulation experiment for the analysis is illustrated and summarized in Fig. 3.6.

³ Note that the total duration of the transient is set to be 600 [s] and each TRACE run required between ~ 400 – 600 [CPUs], where [CPUs] is "Central Processing Unit seconds".

1. Screening Analysis



2. Variance Decomposition

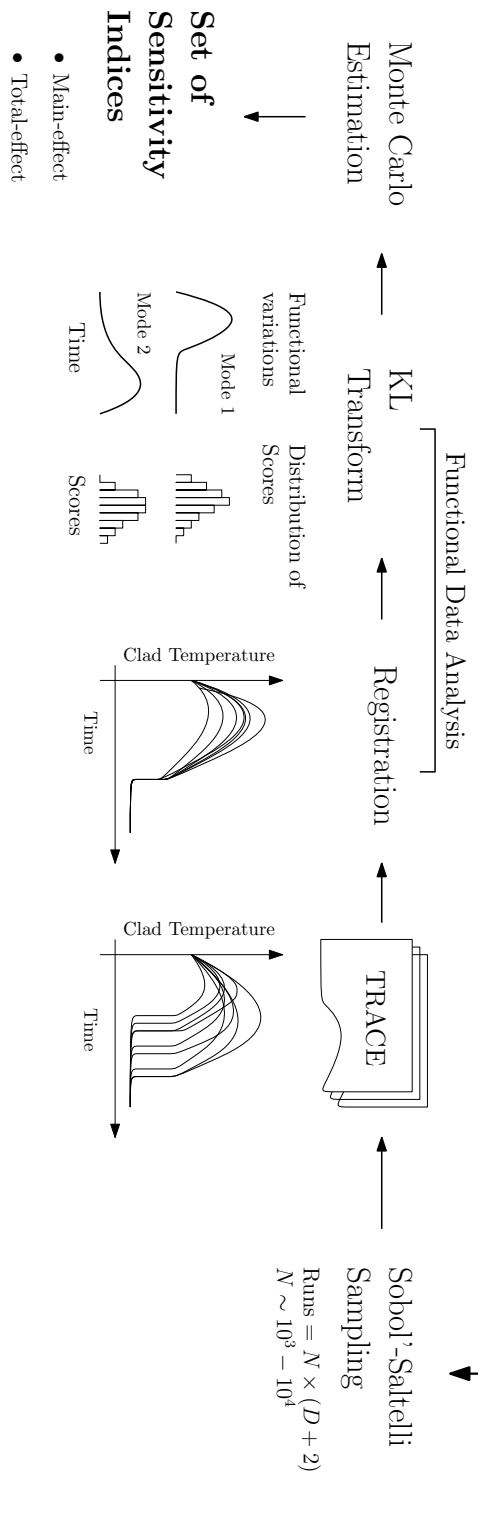


Figure 3.6: Flowchart for the implemented sensitivity analysis methodology applied to the TRACE model of the FEBIA facility.

3.6.2 Screening Analysis

A screening analysis to identify the noninfluential parameters was first carried out on the 27 model parameters using three different methods. The QoIs for this screening analysis were the time-averaged quantities for all output types (clad temperature, channel pressure drop and liquid carryover) as explained in Sec. 3.6.1. 320 replications were used for the Morris method while 1'000 samples were used to estimate the total-effect indices for the Sobol'-Saltelli method. The parameter ranking was constructed based on μ_d^* (for the two Morris methods) and $\hat{S}T_d$ (the total-effect indices).

Fig. 3.7 gives an example of convergence of sensitivity measures μ_d^* and $\hat{S}T_d$ with respect to the average temperature at the middle of the assembly, with increasing number of replications (for the Morris methods) and samples (for the Sobol' total-effect indices estimation). It is shown that all of the sensitivity measures converged and the most important parameters identified by each of the methods are the same (in this case: gridHT, dffbIntDr, dffbWHT, dffbVIHT). Note that the values of the two measures cannot be compared with each other. The convergence of these measures with respect to other outputs was also found to have the same behavior.

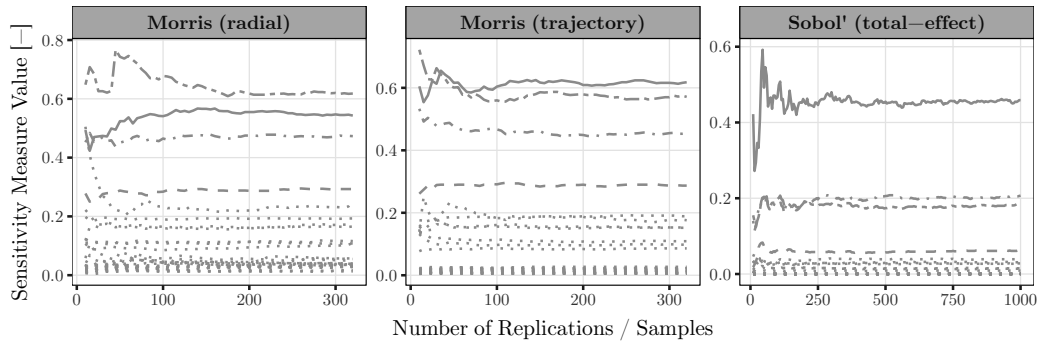


Figure 3.7: Evolution of μ_d^* and $\hat{S}T_d$ with respect to the average temperature at the middle of the assembly as a function of the number of replications or samples. The sensitivity measures related to the four most important parameters, the same for each method, are given unique line types in the plots.

An important finding is that the noninfluential parameters identified by both variants of the Morris methods are confirmed by the total-effect indices based on the variance decomposition, whose values are estimated with small uncertainty. Table 3.2 presents the summary of parameters importance across different outputs. In the table, a parameter is considered noninfluential with respect to a particular output type if its Sobol' total-effect index value falls below 5%. This parameter will be screened out in the downstream analysis. The final selection of 12 important parameters are then made by making a union set of the important parameters identified with respect to the different outputs. Complete numerical results of the sensitivity mea-

asures used in the ranking/screening of each parameter with respect to each of the different outputs are tabulated in Appendix B.2.

Table 3.2: Parameters importance across different outputs, average quantities over the transient. Checkmark signifies a parameter with a Sobol' total-effect indices above 5% and shaded cells signify the final selection of the retained influential parameters.

No.	Parameter	TC (1 is at the top, 8 is at the bottom of the assembly)								DP				CO
		1	2	3	4	5	6	7	8	Bot.	Mid.	Top	Tot.	
1	breakP										✓			
2	fillT							✓						
3	fillV				✓	✓	✓	✓		✓	✓	✓	✓	✓
4	pwr										✓			
5	nicK													
6	nicCP													
7	nicEm													
8	mgoK													
9	mgoCp													
10	vesEps													
11	ssK													
12	ssCp													
13	ssEm													
14	GridK													
15	GridHT	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓
16	iafbWHT							✓						
17	dffbWHT		✓	✓	✓	✓	✓	✓			✓	✓	✓	
18	iafbLIHT													
19	iafbVIHT													
20	dffbLIHT													
21	dffbVIHT	✓	✓	✓	✓							✓		✓
22	iafbIntDr									✓	✓	✓	✓	
23	dffbIntDr	✓	✓	✓	✓	✓	✓			✓	✓	✓	✓	✓
24	iafbWDr													
25	dffbWDr											✓	✓	
26	transWHT													
27	tQuench							✓		✓	✓		✓	

Using a 5% cut-off value to screen out noninfluential parameters is admittedly an *ad hoc* approach. To check the consistency of the screening approach, Fig. 3.8 illustrates the notion of noninfluential and influential parameters, in terms of the effects of their perturbations on the transient of three different outputs. TRACE was executed using 500 samples of parameter value from each set of parameters (i.e., influential and noninfluential). The figure confirms that the use of time-averaged quantity for each output type is a viable QoI for screening. The identified noninfluential parameters were indeed the ones that result in minor variation (black curves) of all outputs transient as compared to the variation brought by the influential parameter perturbations (gray curves). Furthermore, it also confirms that by making the union set of all the important parameters subsets (each with respect to a particular output, using the 5% cut-off value), the final selection of 12 influential parameters is valid for all outputs.

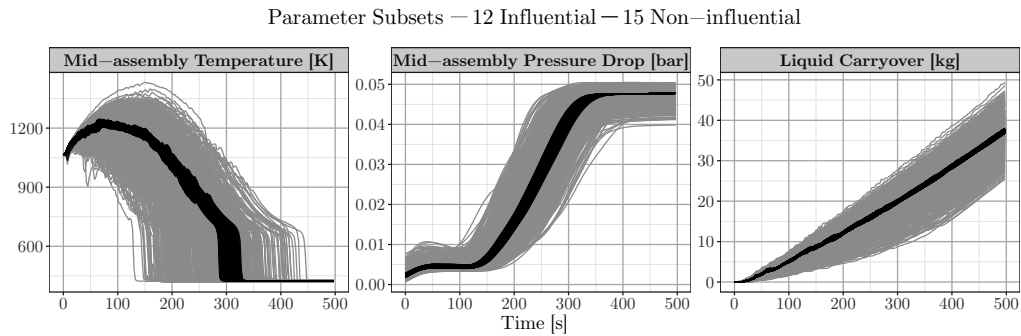


Figure 3.8: Illustration of the variations in the transient of three different outputs using only the 12 influential parameters (background, gray) and only the 15 noninfluential parameters (foreground, black). Each case uses 500 samples.

From the screening analysis results, a more detailed analysis was carried out on the 12-parameter model involving only the aforementioned influential parameters. The detailed analysis consisted of the estimation of the Sobol' main-effect sensitivity index (in complementary with the total-effect index used in the screening above) with respect to different types of time-dependent outputs as well as to different QoIs associated with each of them. The analysis was aimed at exposing how an individual input parameter might have affected particular model behavior as highlighted by the different choices of QoIs.

It should be noted that the estimation of the main-effect indices were relatively more expensive as a larger number of samples was required to reliably estimate the indices (i.e., such that the uncertainty associated with the Monte Carlo estimation was within an acceptable level). In relation to this, the convergence of two different Sobol' main-effect index estimators as well as one Sobol' total-effect index estimator was investigated empirically. The result of the analysis was

*FEBA TRACE
model with 12
influential
parameters*

*Convergence of the
Sobol' indices
estimation*

useful in the planning of the simulation experiments regarding the number of samples in relation to the expected uncertainty (in terms of confidence interval (CI)) of the estimates. It was found that the CI length of a given estimator depended on the QoI, the estimand, the estimator used, and the number of samples. A more detailed discussion is presented in Appendix B.3.

Consequently, by benefiting from the screening procedure taken before (12 influential parameters instead of 27 parameters) and by considering the results of the empirical convergence study, a total of 2'000 samples (which corresponds to 28'000 TRACE runs) was deemed appropriate for the more detailed SA presented below. For conciseness only the results of selected types of output are presented to illustrate the method application, namely the mid-height clad temperature transient (TC4 at elevation $z = 2.4$ [m]), the pressure drop transient at the middle of the assembly (the segment between $z = 1.7$ [m] and $z = 2.3$ [m]), and the liquid carryover.

*Selected results for
detailed analysis*

3.6.3 Sobol' Indices for Conventional QoIs of the Reflood Curve

As explained in Chapter 2, two conventional QoIs to characterize a reflood curve are the maximum clad temperature and the time of quenching. As shown in Fig. 3.9 the variation of the maximum mid-height clad temperature (with standard deviation of 59.7 [K]) was driven mainly by four model parameters, contributing up to 77% of the total output variation. One influential parameter was related to the spacer grid heat transfer enhancement model and the three others were related to the DFFB-regime model parameters (with a combined effect of 63%). Moreover since the sum of all the main-effect indices was relatively close to 1.0, the parameters were not interacting with respect to this particular QoI.

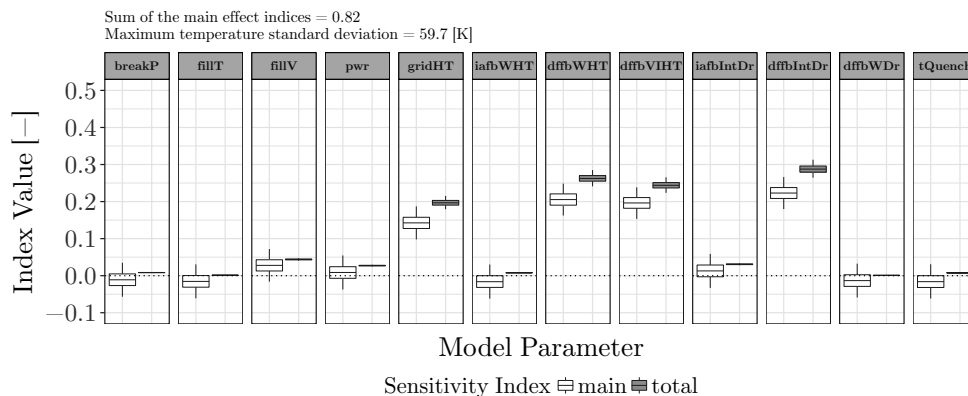


Figure 3.9: The main-effect and total-effect sensitivity indices with the maximum mid-height clad temperature as the QoI. Each boxplot represents the bootstrap sample quartile statistics and the vertical line extends the 95th sample percentile.

The parameter sensitivity with respect to the time of quenching gave a different picture as shown in Fig. 3.10. The variation in the time of quenching (with standard deviation of 59.9 [s]) was driven mainly by the spacer grid heat transfer enhancement parameter model (with contribution close to 50% of the total output variation). The DFFB-related parameters were next in line with a combined contribution of about ~ 18%, while each of the other parameters contributes to less than 10% of the total output variation. Similar to the case of the maximum clad temperature, no strong interaction effect was observed.

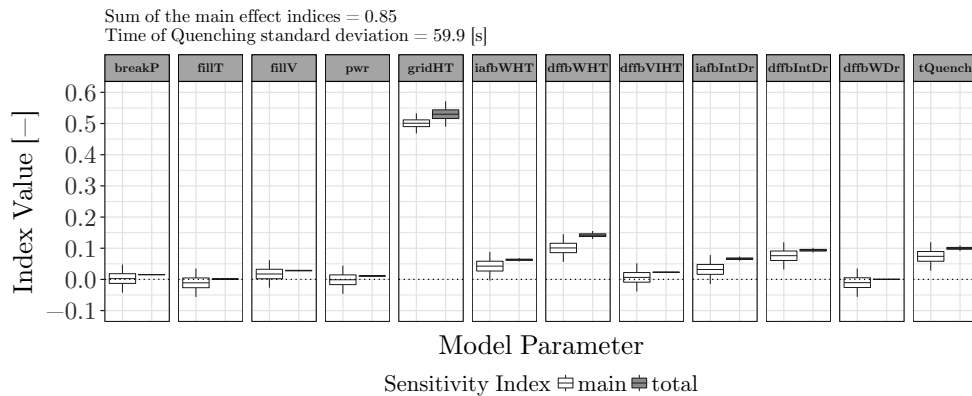


Figure 3.10: The main-effect and total-effect sensitivity indices with the time of quenching at the mid-height of the assembly as the QoI. Each boxplot represents the bootstrap sample quartile statistics and the vertical line extends the 95th sample percentile.

To better understand how the output sensitivity to the model parameters is changing over the course of the reflood transient, the mid-height clad temperature at each time step was taken as the QoI and the main-effect indices were calculated. This resulted in a set of sensitivity indices at each time with respect to the clad temperature as presented in Fig. 3.11. Note that the indices presented in the figure correspond to the reflood curves in which the phase variations between realizations were removed through the registration procedure.

The top panel of Fig. 3.11 shows how the relative importance of the parameters and their interactions in a dynamic model change with time. With respect to the clad temperature, up to 120 [s], the model parameters were non-interacting as indicated by the sum of the main-effect indices that was close to 1.0. The spacer heat transfer enhancement and DFFB-related model parameters were found to be the most important parameters in this time period.

However, from 120 [s] onward, stronger parameter interactions took place, as indicated by the decreasing sum of the main-effect indices which at its minimum only explained well below 20% of the total output variation. Furthermore, other parameters also became more prominent at a later stage of the transient. The quench temperature (t_{Quench}), which for the most part of the transient was non-influential started to top after 200 [s]. At ~ 300 [s], the temperature transient vari-

ations suddenly were driven only by parameter interactions. Finally, the variation of the pressure boundary condition (breakP) accounted for most of the temperature variance at the end of the transient.

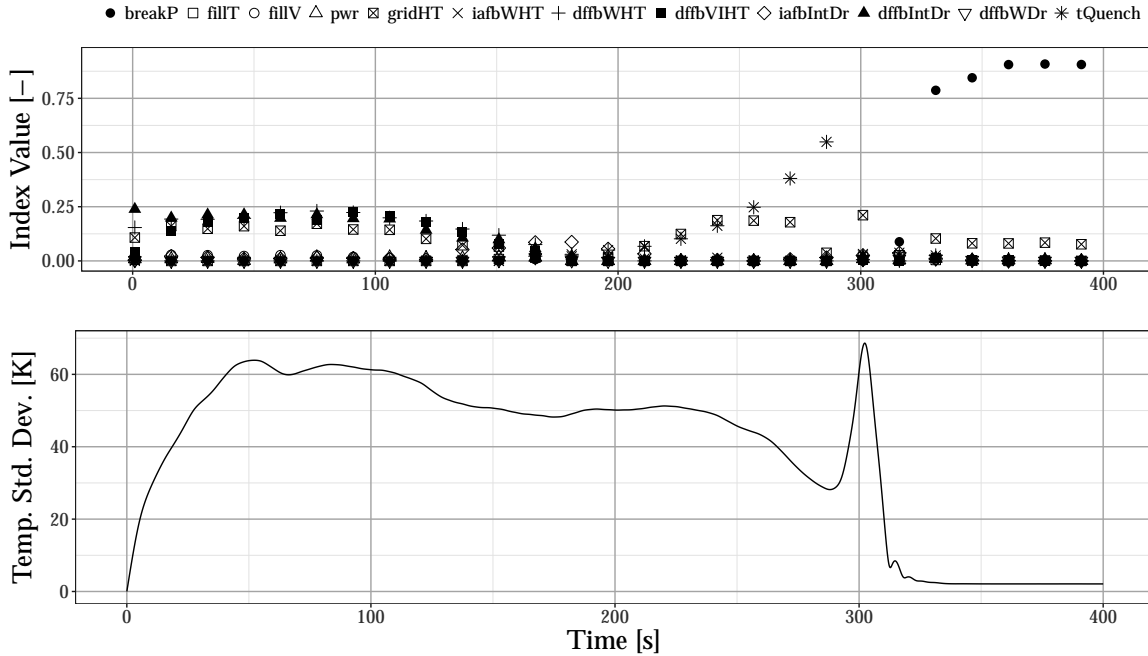


Figure 3.11: (top) the main-effect sensitivity indices at different time steps during the reflood transient. (bottom) The clad temperature standard deviation at different time steps during the same transient.

To put the dynamic behavior of the sensitivity indices in context, the temperature variation is also given for each time step in the bottom panel of Fig. 3.11. Note that in the plot, the last part of the transient (where the pressure boundary condition becomes visibly important) amounts to 2 [K], a hardly relevant magnitude in the current context. After quenching, the clad surface temperature is essentially commensurate with the coolant temperature. The small temperature variation, in turn, corresponds to the change in the saturation temperature at the outlet due to variation in the pressure boundary condition.

The figure also shows some sign of imperfection in the registration procedure. The sudden jump of variation around the time of quenching can be attributed to a residual misalignment that still exists in the registered dataset. As the landmark registration is supposed to remove the phase variation with respect to the time of quenching (one of the landmarks), temperature variation of this magnitude at the particular time should not have been observed.

3.6.4 Principal Components of the Reflood Curve

The time-dependent clad temperature, pressure drop, and liquid carryover were decomposed in their respective functional principal components (fPCs) to better quantify the mode of variations of the whole time-dependent curves. Therefore, the variance decomposition was also carried out on the fPC scores associated with the fPCs. Because each fPC is associated with a particular mode of variation over the whole transient, it parsimoniously describes the overall variation of the time-dependent curve in a smaller set of numbers. The fPC analysis of all the time-dependent curves for each type of output showed that the first two respective fPCs account for more than 85% of the overall functional variations (see Fig. 3.12).

Explained variance

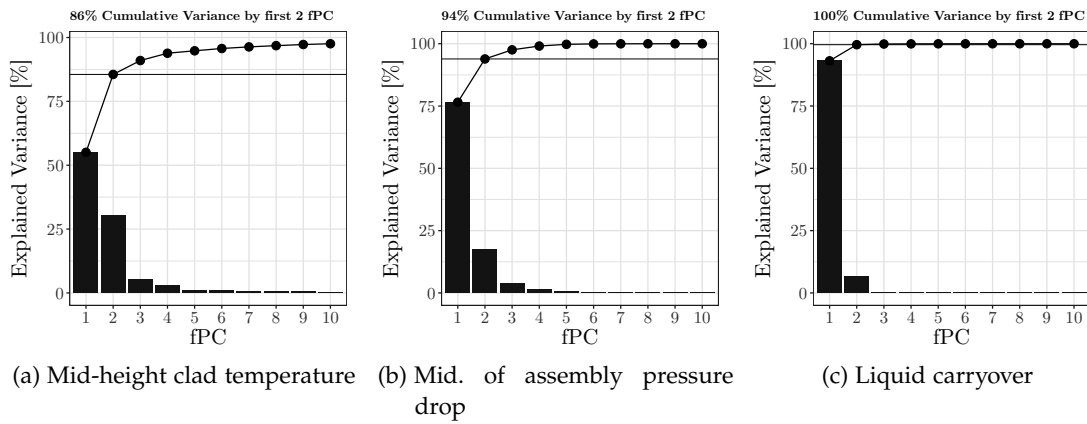


Figure 3.12: The proportion of explained variance for each fPC extracted from selected time-dependent outputs. The points with a connecting line are the cumulative explained variance, while the horizontal line is variance explained by first two fPCs.

The first fPC of the (registered) clad temperature at the mid-height of the assembly (in the reflood transient) and the effect of its perturbation around the mean function are shown in Fig. 3.13. The fPC shown in Fig. 3.13a was obtained by multiplying the eigenfunction $\xi_j(t)$ with the square root of the respective eigenvalue, i.e., $fPC_j = \sqrt{\rho_j} \times \xi_j(t)$. As the eigenfunction only represents the shape (mode) of function variation, this multiplication was done to give it a sense of scale with respect to the clad temperature variation (as $\sqrt{\rho_j}$ represents the standard deviation of the mode j). The perturbation around the mean function (Fig. 3.13b) is done by adding to and subtracting from the mean function, the eigenfunction multiplied by twice the corresponding score standard deviation $\sqrt{\rho_j}$, i.e., $\bar{y}(t) \pm 2 \times \sqrt{\rho_j} \times \xi_j(t)$.

1st fPC of the (registered) mid-height clad temperature transient

This particular fPC corresponds to a mode of variation that relates to the amplitude of the temperature reversal period (see Section 2.2). This is the strongest mode of variation, accounting for 55% of the overall clad temperature variation.

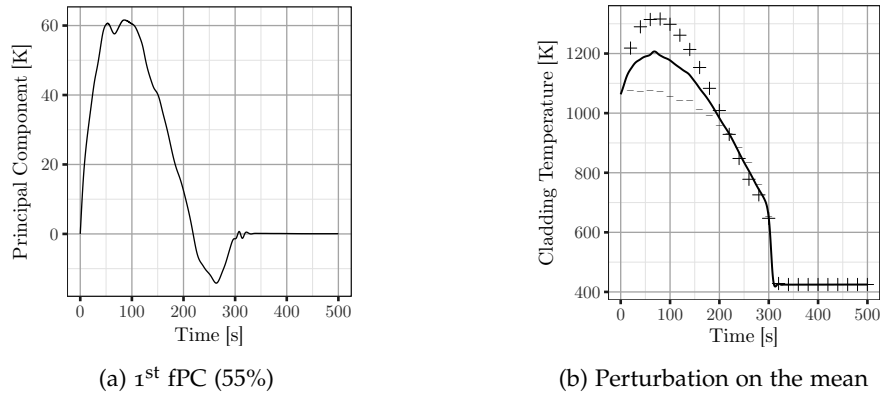


Figure 3.13: The 1st fPC of the (registered) mid-height clad temperature transient and the effect of its perturbation on the mean function.

2nd fPC of the (registered) mid-height clad temperature transient

Fig. 3.14 shows the results for the second fPC of the (registered) mid-height clad temperature transient and the effect of its perturbation on the mean function. This mode relates to the variation in the temperature descent after reaching the maximum temperature, prior to quenching. Visibly, some realizations tend to bring about more convexity in the temperature descent than others. This mode of variation constitutes 30% of the overall variation.

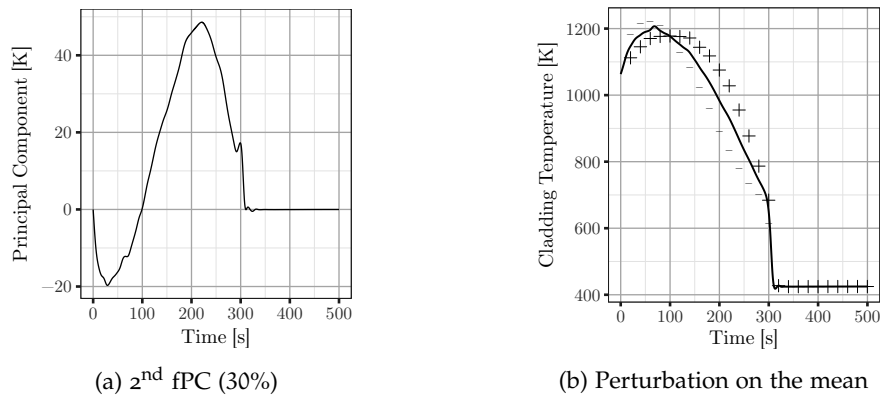


Figure 3.14: The 2nd fPC of the (registered) mid-height clad temperature transient and the effect of its perturbation on the mean function.

1st fPC of the warping function for the mid-height clad temperature transient

The previous fPCs were carried on the registered clad temperature transient where the phase variations in the data set have been removed. It is also interesting to see the phase variations in the data set separately. This can be done by carrying out the same procedure on the resulting warping functions associated with each clad temperature realization. Fig. 3.15 shows the 1st fPC and the effect of its perturbation on the mean function. The mode corresponds to the overall shift in the timing of the two landmarks compared to the mean func-

tion. From the figure, a delay in the maximum temperature tends also to result in a delay in the time of quenching, and vice versa. However, the variation in the time of the maximum temperature tends to be much smaller than the variation in the time of quenching.

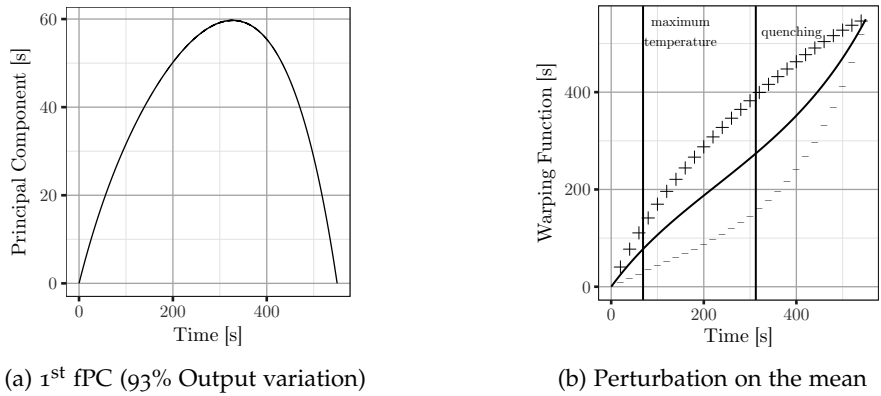


Figure 3.15: The 1st fPC of the warping function of the clad temperature transient at the mid-height of the assembly and the effect of its perturbation on the mean function.

The first fPC of the pressure drop transient curves at the middle of the assembly is shown in Fig. 3.16. The fPC, taking into account 77% of the output variation, is mostly responsible for the variation during the pressure drop rise, where the channel segment is continuously quenched from the bottom. That is, some realizations rise more quickly (or more slowly) in reaching the equilibrium pressure drop.

1st fPC of the pressure drop transient at the middle of the assembly

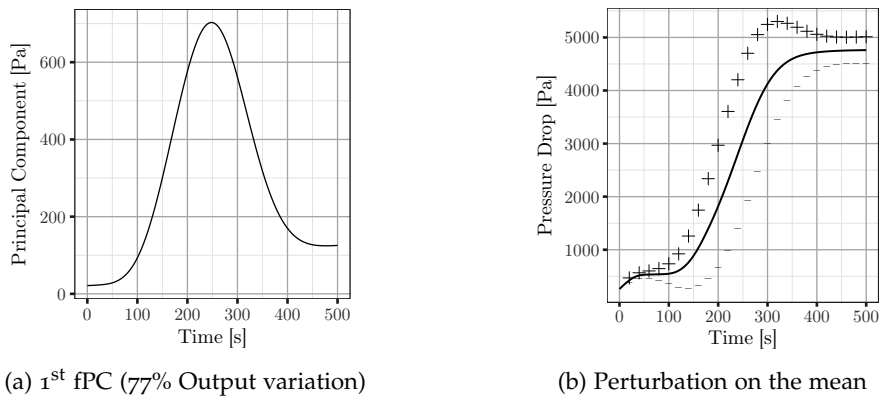


Figure 3.16: The 1st fPC of the pressure drop transient at the middle of the assembly and the effect of its perturbation on the mean function.

The first fPC of the liquid carryover transient curves, shown in Fig. 3.17, are straightforward to interpret. The fPC, taking into account 93% of the output variation, is the linear change of the average

1st fPC of the liquid carryover transient

liquid carryover during the transient. In other words, the perturbation on the liquid carryover is accumulated linearly over time.

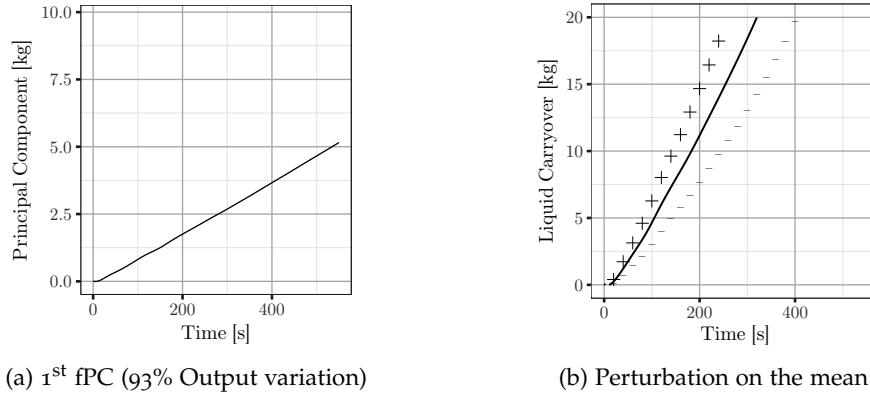


Figure 3.17: The 1st fPC of the liquid carryover transient and the effect of its perturbation on the mean function.

3.6.5 Sobol' Indices for QoIs based on Principal Components

The fPC score $\theta_{n,i}$ associated with each realization n and a principal component i is used as the QoI in a SA similar to what was done for the conventional QoI in Section 3.6.3. In other words, the variance of the score is decomposed into the variance contribution associated with each input parameter.

The estimated Sobol' indices for the first fPC of the (registered) mid-height clad temperature transient are given in Fig. 3.18. As shown, the variation in the amplitude of the temperature reversal was mainly due to the spacer grid heat transfer enhancement and the DFFB-related model parameters. The other parameters are proved to be almost noninfluential. This result is consistent with the result obtained when using the maximum clad temperature as the QoI and confirms the maximum clad temperature as a viable representative QoI during the temperature reversal period.

Fig. 3.19 shows the Sobol' sensitivity indices using the scores associated with the second fPC of the (registered) mid-height clad temperature transient as the QoI. Contrary to the first component, most of the variation in the second fPC can only be explained through interactions between input parameters since the main-effect indices only summed up to 27% of the total variance. The difference between the main-effect and total-effect indices are large for all input parameters, especially for the DFFB-related parameters. These parameter interactions, associated with a particular mode of variation, could not be captured from the conventional QoIs (e.g., the maximum clad temperature). It could only be speculated from the time-dependent rep-

Sobol' indices for the 1st fPC of the mid-height clad temperature transient

Sobol' indices for the 2nd fPC of the clad temperature transient

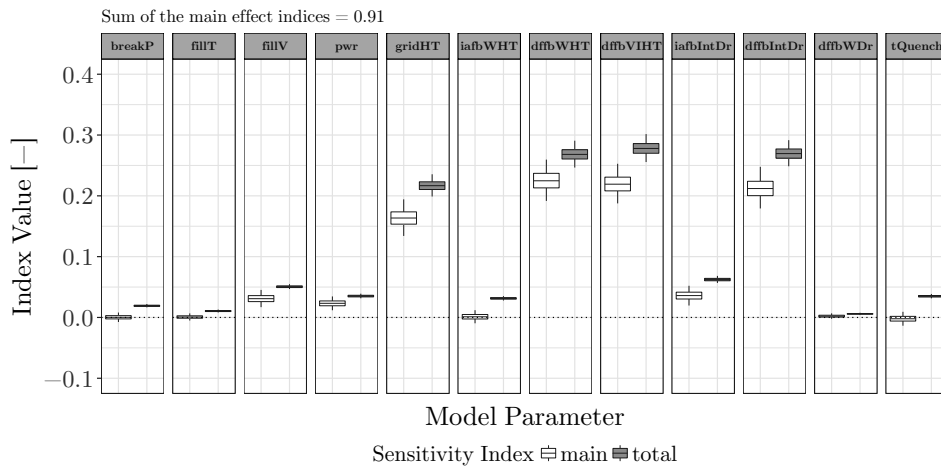


Figure 3.18: The sensitivity indices with respect to the 1st fPC of the (registered) mid-height clad temperature transient. Each boxplot represents the bootstrap sample quartile statistics and the vertical line extends the 95th sample percentile.

resentation of the sensitivity indices showed in Fig. 3.11, but with a less concise description of the parameter sensitivity.

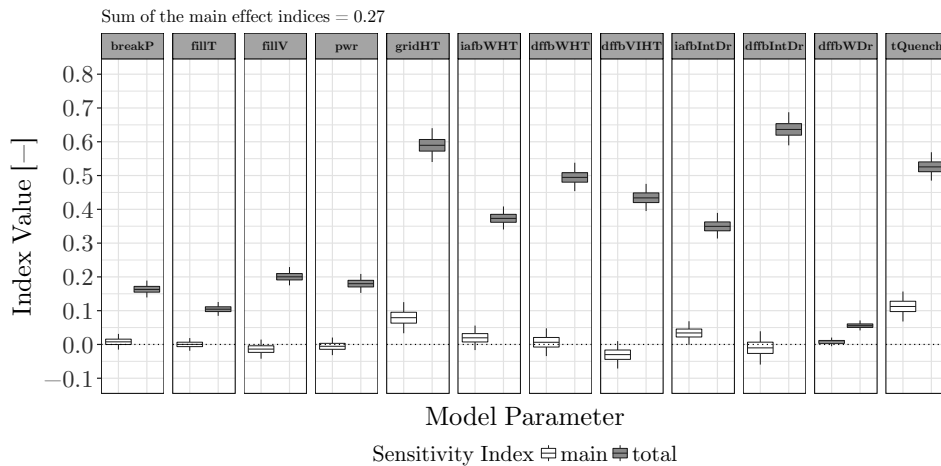


Figure 3.19: The sensitivity indices with respect to the 2nd fPC of the (registered) mid-height clad temperature transient. Each boxplot represents the bootstrap samples quartile statistics and the vertical line extends the 95th sample percentile.

The sensitivity indices with respect to the first fPC of the warping functions for the mid-height clad temperature transient are shown in Fig. 3.20. The spacer grid heat transfer enhancement parameter is the main source of variation in the time-shift of the landmarks, while two DFFB-related parameters (dffbWHT and dffbIntDr) and the quenching temperature each contributes around 10% to the total output variation. In comparison, the rest of the parameters have a trivial effect to the shift. Only a small portion of the output variation is due to parameter interactions from the fact that the main-effect indices summed up to a value close to 1.0 (94%).

Sobol' indices for the 1st fPC of the warping function for the clad temperature transient

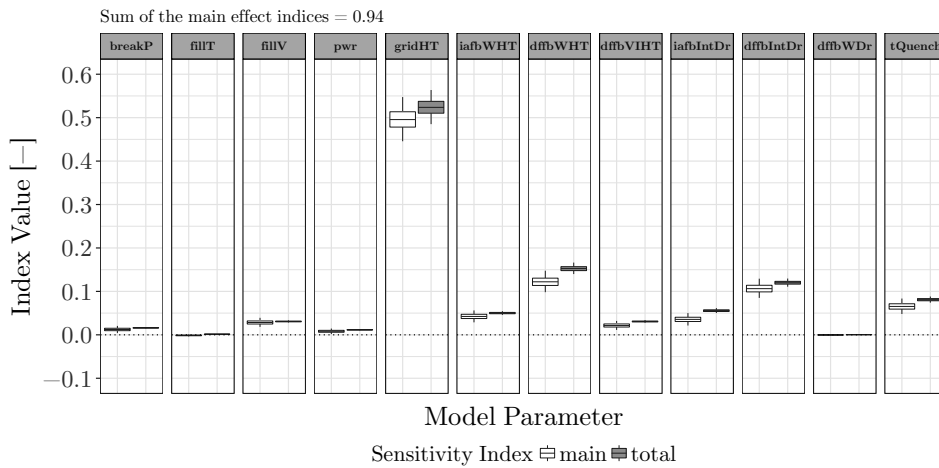


Figure 3.20: The sensitivity indices with respect to the 1st fPC of the warping function for the mid-height clad temperature transient. Each boxplot represents the bootstrap sample quartile statistics and the vertical line extends the 95th sample percentile.

Fig. 3.21 presents the sensitivity indices with respect to the scores associated with the first fPC of the pressure drop transient at the middle of the assembly. The inlet mass velocity parameter (fillV) is the main contributor to the overall output variation (~ 30%), while the two interfacial drag parameters of the reflow model amount to the same combined contribution.

Sobol' indices for the 1st fPC of the pressure drop transient

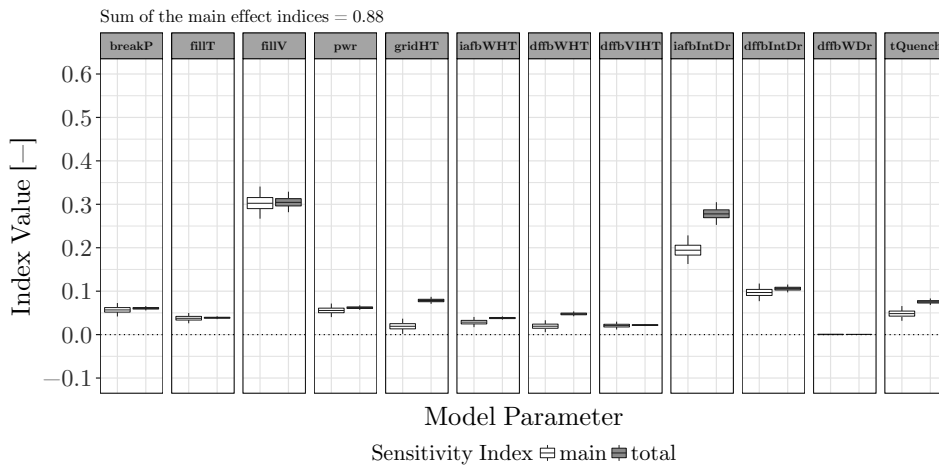


Figure 3.21: The sensitivity indices with respect to the 1st fPC of the pressure drop transient at the middle of the assembly. Each boxplot represents the bootstrap sample quartile statistics and the vertical line extends the 95th sample percentile.

Finally, Fig. 3.22 shows the sensitivity indices with the first fPC of the liquid carryover transient as the QoI. The inlet mass velocity parameter (fillV) is by far the main source of variation in the output variation (~ 90%), followed by minor contributions (~ 9%) from two DFFB-related parameters (dffbVIHT and dffbIntDr), and the rest of the parameters have a negligible effect.

Sobol' indices for the 1st fPC of the liquid carryover transient

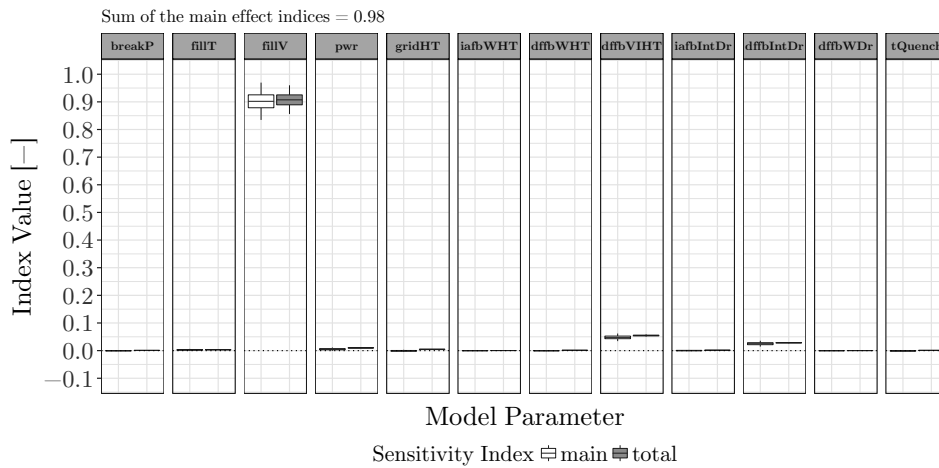


Figure 3.22: The main-effect and total-effect sensitivity indices with respect to the 1st fPC of the liquid carryover transient. Each boxplot represents the bootstrap sample quartile statistics and the vertical line extends the 95th sample percentile.

The numerical results of the estimated Sobol’ indices presented above are tabulated in Table B.14 through Table B.20 in Appendix B.4. To give a measure of uncertainties on the estimates due to MC sampling, the results in the tables are complemented by the 95th bootstrap percentile confidence interval CI_{pct} [155].

3.6.6 Discussion

The Morris screening method was used to filter out noninfluential parameters from further analysis (Table 3.2). It was shown that such reduction was valuable to the downstream analysis by reducing the size of the problem (i.e., the number of parameters). The screening results with respect to the average temperature showed that most of the model parameters related to the IAFB regime have relatively lower importance than the ones related to the DFFB regime. This finding confirms that the implementation of the reflood models in TRACE is consistent with the widely accepted phenomenological view on the relevance of DFFB for heated channel reflooding at low flooding rate [156]. Intuitively, most drag related parameters becomes more prominent with respect to the average pressure drop output, though correlation between outputs does not exclude the common importance of heat transfer-related parameters. Finally, with respect to average liquid carryover, only four parameters were found to be important. It is also in accordance with the expected simulated physical process.

Screening analysis results

Those findings also illustrate the fact that the Morris screening method could serve as a preliminary analysis of model development to verify if the model behaves (in terms of parameters importance) as expected with limited number of code runs. In Ref. [153] the Morris method was used with this perspective in mind.

*Morris with radial
design vs. trajectory
design*

A comparison between the importance rankings obtained by the two Morris screening method variants showed a consistent result. The radial design, however, exhibits more erratic variations in the elementary effect statistics estimations and thus requires slightly more replications (thus code runs) to stabilize. This is due to the fact that, in the radial design, grid jump varies from replication to replication and from parameter to parameter excluding the possible bias due to an unexplored area of the input parameter space. The trajectory design, in contrast, uses a constant grid jump which constrains the possible parameter perturbation. Increasing the number of replications while keeping the same grid jump might give an impression that the elementary effects statistics converge quickly, especially if the grid jump is relatively large. Thus, to exclude this source of bias, different sizes of grid jump should be considered before a more robust conclusion on the ranking can be drawn. This, however, entails more code runs.

*Utility of Sobol'
total-effect indices*

The elementary effects statistics, however, are deemed qualitative as they do not quantify exactly the contribution of the parameters variations to the output variations. The comparison between two parameters whose value of the first μ^* is larger than the second is hard to intuit beyond the fact that the first parameter is relatively more important than the second. In this regard, the Sobol' total-effect indices were found to be useful for screening application in a more quantitative manner, but required more code runs as compared to the Morris method ($\sim 3'000$ vs $\sim 7'000$). As explained, the total-effect index of a parameter is the proportion of output variance due to the variation of the parameter, including all the possible interactions of any order with any other input parameters. A parameter with low total-effect index implies that the parameter is simply less influential with respect to the selected output. By setting a cut-off value, a parameter was classified as either influential and noninfluential in a quantitative and consistent manner with reference always to the same output variance. Nevertheless, the selection of the cut-off value is admittedly subjective and the results need to be further verified. This was done through uncertainty propagation using influential and noninfluential parameter subsets which is presented in Fig. 3.8.

*Sensitivity with
respect to
conventional QoIs*

With respect to the Sobol' indices, the parameters driving the variations of the maximum clad temperature and the time of quenching were found to be different (Figs. 3.9 and 3.10). Since the two events occurred at two separate instants of the reflood transient, the results indicated that the shape of the temperature curves varied in a complicated manner. This variation, however, was insufficiently characterized by the two conventional reflood QoIs. Indeed, the importance of the model parameters varied across the transient (Fig. 3.11).

*Time-dependent
sensitivity*

The depiction given in Fig. 3.11 might give a misleading impression that the parameters themselves were time-dependent or were being perturbed at different times in the transient. This was not the case;

the parameters were constant and perturbing them at the start of the transient will affect the whole course of the transient. To increase the interpretability of the effect of parameter perturbation, different features of the transient output variations were explored using fPCA.

The model parameters influenced the amplitude of the clad temperature reversal of the reflood transient (Fig. 3.13) with minor interactions among themselves (Fig. 3.18). These parameters were mainly related to the spacer grid heat transfer enhancement model [128] and the DFFB-related heat and momentum transfer parameters [28]. As the model was found to be largely additive with respect to this part of the transient, temperature data from the experiments could in principle be used to inform these parameters although such an application would require further investigation (e.g., in the case of colinearity between these parameters).

On the other hand, the temperature descent up to and around the time of quenching (Fig. 3.14) proved to be influenced by interactions between parameters (Fig. 3.19). From the reflood modeling point of view, this can be explained by the fact that the temperature descent (which occurs at later stage of the transient) is more affected by flow regime changes. This observation is inferred from the total-effect index for two parameters of the IAFB flow regime which was found to be no less influential as compared to the relevant DFFB-related parameters. Indeed, the conditions and the criteria leading to the changes from one regime to another in the TRACE code depend indirectly on the simultaneous perturbation of these parameters.

From a numerical point of view, this can also be explained by the fact that the variance of the clad temperature transient tends to grow over time up to the quenching. As such, any given parameter perturbation which has a minimal impact at the early phase of the transient might interact with the others and accumulate their small effects over time and later be responsible to the growing variance of the output.

The existence of parameter interactions also marks the the fact that hydrodynamic processes (e.g., wall and interfacial drags) are indeed coupled with heat transfer processes (e.g., wall and interfacial heat transfers) in the TH system code TRACE mainly through the void fraction [157]. Thus, the simulation of a reflood process can be expected to reflect this coupling. It does, however, also complicate the task of model parameters calibration if done solely on the basis of temperature transient data because multiple combination of parameter values might give a similar clad temperature prediction at this particular phase of the transient. Hence, to better inform the model, additional types of data associated with different types of outputs (e.g., pressure drop and liquid carryover) should be considered.

And although it was shown that a high degree of parameter interactions existed with respect to this particular mode of variation, the nature of these interactions among the parameters is still poorly

*Mid-height clad
temperature
transient, variation
and sensitivity, 1st
fPC*

*Mid-height clad
temperature
transient, variation
and sensitivity, 2nd
fPC*

*Parameter
interactions*

*Sobol' indices,
second-order*

known. The estimation of the second-order Sobol' sensitivity indices would be required. These indices can give a clearer picture on the actual structure of the parameter interactions. In relation to this, one can notice the analogy between the different phenomenological phases of the reflood curve defined in the FEBA evaluation report [123], namely the mist cooling, the film boiling, and the quenching phase, with the three fPCs empirically obtained from FDA, namely the temperature reversal, the temperature descent, and the quenching (the third fPC is not discussed in this thesis for conciseness but is shown in Ref. [142]).

In other words, GSA using FDA-based QoIs concisely and quantitatively shows how the effect of the entrained droplets (mist) on the clad temperature, which is implicitly captured by the DFFB-related parameters, dominates the variation of the clad temperature during the mist cooling phase (as labeled by the FEBA experimentalists). Furthermore, a more intricate picture can be inferred during the film boiling phase, which happens at a later phase and may relate to the interpolation in TRACE between the DFFB and the IAFB regimes (i.e., the inverted slug regime).

*Warping function
for the clad
temperature
transient, variation
and sensitivity*

The analysis of the clad temperature transient presented above was conducted after the phase variations in the timing of the two reflood landmarks (i.e., maximum temperature and quenching) were removed through registration. The variations of the warping functions were separately analyzed (Fig. 3.15) and their sensitivity indices were derived (Fig. 3.20). The results showed that the parameter responsible for the time shift of the two reflood landmarks was mainly the one related to the spacer grid heat transfer enhancement model. This is consistent with the results obtained using the time of quenching as the QoI. However, Fig. 3.15 also succinctly presented the finding that although a delay in the time of the maximum temperature implies a delay in the time of quenching, the variation of the former was much smaller than the variation of the latter.

*Pressure drop
transient, variation
and sensitivity*

The variation of the pressure drop transient at the middle of the assembly (Fig. 3.21) was mainly related to the rate of pressure drop rise along the segment. Following the sensitivity analysis result (Fig. 3.21), the interfacial drag of the IAFB regime became relatively influential along with the inlet mass flow rate boundary condition. This was not the case for the clad temperature outputs at different axial locations and was found to be consistent across all pressure drop outputs. As such, it might be worthwhile to consider pressure drop output for model calibration, especially for the parameter *iafbIntDr*.

*Liquid carryover
transient, variation
and sensitivity*

Finally, the functional variation of the liquid carryover transient (Fig. 3.17) showed that the variation in liquid carryover transient was straightforward to interpret, either faster or slower rate in comparison with the mean. The sensitivity analysis (Fig. 3.22) results are reasonable in the sense that the variation could be attributed to the variation in the amount of liquid (and droplets) being transported (as repre-

sented by the inlet mass flow boundary condition (`fillV`) and the interfacial drag (`dffbIntDr`) as well as the variation in the amount of droplets being evaporated (as represented by the interfacial heat transfer parameter (`dffbVIHT`)). However, the analysis showed that the inlet mass flow rate boundary condition was much larger than the two reflood model parameters. This puts into question the value of liquid carryover data to calibrate the two reflood model parameters under the uncertainty of inlet mass flow rate boundary condition whose variability is assumed to be irreducible.

All in all, the sensitivity indices obtained confirms the consistency of the phenomenological reflood model implemented in TRACE in simulating an experimental reflood transient. Moreover, it has been shown here for the first time how the variability in the parameters relevant to the simulation of the reflood phenomena affects the output and to what extent. These quantitative aspects have been confirmed for different types of QoIs and for different types of outputs.

These results can be compared, to a certain degree, to Ref. [158]. There, the SA was also carried out for the same problem (FEBA experiment) using the same code (TRACE). Yet, the difference in the sensitivity measures (based on the Pearson product-moment correlation), the difference in the choice of parameters, and the difference in the a priori ranges of variations for the parameters make direct comparison between the two studies difficult. This underlines the problem faced in using a global statistical framework for SA; the choices of the parameters as well as the assumed range of variations to derive a sensitivity measure have to be consistent across different studies for the obtained measure to be comparable. These differences, in turn, might be due to the different objectives of the respective studies.

3.7 CHAPTER SUMMARY

The global sensitivity analysis (GSA) methodology part of the proposed statistical framework has been presented in this chapter. The objective of GSA was to increase the understanding of the relationships between model input parameters and time-dependent output, within a selected region of interest in the input parameter space. This understanding is beneficial for the follow-up work presented in Chapters 4 and 5. In Chapter 4, a statistical metamodel is constructed based only on the influential parameters, thus avoiding an unnecessarily large number of training samples associated with a large input parameter space. In Chapter 5, the initial range of variations for (some of) the input parameters assumed in this chapter are updated based on available experimental data. There, the results of the SA can provide some ideas as to which parameters can be informed (the sensitive ones), cannot be informed (the insensitive ones), or will present

possible complications (the interacting ones) when considering the available experimental data.

In accordance with the aim of increasing this understanding, a novel set of QoIs was derived using FDA techniques to characterize the overall functional output variation. This allowed us to capture the most essential features of the model behavior through its time-dependent output, thus significantly departing from the more conventional *ad hoc* QoIs (e.g., minimum, maximum, or time-average scalar value) that have been used so far in similar SA studies of nuclear reactor evaluation models.

The methodology was applied to the running case study of the simulation of a reflood experiment conducted at the FEBA facility using TRACE. The value and limitation of screening methods were first demonstrated for this type of application. Although the two variants of the Morris method yielded similar results with relatively small number of code runs, the Sobol' total-effect indices (also estimated with small number of runs) provide a more quantitative approach to screen the noninfluential parameters.

The noninfluential parameters were then excluded from a detailed variance decomposition. The results were consistent with the expected phenomenological behavior of the reflood model implemented in the TRACE code. The method was successful in apportioning the variation of scalar physical outputs (the maximum temperature and time of quenching) to the variation of the input parameters.

When considering FDA-based QoIs, which better represents the whole transient of selected outputs (clad temperature, middle pressure drop, and liquid carryover), it was found that the important parameters and the nature of their interactions were changing during the transient. For instance, during the early phase of the transient (when the temperature was increasing and during the early reflooding phase), the simulation model showed weak interactions among the prominent parameters (namely, the parameters related to the spacer grid HT enhancement model and the DFFB regime). But, during the temperature descent and around the quenching, most of the variation in the clad temperature transient can only be attributed to parameter interactions. The nature of these interactions, however, remains to be investigated and is outside the scope of this thesis.

Lastly, this chapter demonstrates the added value of the proposed FDA-based QoIs for GSA of transient simulation models. The provided example demonstrates that considering different outputs of the same transient or different aspects of the same output (as described by different QoIs) can highlight different model behaviors with respect to the input/output relationship. This confirms the selection of pertinent QoIs as one of the most crucial steps in a GSA.

GAUSSIAN PROCESS METAMODELING: EMULATING CODE INPUTS/OUTPUTS FOR FASTER EVALUATION

Under Bayesian calibration framework, tens, if not hundreds, of thousands code runs are to be expected to appropriately explore the posterior probability distribution using different values of the input parameters. Such a large number of runs are only feasible for simulation with a negligible running time. Therefore, to balance the need for vast number of code runs with the finite computing resources and time, an alternative approach is required to approximate the inputs/outputs relationship of the code for the selected relevant outputs within a selected input domain of interest.

This chapter describes an approach to construct a fast surrogate model (metamodel) that approximates (or emulate) the inputs/outputs relationship of an expensive code for faster evaluation at any given input parameters values located in the specified domain. As argued in Section 1.4.2 this thesis used the one based on Gaussian stochastic process, which results in a statistical metamodel. A statistical framework of metamodeling along with necessary notational conventions are first presented in Section 4.1. The framework casts the problem of metamodeling as a problem of nonlinear regression where a set of limited actual code runs (with input parameters values judiciously selected) is used to predict the code output at any other input values.

Afterward, a review on several fundamental concepts of multivariate Gaussian random variable and Gaussian stochastic process is presented in Section 4.2. The section also establishes an intuitive connection between multivariate Gaussian random variable and Gaussian stochastic process. Section 4.3 then presents the formal Gaussian stochastic process formulation used for metamodeling followed by the important aspects of constructing it in Section 4.4. Section 4.5 specifically deals with an approach to tackle the case of code with multiple outputs.

The application of the metamodeling approach to the TRACE model of the FEBA facility is given in Section 4.6. The suitability of Gaussian process metamodel on the TRACE model is assessed using different choices made during the metamodel construction. The results and important findings of this step are subsequently presented and briefly discussed. Finally, Section 4.7 summarizes the chapter.

4.1 STATISTICAL FRAMEWORK

Regression problem

Consider a general *regression* problem: Given a deterministic computer simulator (which, in essence, is a function) $f : \mathbf{x} \in \mathbf{X} \subseteq \mathbb{R}^D \mapsto \mathbb{R}$ evaluated at \mathbf{DM} , an experimental design matrix $\{\mathbf{x}_n\}_{n=1}^N$, yielding N outputs $\mathbf{y} = \{f(\mathbf{x}_n) = y_n\}_{n=1}^N$, the objective of the regression is to compute (or *predict*) the value of $f(\mathbf{x}_o)$ with $\mathbf{x}_o \notin \mathbf{DM}$.

Training data, training samples, and training outputs

The set $\mathcal{D} \equiv \{(\mathbf{DM}, \mathbf{y})\} = \{(\mathbf{x}_n, f(\mathbf{x}_n) = y_n)\}_{n=1}^N$ of N observations is often referred to as the training data, though the term is used interchangeably with the training outputs \mathbf{y} . The experimental design matrix \mathbf{DM} introduced in the previous chapter is interchangeably referred to as the training samples, inputs, or points in this chapter. As before, the domain \mathbf{X} is often rescaled such that $\mathbf{x} \in [0, 1]^D$.

Emulator, surrogate model, and metamodel

To evaluate f at any given $\mathbf{x}_o \notin \mathbf{DM}$, the code of course can be simply run at that input. Unfortunately, the true underlying function $f(\circ)$ that produces \mathbf{y} itself might be too complex and expensive to evaluate. As such, the response surface of the function has to be reconstructed or estimated based only on the small training data set before the prediction is made. The estimated function is chosen to be a simpler function that can be evaluated much faster (such as polynomials). Although simpler, such an approximation should capture the most, if not all, important aspects of the inputs/outputs relationship of the true underlying function. This simpler, approximating function is often referred to as an *emulator, surrogate model, or metamodel*.

Gaussian process metamodel

In this thesis, the metamodel is represented using Gaussian Process (GP), following the seminal works of Sacks et al. [53, 59] and interpreted through a Bayesian perspective. The advantages of using GP to represent an unknown function are its ability to model a complicated multi-dimensional function with limited number of parameters [159] as well as the provision of prediction error estimate [35, 160]. Furthermore, being a statistical model based on a stochastic process, it fits the statistical calibration framework of computer model presented in the next chapter.

Two interpretations

The GP metamodel, like many statistical models, can be interpreted either in frequentist sense or Bayesian sense. In the frequentist sense, the stochastic process $\mathcal{Y}(\circ)$ is one particular realization of an unknown stochastic process. The prediction at a particular value of \mathbf{x}_o is made based on the process as estimated according to the training data. On the other hand, in the Bayesian sense, a Gaussian process is first set up as the prior for the stochastic process and the prediction of the output at \mathbf{x}_o is made based on the posterior (or conditional) process as updated by the training data¹.

¹ The frequentist case is the classical approach first developed as spatial interpolation tool in geostatistics by Krige dating back to the 1950s [60] and formalized by Matheron in the 1960s [61]. In fact, Kriging model (due to Krige) is the more popular term for GP metamodel. These two terms, Kriging model and GP metamodel, will be used interchangeably in this thesis.

Both of these interpretations give equivalent results. The subtle difference lies in the interpretation of prediction error. In the frequentist case, the error is defined as the mean squared of error between the prediction made by the estimated process and the (hypothetical) true process [161]; while in the Bayesian case the error corresponds to the *epistemic* uncertainty of the prediction conditional on the observed data. That is, though the underlying computer simulation itself might be deterministic, the uncertainty of the prediction at x_0 stems from the fact that the simulator was not actually run at that input and thus the output is not *known*. The Bayesian perspective, as argued in Refs. [4, 35, 160], gives a more intuitive interpretation of the prediction error. This perspective is illustrated in Fig. 4.1.

Prediction error

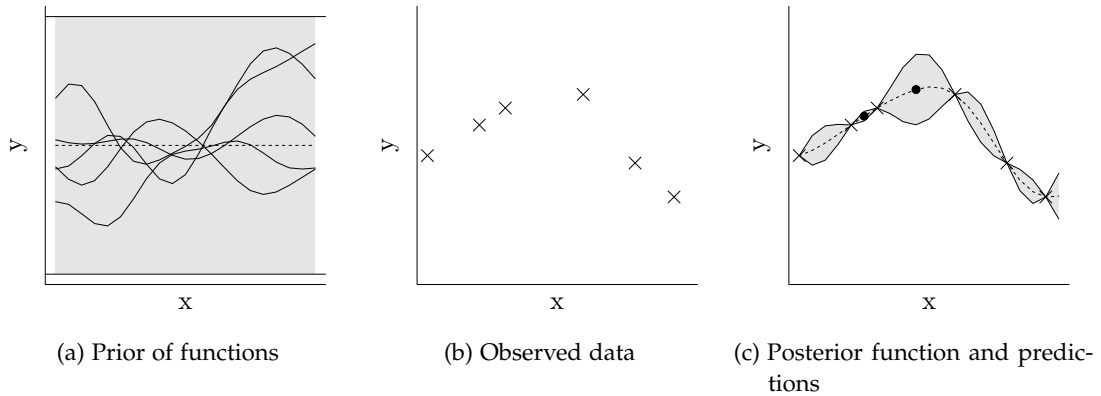


Figure 4.1: Gaussian process prior is equivalent to setting a prior over functions. After observing the data, the process is updated to obtain the posterior process with reduced uncertainties. Uncertainties are attached to each prediction made at arbitrary inputs which lie outside the observed data (e.g., black points). Dashed lines and gray region represent the mean and $3 \times \sigma$, respectively. The scales in the axes are arbitrary.

4.2 GAUSSIAN PROCESS FUNDAMENTALS

This section reviews the basics of GP. The connection between the stochastic process and multivariate Gaussian random variable (Gaussian random vector) is first established. Appendix D.2 gives some basic concepts of multivariate random variable such as joint, marginal, and conditional probabilities, while Appendix D.3 gives more detail on Gaussian random vector (Multivariate Normal (MVN)).

4.2.1 From Multivariate Gaussian to Gaussian Process

To illustrate the notions of joint, marginal, and conditional distributions, an example of a bivariate random variable, a Gaussian random vector $\mathcal{Z} = [z_1, z_2] \in \mathbb{R}^2$ is given. It has the following mean vector

Random vector, an example

and variance-covariance matrix, respectively,

$$\begin{aligned} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} &\sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \\ \boldsymbol{\mu} &= [0, 0]^T \\ \boldsymbol{\Sigma} &= \begin{pmatrix} \mathbb{V}[z_1] & \text{Cov}[z_1, z_2] \\ \text{Cov}[z_2, z_1] & \mathbb{V}[z_2] \end{pmatrix} = \begin{pmatrix} 0.5 & -0.265 \\ -0.265 & 0.25 \end{pmatrix} \end{aligned} \quad (4.1)$$

The joint, marginal, and conditional PDFs of random vector \mathbf{z} are illustrated in Fig. 4.2.

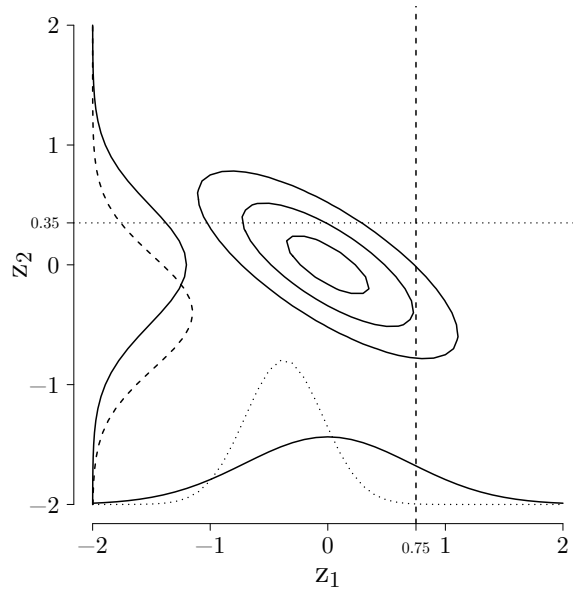


Figure 4.2: An illustration of bivariate Gaussian distribution of random vector $\mathbf{z} = [z_1, z_2] \in \mathbb{R}^2$ having marginal means of 0.0 and variances of 0.5 and 0.25, respectively and with covariance of about -0.265 . The solid ellipsoids indicate the contour of joint PDF of random vector $[z_1, z_2]$. The two solid curves at the x - and y -axes indicate the marginal PDF of z_1 and z_2 , respectively. The dotted curve shows the conditional density of random variable z_1 given $z_2 = 0.35$, while the dashed curve shows the conditional density of z_2 given $z_1 = 0.75$.

*Joint density,
illustrated*

The joint density for Gaussian random vector is given in Eq. (D.17). For the bivariate random variable in the example, the density can be shown as contour plot in Fig. 4.2. In the figure, the solid ellipsoids are the iso-contours of the distribution, where each pair of values lies along the contour line has the same probability density value.

*Marginal density,
illustrated*

The two marginal densities for the example are shown as the solid curves plotted in the x and y -axes, respectively. As illustrated, the marginalization of the joint distribution can be thought as a *projection* of the 2-dimensional distribution into each of the corresponding dimension.

Finally, two conditional distributions $p(z_1 | z_2 = 0.35)$ and $p(z_2 | z_1 = 0.75)$ are given as examples of conditioning a probability distribution in Fig. 4.2. They are shown as dotted and dashed curves plotted in both axes. Conditioning can be thought of as *slicing* the 2-dimensional distribution. Conditioning two correlated random variables on one, in general, changes the shape of the distribution of the other variable. From the figure, conditioning shifts the mean and reduces the variance of the resulting conditional distribution.

Conditional density, illustrated

GP can often be thought simply as a generalization of finite multivariate Gaussian random variable into an infinite multivariate one. To illustrate this idea, the marginal and conditional distributions of a 15-variate MVN distribution are plotted with the random variables at one common axis (x) while the range of values of the variables are plotted in another axis (y). This is practically an extension to the bivariate case exemplified before. The origin of the underlying 15×15 covariance matrix is at the moment unimportant, but what the matrix does is defining how the variables are correlated to each other. Fig. 4.3 shows the depiction. Fig. 4.3a shows the marginal distributions of random variables z_1 to z_{15} . Suppose the variables $z_2, z_4, z_7, z_9, z_{12}$, and z_{14} is observed (Fig. 4.3b). Now Fig. 4.3c shows the conditional distribution of the non-observed variables (the rest). As can be seen, the conditional probability of the non-observed random variables are shifted (from the zero-mean unconditional distribution) and their standard deviation are reduced.

An entry to Gaussian Process

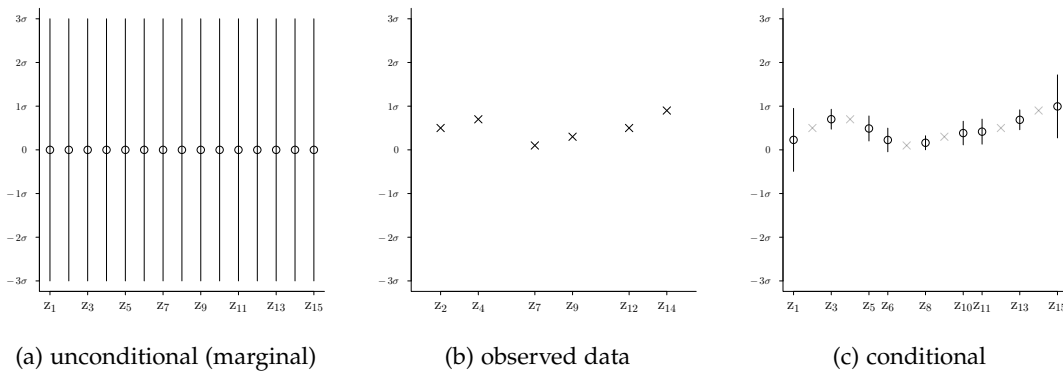


Figure 4.3: A 15-variate MVN random variable \mathcal{Z} . Prior to observing data, the mean and variance of each variable correspond to the marginal mean (in this case 0) and variance. Conditioning on the observed data shifts the mean and reduces the variance. Illustration is adapted from Ref. [162].

Gaussian stochastic generalizes this procedure beyond the 15-variate Gaussian random variable to an arbitrary number of variables at arbitrary locations on the real line. It is easy to imagine that the shape of both marginals and conditionals will become smoother and smoother with increasing number of random variables in the x -axis, thus re-

sembling more and more a smooth function. In fact, it is one of the interpretations of Gaussian process: a distribution over functions [4].

4.2.2 Gaussian Process

Stochastic process

Gaussian stochastic process is a particular class of *stochastic* or *random process*. Stochastic process is a collection of random variables, each of which are indexed with certain underlying rules or ordering. To be precise, a stochastic process is a set of random variables $\mathcal{Y} = \{y^{(i)}, i \in I\}$, where I is an index set, and it is defined on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$, where Ω , \mathcal{F} , and \mathbb{P} are the sample space, the set of events, and the assigned probability to the event, respectively [163].

Stochastic process applications

For example, a time series can be modeled using stochastic process where the random variables are the observations taken at different time ordered sequentially. In this case the index set is the time index of the observations. A spatial model, as another example, can be modeled as a collection of random variables indexed by their locations in space. And finally, in the metamodeling application, the random variables are collection of computational model output values at different input values.

Gaussian process

Gaussian stochastic process (GP, or Gaussian Random Field GRF) is defined as a collection of random variables \mathcal{Y} , *any arbitrary number of which is a multivariate Gaussian random variable* [62, 164]. To establish the connection with the notion of *random function*, the collection of the random variables \mathcal{Y} refers to the collection of values of a random function $\mathcal{Y}(\circ)$ at various possible input \mathbf{x} in the domain $\mathbf{X} \subseteq \mathbb{R}^D$. Specifically, $\mathcal{Y}(\mathbf{x})$, for $\mathbf{x} \in \mathbf{X} \subseteq \mathbb{R}^D$ is a *Gaussian process* if and only if for any choice from the finite set of input $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_L; L \geq 1\}$, the random vector $[\mathcal{Y}(\mathbf{x}_1), \mathcal{Y}(\mathbf{x}_2), \dots, \mathcal{Y}(\mathbf{x}_L)]$ is a multivariate Gaussian random variable [35].

A GP is fully specified by its mean and covariance functions, instead of a mean vector and a covariance matrix. A GP $\mathcal{Y}(\mathbf{x})$ on $\mathbf{X} \subseteq \mathbb{R}^D$ with a given mean function m and covariance K is denoted as

$$\mathcal{Y}(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), K(\mathbf{x}, \mathbf{x}^*)) \tag{4.2}$$

Mean function

The mean function of a Gaussian process $\mathcal{Y}(\mathbf{x})$ is the function $m : \mathbf{X} \subseteq \mathbb{R}^D \mapsto \mathbb{R}$ defined as,

$$m(\mathbf{x}) = \mathbb{E}[\mathcal{Y}(\mathbf{x})] \tag{4.3}$$

Covariance Function

The covariance function of a Gaussian process $\mathcal{Y}(\mathbf{x})$, on the other hand, is the function $K : (\mathbf{X} \subseteq \mathbb{R}^D) \times (\mathbf{X} \subseteq \mathbb{R}^D) \mapsto \mathbb{R}$ defined as,

$$K(\mathbf{x}_i, \mathbf{x}_j) = \text{Cov}[\mathcal{Y}(\mathbf{x}_i), \mathcal{Y}(\mathbf{x}_j)] \tag{4.4}$$

Notice that while the covariance function describes the covariance between pairs of random function values, it is defined only as a function of the two inputs, \mathbf{x}_i and \mathbf{x}_j . The covariance function is also

sometimes referred to as the *covariance kernel* function as it defines the elements of the covariance matrix (see example below). Not all functions of the pair of inputs $\mathbf{x}_i, \mathbf{x}_j$ are *valid* covariance functions, but only the ones that yield a valid variance-covariance matrix given by the condition in Eq. (D.21).

Finally, the process variance is defined as the covariance between two random function values at the same input,

$$K(\mathbf{x}_i, \mathbf{x}_i) = \text{Cov}[y(\mathbf{x}_i), y(\mathbf{x}_i)] = \mathbb{V}[y(\mathbf{x}_i)] \quad (4.5)$$

Process variance

For a given finite L , a GP is reduced to a Gaussian random vector with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$,

$$\begin{aligned} [y(\mathbf{x}_i)] &\sim \mathcal{N}_L(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \quad ; i = 1, 2, \dots, L \\ \boldsymbol{\mu} &= [m(\mathbf{x}_1), m(\mathbf{x}_2), \dots, m(\mathbf{x}_L)]^T \\ \boldsymbol{\Sigma} &= \begin{pmatrix} \mathbb{V}[y(\mathbf{x}_1)] & \cdots & \text{Cov}[y(\mathbf{x}_1), y(\mathbf{x}_L)] \\ \vdots & \ddots & \vdots \\ \text{Cov}[y(\mathbf{x}_L), y(\mathbf{x}_1)] & \cdots & \mathbb{V}[y(\mathbf{x}_L)] \end{pmatrix} \end{aligned} \quad (4.6)$$

The shape of the random function drawn from a GP is characterized by its mean and covariance functions. Brief explanations of these functions will be provided in the next two subsections. In the meantime, an example of a fully specified Gaussian process will be used to illustrate how samples of functions can be drawn from such a stochastic process. For the example, the following mean and covariance function will be used

Fully specified GP, an example

$$\begin{aligned} m(x) &= 0 \\ K(x, x^*) &= \sigma^2 \exp\left[-\frac{(x - x^*)^2}{2\theta^2}\right] = 10 \exp\left[-\frac{(x - x^*)^2}{0.98}\right] \end{aligned} \quad (4.7)$$

where x is a 1-dimensional input parameter such that $x \in [-2, 2]$. The mean function is set to constant zero, while the covariance function is chosen to be the so-called *Gaussian covariance function* (which will be detailed in the sequel). The Gaussian covariance function is parameterized by the characteristic length scale θ which is set to 0.70. This parameter is often referred to as the *hyper-parameter* of the function. Lastly, σ^2 is the variance of the stochastic process and it is set to 10.

To generate random draws of function from the fully specified GP given in Eq. (4.7), first it must be specified at which input x the function values are to be drawn. For the present example, x is chosen to be uniformly distributed $\{-2 + 0.2 \times i\}_{i=0}^{20}$. By specifying these locations, the 21-variate Gaussian random variable can be constructed using Eq. (4.6) with the elements of variance-covariance matrix computed by the formula in Eq. (4.7) for all pairs of inputs. Sampling from such a distribution can be done using algorithm outlined in Appendix D.5. Examples of five realizations from the GP are shown in Fig 4.4a. A

Sample path of a GP

realization of a GP on selected input locations is also called a *sample path* of the process [35], a term which will be used interchangeably with the term realization of a GP (or a stochastic process in general).

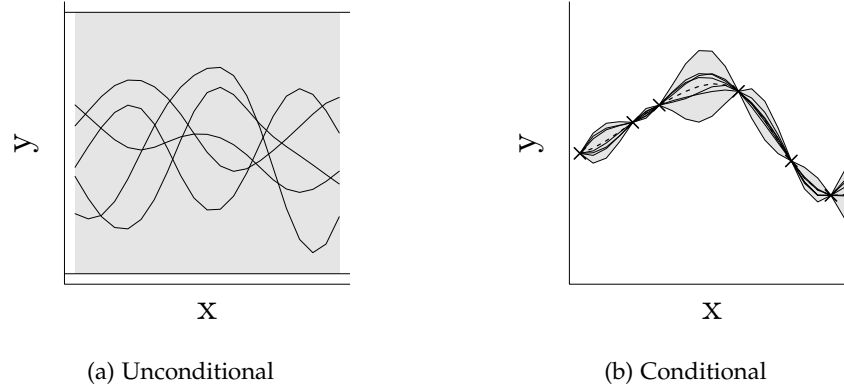


Figure 4.4: Five realizations (sample paths) of a Gaussian process specified in Eq. (4.7) at $\mathbf{x}_i = \{-2 + 0.2 \times i\}_{i=0}^{20}$. Shaded area indicates the area enveloped by twice standard deviation of the process (or 95% probability region). In the right panel, the sample paths are drawn conditional on six observed values (cross symbols).

Conditional sample path

Suppose now that values of six variables are fully observed as follows $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^6 = \{(-2.0, -0.75), (-1.2, 1.5), (-0.8, 2.75), (0.4, 3.75), (1.2, -1.3), (1.8, -3.8)\}$. The conditional 15-variate Gaussian distribution can be constructed in the same manner as before with the conditional mean and covariance following Eq. (D.24). Examples of five sample paths from such conditional distribution are shown in Fig. 4.4b. Observe that the standard deviations of the observed variables are zero and the gray areas between them are substantially reduced.

Strongly stationary process

An assumption for a class of stochastic process commonly made for convenience is *stationarity*. A stochastic process $\mathcal{Y}(\circ)$ is called *strictly/strongly stationary* if and only if for any finite set of inputs $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_L\} \in \mathbf{X} \subseteq \mathbb{R}^D$ with $L \geq 1$, and for $\mathbf{h} \in \mathbb{R}^D$ such that $\{(\mathbf{x}_1 + \mathbf{h}), (\mathbf{x}_2 + \mathbf{h}), \dots, (\mathbf{x}_L + \mathbf{h})\} \in \mathbf{X}$, the distribution of random vector $[\mathcal{Y}(\mathbf{x}_1 + \mathbf{h}), \mathcal{Y}(\mathbf{x}_2 + \mathbf{h}), \dots, \mathcal{Y}(\mathbf{x}_L + \mathbf{h})]$ is *the same* as the distribution of random vector $[\mathcal{Y}(\mathbf{x}_1), \mathcal{Y}(\mathbf{x}_2), \dots, \mathcal{Y}(\mathbf{x}_L)]$ [35, 165]. In other words, the process is invariant under translation.

Weakly stationary process

The *weakly stationary process* used additional weaker assumption than the strongly stationary process. A stochastic process $\mathcal{Y}(\circ)$ is called *weakly stationary* if and only if the first two moments of the process are constant. As such, the weakly stationary process is also referred to as *second-order stationary process*.

Stationary, isotropic covariance function

However, as mentioned before, a GP is fully defined by its mean and covariance functions. Therefore, if two GPs have the same mean and covariance functions defined over the same domain then the two

processes have exactly the same distribution and are the same process. For the case of GP, the notions of strongly stationary and weakly stationary coincide. This implies that a stationary GP has a constant mean and a constant variance, as well as a covariance function that satisfies the condition of being invariant under translation as follows,

$$\text{Cov}[y(\mathbf{x}_i), y(\mathbf{x}_j)] = \text{Cov}[y(\mathbf{x}_i + \mathbf{h}), y(\mathbf{x}_j + \mathbf{h})] = K(\mathbf{x}_i - \mathbf{x}_j) \quad (4.8)$$

In stationary GP, the covariance of random function values between two input points is only determined by the *distance* between the two inputs and the covariance function is called *stationary, isotropic covariance function* [62]. The notion of distance used in the above definition depends on the specific type of the covariance function as will be explained in the next subsection. Additionally, following Eq. (4.8), the process variance can be defined as the covariance at zero distance or $K(0)$, which is constant across input parameter space.

A more flexible class of GP models can be constructed by relaxing the stationarity assumption. However, stationarity is often assumed because it requires less assumption than the alternatives, considered non-informative, and therefore more generic [160]. Moreover, the stationary process remains important to study as they serve as building block for more advanced models [35]. For instance, the stationarity assumption can be relaxed simply by considering a non-constant mean function as proposed in Refs. [166, 167], while keeping the covariance part stationary. Another alternative is to consider multiple stationary covariance functions defined for each partitioned region of the whole input parameter space as proposed in Ref. [168].

*Non-stationary
process*

4.2.3 Covariance Kernel Function

Covariance kernel function determines the covariation structure of dependent data. This, in turn, determines the behavior (or shape) of the sample path of the outputs between input points. For a stationary covariance function, it is more convenient to separate the constant stochastic process variance σ^2 and the stochastic process kernel correlation function $R(\circ, \circ)$ between two input points using the following relation,

$$K(\mathbf{x}_i, \mathbf{x}_j) = \sigma^2 R(\mathbf{x}_i, \mathbf{x}_j) \quad (4.9)$$

where R , the correlation kernel function, is defined such that $\mathbf{x}_i, \mathbf{x}_j \in \mathbf{X} \subseteq \mathbb{R}^D \forall i, j$; and σ^2 is the aforementioned stochastic process variance, which determines the scale of variation magnitude of the output space.

In the following, three different types of stationary correlation kernel functions are presented. These functions, namely *Gaussian, power-*

*one-dimensional
correlation kernel, r*

exponential, and *Matérn class* kernels are widely applied in the simulation metamodeling literature. At first, only 1-dimensional kernel functions denoted by $r(x_i, x_j)$ are described. Later on, these 1-dimensional functions are used to create a multidimensional kernel function $R(x_i, x_j)$ by means of tensor product.

For each, the correlation function is defined and several sample paths are drawn to illustrate the effect of using different kernels as well as respective parameters on the sample path. It is important to note that it is a sample path of a stochastic process that is used as a metamodel and thus it is important to study its properties. For a stationary Gaussian stochastic process, only the correlation function determines the main properties of sample path, namely its continuity and differentiability (or smoothness). In particular, the continuity of a stationary correlation function at the origin guarantees the continuity of the sample path, and the smoothness of the correlation function determines the smoothness of the sample path. The mathematics behind these assertions is beyond the scope of this thesis, but an accessible reference on the topic can be found in [169].

4.2.3.1 Gaussian Kernel

The Gaussian correlation kernel function, also known as the *squared exponential* kernel, is given by the following formula [35, 62, 170],

$$r(x_i, x_j; \theta) = \exp \left[-\frac{(x_i - x_j)^2}{2\theta^2} \right] \quad (4.10)$$

The Gaussian kernel is parameterized by a single *hyper-parameter* θ that defines the characteristic length-scale of the process (or the *range* parameter). Fig. 4.5 shows the correlation value as function of Euclidian distance, $(x_i - x_j)^2$, between input points according to the Gaussian kernel, for three different range parameters. Obviously, for smaller θ the correlation between two inputs drops more quickly over shorter distance, and vice versa.

Characteristic
length-scale (range)
parameter

The range parameter of a Gaussian kernel determines the range over which the distance between two input locations affects the output correlation. To be precise, the notion of how similar (or dissimilar) two input locations are is defined relative to the characteristic length-scale. With a very short range, the output of random functions becomes easily uncorrelated except for a very close (similar) inputs. The realization of the process, therefore, will exhibit more erratic behavior at short ranges as it allows for changes that are more abrupt over shorter distance and less dependent of the neighboring values. On the other hand, with a longer range, the output of random function tends to be highly correlated except for very different input values and thus the realization will exhibit more rigid pattern. Gaussian kernel, however, always produces smooth realization. That

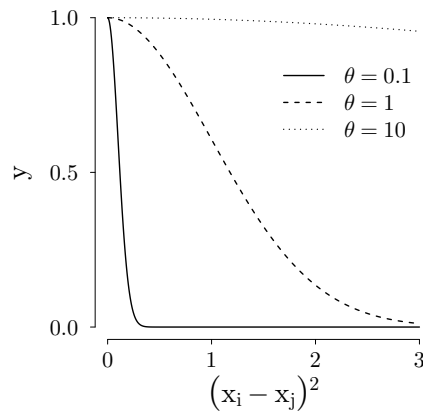


Figure 4.5: Examples of Gaussian correlation kernels with three different range parameters.

is, at any given point the Gaussian kernel is continuous and differentiable (see the neighborhood of the origin of Fig. 4.5). The Gaussian kernel is widely applied in the metamodeling literature and almost become a default choice for the correlation kernel [171], though as mentioned in Ref. [62] the overly smooth process can result in either physically unrealistic or numerically difficult situations (i.e., the resulting variance-covariance matrix is ill-conditioned for poorly selected design points).

Fig. 4.6 shows a comparison between realizations of a GP using Gaussian kernel for three different range parameters. The short range, illustrated on the left panel, allows for more sudden change in the output values while the long range on the right shows smoother (and rigid) pattern for the same input domain ($0.0 \leq x \leq 3.0$). Also notice that the realizations with the shorter range produces more local maxima and minima.

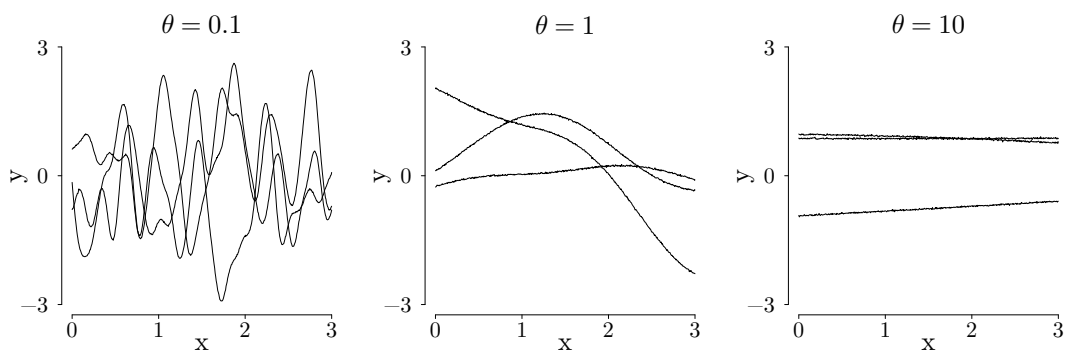


Figure 4.6: Examples of realizations from GP with Gaussian correlation kernel for three different values of range parameter. The plotting range of the y-axis for each panel are set to $\pm 3 \times \sigma$. Each process has the same process variance, $\sigma^2 = 1.0$

4.2.3.2 Power-Exponential Kernel

The Gaussian correlation kernel belongs to a wider class of 2-parameter kernel function family called the *power-exponential* kernel and is given by [35, 62, 170],

$$r(\mathbf{x}_i, \mathbf{x}_j; \theta, p) = \exp \left[- \left(\frac{|\mathbf{x}_i - \mathbf{x}_j|}{\theta} \right)^p \right] \text{ for } \theta > 0.0 \text{ and } 0 < p \leq 2 \quad (4.11)$$

The parameter θ remains the range parameter of the process, while the additional parameter p is referred to as the shape parameter of the process. Specifically, the shape parameter p determines the differentiability of the process at the origin [169]. Fig. 4.7 shows the correlation value of power-exponential kernel with three different values of p and θ as function of L_1 norm ($|\mathbf{x}_i - \mathbf{x}_j|$).

Shape parameter p

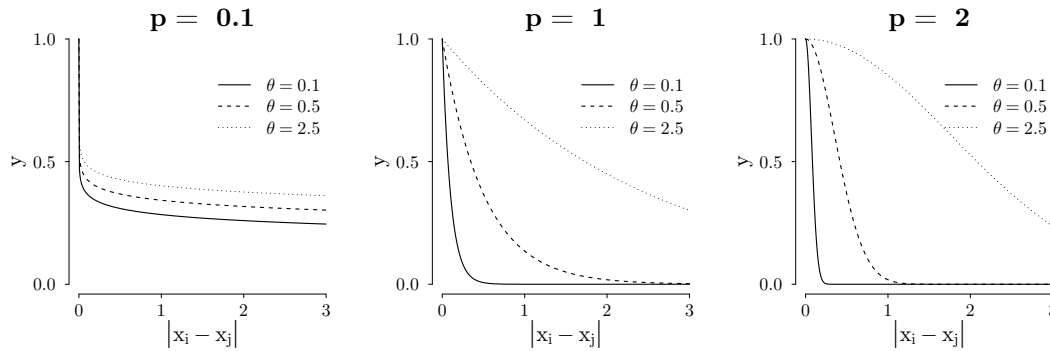


Figure 4.7: Examples of power exponential kernel functions for different values of shape parameter p and range parameter θ as function of L_1 norm.

Although, strictly speaking, only when $p = 2$ is the power-exponential correlation differentiable at the origin (thus guarantee the smoothness of the realization), the shape parameter dictates the apparent roughness of the sample path drawn from the process as illustrated in Fig. 4.6 [62].

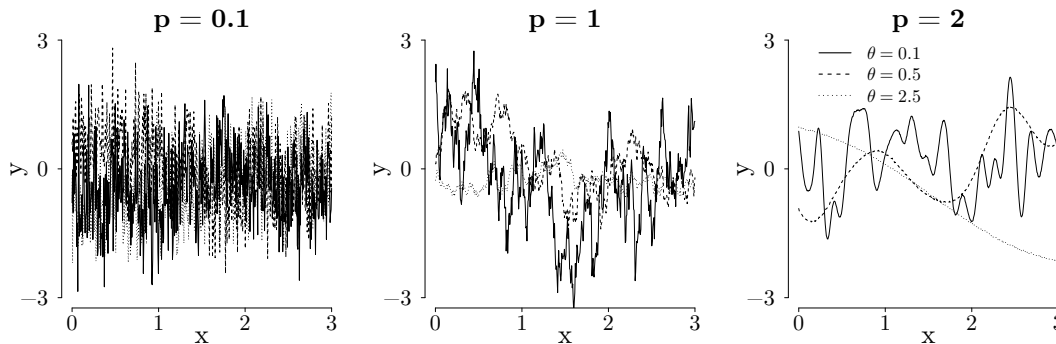


Figure 4.8: Several realizations from GP with power-exponential kernel functions of different shape p and scale θ parameters.

It is argued in Ref. [166] that the power-exponential kernel function is an appropriate choice in metamodeling application due to its flexibility of representing different shape with respect to its regularity and differentiability mainly controlled through the additional parameter p . For instance, Gaussian correlation kernel is a special case of the power-exponential kernel when p equals to 2. Another special case is when $p = 1$ which is called the *exponential* kernel. In this particular case, realizations of which are depicted in Fig. 4.8b, the process is continuous but not differentiable [62].

Exponential kernel

4.2.3.3 Matérn Class Kernel

The Matérn class correlation kernel is another 2-parameter kernel family and it is given by the following formula [35, 62],

$$r(x_i, x_j; \theta, \nu) = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{2\sqrt{\nu}|x_i - x_j|}{\theta} \right)^\nu K_\nu \left(\frac{2\sqrt{\nu}|x_i - x_j|}{\theta} \right) \quad (4.12)$$

where positive ν and θ are the correlation kernel parameters; $\Gamma(\circ)$ is the Gamma function; and $K_\nu(\circ)$ is the modified Bessel function of order ν . In the literature, the value of ν is often restricted to half integer $\nu = n + \frac{1}{2}; n \in \{0, 1, \dots\}$, because in that case the resulting modified Bessel function can be written simply as a finite series given by

$$K_\nu(t) = \exp(-t) \sqrt{\frac{\pi}{2t}} \sum_{k=0}^n \frac{(n+k)!}{k!(n-k)!} \frac{1}{(2t)^k} \quad (4.13)$$

The Matérn class is considered more flexible than the power-exponential kernel because the *shape* parameter ν directly controls the number of differentiability of the process [172]². However, it was argued in [62], that for machine learning application (i.e., regression and classification) only $\nu = 3/2$ (once differentiable) and $\nu = 5/2$ (twice differentiable) are of practical interest. This is due to the fact that for $\nu < 3/2$ the process becomes too rough³, while for $\nu \geq 7/2$ the smoothness of the process realization cannot be distinguished anymore from an even smoother process. These two Matérn correlation kernels are given below [62, 170],

shape parameter ν

$$r_{\nu=3/2}(x_i, x_j; \theta) = \left(1 + \frac{\sqrt{3}|x_i - x_j|}{\theta} \right) \exp \left[-\frac{\sqrt{3}|x_i - x_j|}{\theta} \right] \quad (4.14)$$

$$r_{\nu=5/2}(x_i, x_j; \theta) = \left(1 + \frac{\sqrt{5}|x_i - x_j|}{\theta} + \frac{5|x_i - x_j|^2}{3\theta^2} \right) \exp \left[-\frac{\sqrt{5}|x_i - x_j|}{\theta} \right]$$

² That was not the case for power-exponential kernel because, strictly speaking, the process is only differentiable at $p = 2$ (and in fact, infinitely differentiable).

³ In fact, it reduces to the exponential correlation function.

$$(4.15)$$

As the two previous kernels, the parameter θ serves as the range parameter of the process. Example plots of the Matérn kernel with different shape and range parameters are shown in Fig. 4.9.

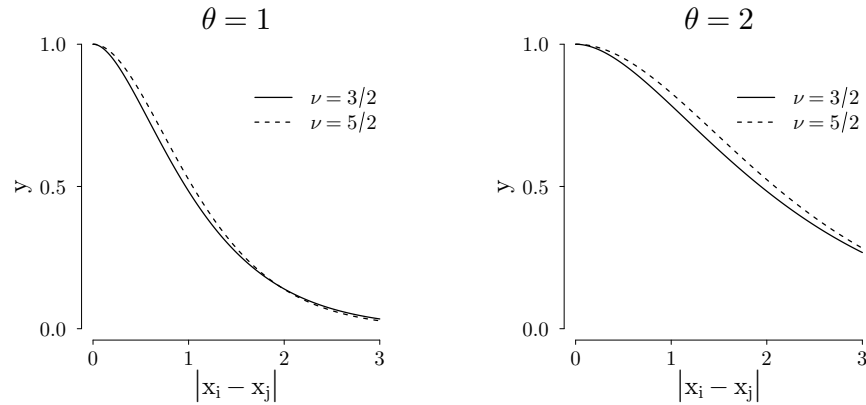


Figure 4.9: Matérn kernels for two different range parameters θ and, for each, two different shape parameters ν .

Examples realizations drawn from GPs with Matérn kernel with different shape and range parameters are shown in Fig. 4.10. As expected, the realizations from Matérn kernel with $\nu = 5/2$ is smoother than the ones from $\nu = 3/2$.

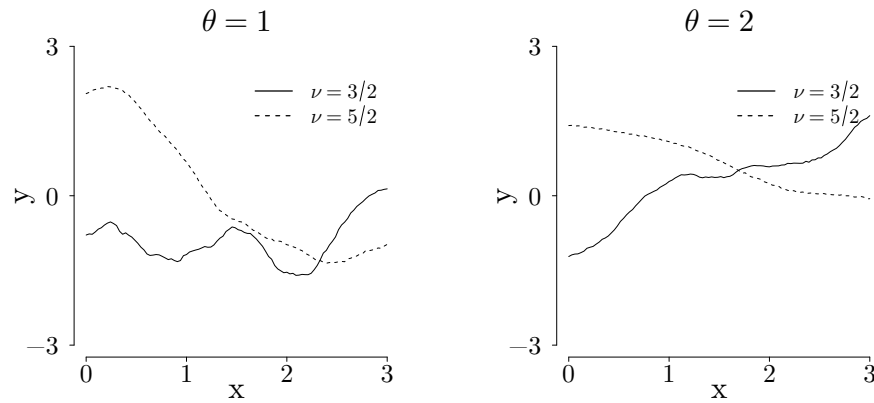


Figure 4.10: Example of sample paths drawn from GPs with Matérn kernel for different range and shape parameters. One realization is drawn from each combination of the parameters.

4.2.4 Multidimensional Construction

In order to create a valid multidimensional correlation kernel function from a valid 1-dimensional correlation function given above, a

Tensor product

tensor product construction is used as follows,

$$R(\mathbf{x}_i, \mathbf{x}_j) = \prod_{d=1}^D r_d \left(x_i^{(d)}, x_j^{(d)} \right) \tag{4.16}$$

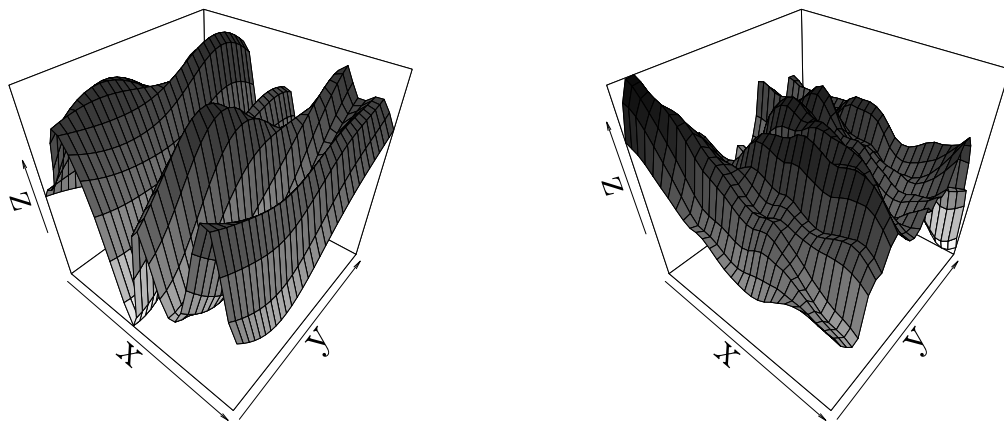
where r_d is a 1-dimensional correlation kernel function for the d -th input dimension; while $x_i^{(d)}$ and $x_j^{(d)}$ are a pair of values in the d -th input dimension.

Although it is possible to mix different types of correlation function or use different kind of multidimensional construction (see for example Ref. [173]), the tensor product with the same correlation function for each input dimension is the most well-established and, by far, the most popular approach in the applied metamodeling literature to date [35, 53, 59, 159, 160, 166, 170, 171, 174].

Fig. 4.11 shows two examples of realizations of random surface drawn from a multidimensional GP with the same process variance ($\sigma^2 = 10.0$) but with two different correlation kernels. On the left is an example of a realization drawn from the GP using two Gaussian correlation kernel functions in which the characteristic length scale in the y -direction is four times the scale in then x -direction. On the right is an example of a realization drawn from the GP using Matérn correlation kernel functions. For this case, the shape parameter is three times larger in the x -direction than in the y -direction. As such, for both cases, the surface appears less smooth in one of the direction.

Mixing kernels

Random surface



(a) Gaussian, $\theta_x = 0.5, \theta_y = 2.0$

(b) Matérn $\nu = 5/2, \theta_x = 1.5, \theta_y = 0.5$

Figure 4.11: Two random surfaces drawn from two different multidimensional GP with the same process variance of $\sigma^2 = 9.0$. Differences in the scale (for Gaussian) and shape (for Matérn) parameters for the inputs yield smoother path in one direction. The color scheme is the same for both plots with the range of $\pm 3 \times \sigma$.

4.2.5 Process Variance

For stationary GP, the shape of the sample path is determined solely by the form of the correlation. The role of the process variance according to Eq. (4.9) is to determine the scale of the magnitude of the output variation. Fig. 4.12 gives an illustration of the realizations drawn from a set of GPs with the same kernel correlation function (i.e., Gaussian kernel with $\theta = 1.0$), but with different values of process variance. As shown, the visible features of the realizations remain very similar to each other. What has changed, however, is the scale of the variation in the output space.

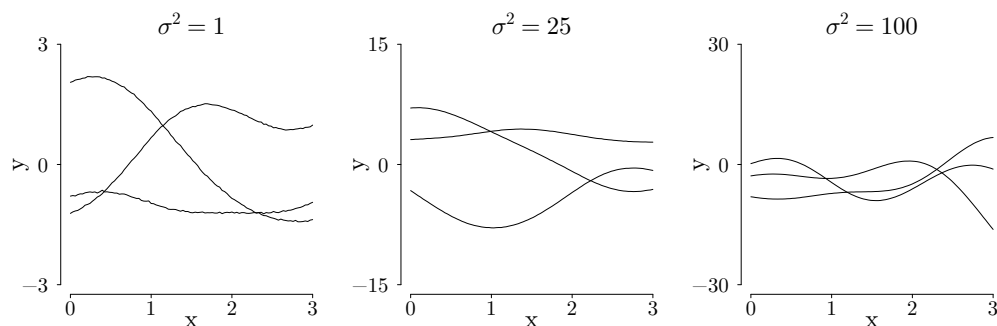


Figure 4.12: Realizations of GP with Gaussian correlation kernel for three different values of process variance. The plotting range of the y-axis for each panel is set to $\pm 3 \times \sigma$.

4.2.6 Mean Function

Mean function is the drift term in the GP model. Strictly speaking, incorporating other than a constant mean function to the specification of a GP introduces non-stationarity to the process. But as the known mean function can always be removed from the formulation (i.e., by *centering*), the process, especially with respect to its correlation function, can still be considered stationary. Fig. 4.13 shows several realizations drawn from three GPs having the same covariance kernel, but with three different mean functions. As it can be seen, the process are centered differently for the three GPs. The choice of the mean function determines the behavior of the conditional process (where it is constrained by the observed data) in the region far away from the available data.

The use of the mean function provides an opportunity to incorporate prior knowledge of the process before observing any data or to improve the resulting model performance for extrapolation purpose [162, 167]. However, without a very specific knowledge of how a process is expected to behave, it is difficult to completely specify a justifiable mean a priori. Indeed, it was argued in Ref. [160] that the use of either zero or constant mean in GP to model a process signifies a

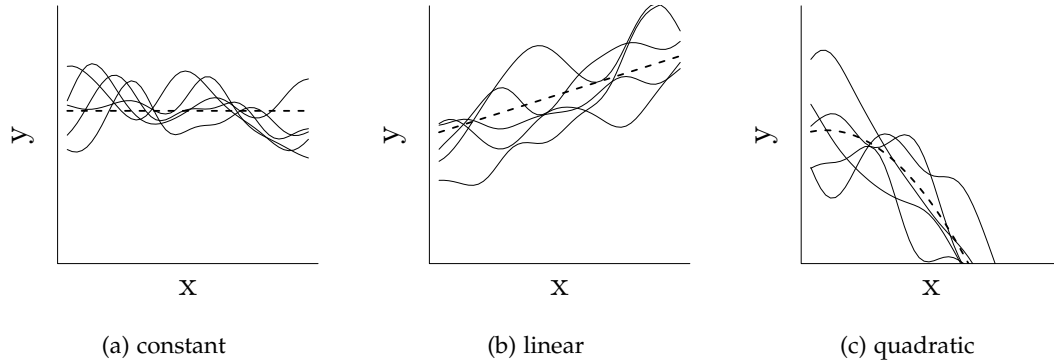


Figure 4.13: The effect of using three different mean functions (in dashed lines) on the realization of GP having the same covariance kernel (Gaussian). The scales in the axes are arbitrary.

vague or the least informative prior to the unknown. This eventually leads to the most generic formulation.

4.3 GAUSSIAN PROCESS METAMODEL

To formalize the use of GP in the metamodeling of a simulator, consider once again the regression problem of predicting the output at an arbitrary input $f(\mathbf{x}_o)$; $\mathbf{x}_o \notin \mathbf{DM}$ given $\{(\mathbf{DM}, \mathbf{y})\}$; where f , \mathbf{DM} , and \mathbf{y} are the function representing the simulator, the design matrix, and the training output, respectively. A GP metamodel makes the prediction as

$$\mathcal{Y}(\mathbf{x}_o) = \mu(\mathbf{x}_o) + \mathcal{Z}(\mathbf{x}_o) \quad (4.17)$$

The equation above, the *Kriging* model, consists of two components:

- The *mean/drift/trend* term, $\mu : \mathbf{x} \in \mathbf{X} \subseteq \mathbb{R}^D \mapsto \mathbb{R}$, is a deterministic function. The choice of the trend term distinguishes different classes of Kriging model. *Simple Kriging* (SK), refers to a class of Kriging whose arbitrary trend function is fully specified. *Universal Kriging* (UK), on the other hand, is a class of Kriging where a general polynomial model is assumed, but its coefficients are unknown [166, 167, 175],

Mean term

$$\mu(\mathbf{x}) = \sum_{j=0}^J \beta_j h_j(\mathbf{x}) \quad (4.18)$$

where h_j are polynomials basis functions; and β_j are the associated unknown coefficients. *Ordinary Kriging* (OK) is a special case of UK where the trend is set as an unknown constant ($h_0(\mathbf{x}) = 1$; $J = 0$).

Residual term

- The *bias* or *residual* term is a stochastic process. In particular, it is modeled using a zero-mean, stationary Gaussian stochastic process,

$$\mathcal{Z}(\mathbf{x}) \sim \mathcal{GP}(0, \sigma^2 \mathbf{R}(\mathbf{x}, \mathbf{x}^*)) \quad (4.19)$$

where σ^2 and \mathbf{R} are the process variance and a stationary correlation function (such as the ones presented in Section 4.2.3), respectively. The residuals, being modeled as a GP, are correlated and this correlation is a function of the input parameters. As such, a Kriging model can be thought of as a generalized linear model whose elements of the correlation matrix are specified explicitly by a parametric function [176]. Note that the predictor in Eq. (4.17) becomes a stochastic process due to this term.

Hyper-parameters

According to the above, a GP metamodel contains several parameters called the *hyper-parameters*. This term is used to distinguish them from the input parameters associated with the original simulation model. The hyper-parameters of a GP metamodel are the ones associated with the chosen trend function (Eq. 4.18); the ones associated with the selected correlation functions (Section 4.2.3); and the process variance σ^2 . The total number of hyper-parameters depends on the number of model parameters as well as the selected structure of mean and correlation functions. For instance, for a D-parameter simulation model represented by a GP metamodel with linear first-order mean and power-exponential correlation function (Eq. (4.11)), the total number of the hyper-parameters $\Psi = (\beta, \sigma^2, \theta, \mathbf{p})$ is $3D + 2$; while for the same model represented by a GP metamodel with a constant mean and Gaussian correlation functions (Eq. (4.10)), the total number of hyper-parameters $\Psi = (\mu, \sigma^2, \theta)$ is $D + 2$.

As mentioned earlier, two classes of Kriging models can be distinguished depending on what is specified on the trend term: *Simple Kriging* and *Universal Kriging*. Simple Kriging is the simpler case where all the hyper-parameters involved are known. In that case the prediction of the output at an arbitrary input location is straightforward as shall be seen below.

Simple Kriging

Following the formulation above, a GP metamodel, implies that the computer code outputs at every input locations are jointly Gaussian. As such, the code outputs at the training inputs $\mathbf{DM} = \{\mathbf{x}_i\}_{i=1}^N, \mathcal{Y}(\mathbf{DM}) = (\mathcal{Y}(\mathbf{x}_1), \mathcal{Y}(\mathbf{x}_2), \dots, \mathcal{Y}(\mathbf{x}_N))$ and the output at an arbitrary input $\mathbf{x}_o, \mathcal{Y}(\mathbf{x}_o)$ are distributed jointly as an $(N + 1)$ -dimensional Gaussian,

$$\begin{bmatrix} \mathcal{Y}(\mathbf{DM}) \\ \mathcal{Y}(\mathbf{x}_o) \end{bmatrix} \sim \mathcal{N} \left(\begin{bmatrix} \boldsymbol{\mu}(\mathbf{DM}) \\ \boldsymbol{\mu}(\mathbf{x}_o) \end{bmatrix}, \sigma^2 \begin{bmatrix} \mathbf{R}(\mathbf{DM}, \mathbf{DM}) & \mathbf{R}(\mathbf{DM}, \mathbf{x}_o) \\ \mathbf{R}(\mathbf{x}_o, \mathbf{DM}) & \mathbf{R}(\mathbf{x}_o, \mathbf{x}_o) \end{bmatrix} \right) \quad (4.20)$$

where:

- $\boldsymbol{\mu}(\mathbf{DM})$ is the vector of mean values at the training points,

$$\boldsymbol{\mu}(\mathbf{DM}) = [\mu(\mathbf{x}_1), \dots, \mu(\mathbf{x}_N)]^\top \quad (4.21)$$

- $\mu(\mathbf{x}_o)$ is the mean at an arbitrary test location.
- $\mathbf{R}(\mathbf{DM}, \mathbf{DM})$ is the $N \times N$ correlation matrix between outputs at the training points,

$$\mathbf{R}(\mathbf{DM}, \mathbf{DM}) = \begin{bmatrix} R(\mathbf{x}_1, \mathbf{x}_1) & \cdots & R(\mathbf{x}_1, \mathbf{x}_N) \\ \vdots & \ddots & \vdots \\ R(\mathbf{x}_N, \mathbf{x}_1) & \cdots & R(\mathbf{x}_N, \mathbf{x}_N) \end{bmatrix} \quad (4.22)$$

- $\mathbf{R}(\mathbf{DM}, \mathbf{x}_o) = \mathbf{R}(\mathbf{x}_o, \mathbf{DM})^\top$ is the $N \times 1$ vector of correlation between outputs at the training points and the output at the test point,

$$\mathbf{R}(\mathbf{DM}, \mathbf{x}_o) = \mathbf{R}(\mathbf{x}_o, \mathbf{DM})^\top = [R(\mathbf{x}_o, \mathbf{x}_1), \dots, R(\mathbf{x}_o, \mathbf{x}_N)]^\top \quad (4.23)$$

- $\mathbf{R}(\mathbf{x}_o, \mathbf{x}_o)$ is the correlation of the output at the test input with itself. By definition this correlation is equal to 1.

Provided that the outputs at the training inputs are fully observed (i.e., the code is actually run at those inputs), then the output at the test input $\mathcal{Y}(\mathbf{x}_o)$ given the observed outputs $\mathcal{Y}(\mathbf{DM}) = \mathbf{y} = (y_1, y_2, \dots, y_N)^\top$ is a conditional Gaussian random variable,

$$\mathcal{Y}(\mathbf{x}_o) | \mathcal{Y}(\mathbf{DM}) = \{y_i\}_{i=1}^N \sim \mathcal{N}(m_{SK}(\mathbf{x}_o), s_{SK}^2(\mathbf{x}_o)) \quad (4.24)$$

where m_{SK} and s_{SK}^2 are the mean and the variance of the distribution, respectively. They are also often referred to as the *simple Kriging mean* and *simple Kriging variance*, respectively.

The simple Kriging mean (or the *Kriging predictor*) is expressed as follows

$$m_{SK}(\mathbf{x}_o) = \mu(\mathbf{x}_o) + \mathbf{R}^\top(\mathbf{x}_o, \mathbf{DM})\mathbf{R}^{-1}(\mathbf{DM}, \mathbf{DM})(\mathbf{y} - \mu(\mathbf{DM})) \quad (4.25)$$

Simple Kriging mean

The simple Kriging variance, on the other hand, is expressed as

$$s_{SK}^2(\mathbf{x}_o) = \sigma^2(1 - \mathbf{R}^\top(\mathbf{x}_o, \mathbf{DM})\mathbf{R}^{-1}(\mathbf{DM}, \mathbf{DM})\mathbf{R}(\mathbf{x}_o, \mathbf{DM})) \quad (4.26)$$

Simple Kriging variance

The expressions for the mean and the variance above are obtained through the conditioning operation of the Gaussian random vector in Eq.(4.20) (See Appendix D.3). In practice, the Kriging mean are used as a predictor of the code output at an arbitrary input location, while the variance is used as a measure of error of that prediction.

The simple Kriging model has several interesting features:

- The Kriging predictor given by the mean in Eq.(4.25) is a *linear predictor*. In other words, the centered predictor ($m_{SK}(\mathbf{x}_o) - \mu(\mathbf{x}_o)$) is a weighted linear combination of the centered data ($\mathbf{y} - \mu(\mathbf{DM})$). The weights depends on the correlation function $R(\circ, \circ)$, the design of training points \mathbf{DM} , and the distance between the test point and the training points.

Linear predictor

*Kriging as an
interpolant*

- The variance collapses at the training points, that is plugging-in $\mathbf{x}_i \in \mathbf{DM}$ into Eq.(4.26) will yield $s_{\text{SK}}^2(\mathbf{x}_i) = 0, \forall i$. As such, the Kriging predictor is also an *interpolant*, which exactly fits the observed data (i.e., deterministic code output at the training inputs). See Fig. 4.1C for an illustration.

*Variance as function
of distance between
test and training
points*

- The variance on a given test point does not depend on the observed data. Strictly speaking, it is only dependent on the process variance σ^2 and the correlation function $R(\circ, \circ)$. Furthermore, the variance on a given test point is also equal or less than the process variance, the difference of which depends on the distance between \mathbf{x}_o and the training points \mathbf{DM} . The closer \mathbf{x}_o is to the training points, the smaller the variance at that point. See the difference between two black points in Fig. 4.1C in relation to their relative position to the data.

*Variance as measure
of epistemic
uncertainty*

- Being the variance of a conditional Gaussian distribution, the Kriging variance can be intuitively interpreted as the posterior *uncertainty* of the prediction given the observed data. The nature of this uncertainty is epistemic as, in the case of this thesis, the computer code that underlies the observed data is deterministic. That is, the uncertainty associated with the prediction at an arbitrary input is due to the lack of knowledge because the code itself is not run at that point, though the prediction is informed by the observed data as contained in the training data.

*Ordinary and
Universal Kriging*

As mentioned in Section 4.2.6, adding a mean function in the GP metamodel formulation can provide an opportunity for a more flexible metamodel in the extrapolatory region, where prediction is made at a point far away from the training points. Although there is practically unlimited number of possible mean functions, the function is often represented simply by fixed basis function whose coefficients are unknown (Eq. (4.18)). This leads to the Universal Kriging formulation (Ordinary Kriging for constant mean function), where extra hyperparameters are introduced in the metamodel. Even by restricting the mean function to be within this family, the possibility over the choice of such function is still wide. The questions about the degrees, the interaction terms, etc., are now part of the metamodel construction. All of these eventually result in an even more complex metamodel.

*Simple vs. Ordinary
vs. Universal
Kriging*

The literature, however, is split on the usefulness of adding a mean function in the metamodel formulation. Ref. [176] reported that Kriging with complex trend function gave a better prediction performance for the 2-dimensional non-linear test problem used in the article, while Ref. [166] argued that using mean function of one-degree polynomial allows for a global (i.e., extrapolatory) non-stationary model which did not affect the metamodel performance on the test function. On the other hand, Ref. [177] noted that adding a mean function within the Universal Kriging framework affects the prediction in the extrapola-

tory situation, while any formulation yielded the same performance in the interpolatory situation. Ref. [167] concurred with the conclusion and further warned that in high-dimensional problem with small training samples size, all problems tend to be extrapolatory and a misspecification of the mean function bears the risk of large error in the prediction. The studies in Refs. [160, 178] provided less convincing results of using mean functions and thus suggested the use of either zero or constant mean function for simplicity. And indeed, in this work, the mean function is assumed to be zero by first standardizing the output.

All the Kriging models above assume that the correlation function has been selected and its hyper-parameters are fully known. In most practical situations, there are different choices of correlation functions to choose from. Its hyper-parameters are also not known a priori and have to be estimated from a set of observations. These two problems, *model selection* and *model fitting*, will be discussed in the next section.

*Model selection,
model fitting*

4.4 PRACTICAL ASPECTS OF GP METAMODEL CONSTRUCTIONS

Three basic tasks involved in the construction of a valid metamodel outlined in Section 4.3: selecting the design/training points (i.e., generating \mathbf{DM}), model fitting (i.e., estimating the hyper-parameters Ψ), and model validation (i.e., assessing whether the constructed metamodel is appropriate for its intended use: to replace the expensive simulator code).

4.4.1 Selection of Design/Training Points

The metamodeling of deterministic simulator f to obtain the surrogate \tilde{f} is based on the training data ($\mathbf{DM} = \{\mathbf{x}_n\}_{n=1}^N, \mathbf{y} = \{f(\mathbf{x}_n)\}_{n=1}^N$), the design matrix and the corresponding outputs from the actual simulator runs. The accuracy of \tilde{f} , in turn, is determined by the configuration of \mathbf{DM} , the sample size N , and the true underlying relationship of f [179].

The selection of points in the input parameter space, which determines the geometrical configuration of \mathbf{DM} , is aimed at exploring the whole input parameter space \mathbf{X} , at least in the region where the important features of the model (e.g., region of strong non-linearity) are located. As this region (or regions) is often not known in advance, the most straightforward approach that explore the parameter space is by using the grid approach with a fine discretization shown in Fig. 4.14 [175]. In practice, with a constraint on computational budget, the amount of actual code runs is limited. The objective is then to select the limited points more judiciously to obtain as much information about the model as possible with as few points as possible [54, 58].

Grid approach

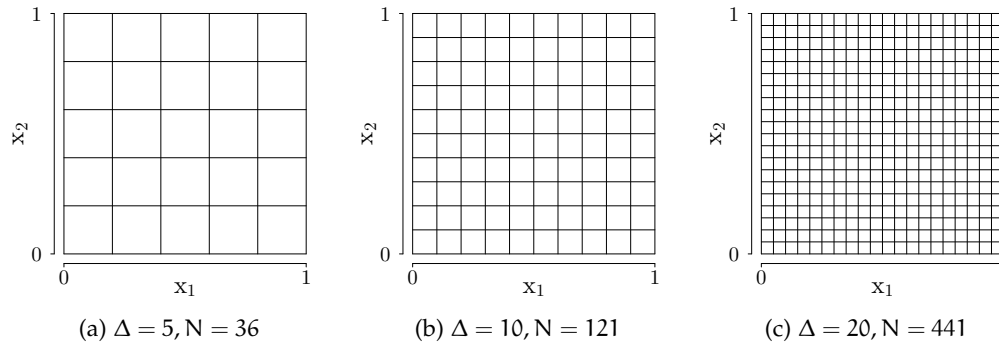


Figure 4.14: Grid approach to select training points becomes prohibitively expensive for high-dimensional problem. Shown here is grid in 2-dimensional input parameter space and the code is supposed to be evaluated at each vertex. In larger-dimension, the problem is worsened with requirement of $N = (\Delta + 1)^D$ code runs, where Δ is the discretization level assumed uniform for all parameters and D is the number of parameters.

Design for computer experiment

Some techniques to select the training points are borrowed from the design of (physical) experiments. Deterministic computer code, however, lacks random error and (hidden) nuisance parameters that renders techniques such as randomization, replication, and blocking irrelevant [35]. On the other hand, computer experiment tends to involve many more input parameters compared to its physical counterpart, which is constrained by cost. A good design for (deterministic) computer experiment, therefore, are constructed based on different set of principles. First, due to the deterministic nature of the underlying code, the design should avoid any repetition of observation. Second, due to the lack of knowledge about the underlying inputs/outputs relationship of the model, the design should spread the available points evenly across input parameter space [35]. In other words, the design should be model-free without assuming any explicit form of inputs/outputs relationship. Third and finally, the design should have a good low dimensional projection properties⁴ [180, 181]. It is further argued in [181] that due to the effect sparsity principle (in relation to parameter interaction), a design with good 2-dimensional projection property is enough to construct an accurate metamodel. Design for computer experiment that roughly follows these principles are generically termed "Space-Filling" [35, 58, 180–182].

*Examples of design:
SRS, LHS, and
Quasi-random
sequence*

Simple random sampling (SRS) (Fig 4.15a) is the simplest and most generic approach to generate design of computer experiment. While technically non-repetitive, the samples generated by SRS are not guaranteed to be well-separated; clusters tends to form around one region of parameter space while leaving other part of the region unexplored.

⁴ Good coverage, no cluster, and does not induce artificial correlation in the projection of the design.

The latin hypercube sampling (LHS) initially developed for the analysis of computer experiment in lieu of SRS [183] has become a popular alternative in computer experiment [184]. LHS guarantees that values for each input dimension is different (Fig. 4.15b) (i.e., has an excellent 1-dimensional projection). The projection in higher dimension, however, is still not guaranteed to be optimal. Its improvement to provide a better uniformity properties in all dimension have been continuously proposed in the literature [35, 54, 181, 182, 184]. More recently, the use of quasi-random sequence originally applied to accelerate the convergence of Monte Carlo integration (see for instance Ref. [185]) has also been applied for constructing experimental design. Fig. 4.15c is an example of such design, generated using Sobol' quasi-random sequence.

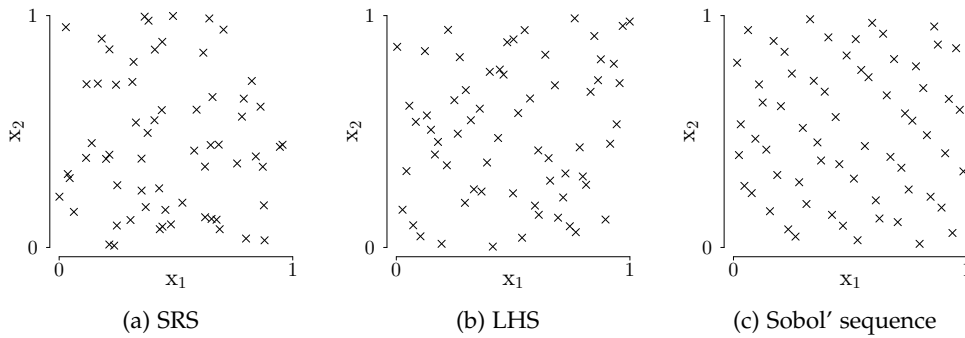


Figure 4.15: Examples of experimental design for metamodel training in 2-dimensional input parameter space. Any 2-dimensional projection from higher dimension is represented in the same manner.

It is also worth noting that the literature has no consensus regarding the extend to which the design of experiment is important for metamodel accuracy [184]. Several authors (such as in Refs. [175, 180, 181]) emphasized the design utmost importance while others (such as in Refs. [58, 178, 186]) considered it to be less important, especially compared to the training sample size. Those three latter studies reported that while a better design might be important for a relatively small sample, the importance of sample size will eventually eclipse the importance of a more efficient design (especially when such a convergence study can be afforded). That is, the accuracy of the resulting metamodel converges to the same value with increasing sample size regardless of the design. On the other hand, the size of training sample at which the metamodel accuracy becomes acceptable, is different from application to application and, as noted in Ref. [187], is closely related to the complexity of the underlying function. The paper proposes the sample size of $N = 10 \times D$ as a rule of thumb for starting point. As the complexity of the underlying function is not known in advance, an empirical study for each case has to be carried out to assess whether the resulting metamodel is acceptable.

On the importance of sample size

*One-shot vs.
sequential design*

As a final remark on the subject of design, all the designs considered in this thesis belong to a strategy called one-stage or one-shot strategy [188, 189]. The strategy means that the training samples are generated at once and a metamodel is constructed and applied only based on that. Generating training samples of larger size might be necessary, but the larger samples will be generated essentially from scratch without using the results obtained from the smaller samples. Sequential design is the alternative approach where the new design points are added sequentially to the initial batch of training set. In essence, it adaptively samples the input parameter space around the more interesting region (with more variation thus more difficult to approximate) based on the previously constructed metamodel. The newly found point is then augmented and a new metamodel is constructed and the process is repeated until the required level of accuracy is attained. Though it potentially leads to a more efficient design (fewer samples required overall), it also adds additional complexity to metamodel construction (see for example Refs. [189, 190]).

4.4.2 Model Fitting/Training

In most metamodeling applications, the values of the hyper-parameters for a selected GP metamodel are not known a priori. The parameter estimation process, a term interchangeably used with *fitting*, *training*, and *learning*, applies mathematical techniques to a set of training data to estimate the values of the hyper-parameters [68]. In the following it is assumed that a particular correlation function has been selected and that the mean function $\mu(\circ)$ is known, with values at training points denoted in the following simply as $\boldsymbol{\mu}$. In other words, it starts from the Simple Kriging formulation.

Likelihood function

To estimate the values of the hyper-parameters $\boldsymbol{\Psi}$ of a chosen structure of mean and covariance functions, it should be first acknowledged that under GP model, the distribution of the observed data given a Gaussian process $(\mathbf{y} | \mathcal{Y}(\mathbf{x}); \boldsymbol{\Psi})$ is Gaussian, such that its PDF is of the form

$$\mathcal{L}(\boldsymbol{\Psi}; \mathbf{y}) = \frac{1}{(2\pi)^{N/2} (\sigma)^{N/2} |\mathbf{R}|^{1/2}} \exp \left[-\frac{(\mathbf{y} - \boldsymbol{\mu})^T \mathbf{R}^{-1} (\mathbf{y} - \boldsymbol{\mu})}{2\sigma^2} \right] \quad (4.27)$$

The term above is called the *likelihood* function. The slight change of perspective from a conditional density function to a common function is due to the fact that the data is already observed [191]. For compactness, the $N \times N$ correlation matrix between outputs at the training points $\mathbf{R}(\mathbf{DM}, \mathbf{DM})$ is written simply as \mathbf{R} ; and the N -dimensional vector of the mean value at the training points as $\boldsymbol{\mu}$. Finally, it is also implied in the formulation that the chosen GP is fully specified, through its hyper-parametrization $\boldsymbol{\Psi}$ such that the notation $\mathcal{Y}(\mathbf{x})$ is removed from the expression. The hyper-parameters $\boldsymbol{\Psi}$, in turn, include

σ^2 and other hyper-parameters related to the correlation kernel function \mathbf{R} ⁵.

Starting from the likelihood formulation, a common approach to estimate the hyper-parameters values is by selecting the ones that maximize the likelihood for a given observed data \mathbf{y} . This estimation procedure, the maximum likelihood estimation, is also known in the literature as *empirical Bayes* [175] where the estimation is derived strictly from available data. The procedure is as follows: First, the hyper-parameters related to \mathbf{R} , noted Θ , are initially assumed to be known to estimate σ^2 by minimizing the negative log likelihood⁶ (which is equivalent to maximizing the likelihood),

Maximum likelihood estimation / empirical Bayes

$$(\hat{\sigma}^2 | \Theta) = \arg \min_{\sigma^2} \left(-\ln \mathcal{L}(\hat{\sigma}^2; \hat{\Theta}) \right) \quad (4.28)$$

yielding

$$\hat{\sigma}^2 = \frac{(\mathbf{y} - \boldsymbol{\mu})^\top \mathbf{R}_{\Theta}^{-1} (\mathbf{y} - \boldsymbol{\mu})}{N} \quad (4.29)$$

The estimated $\hat{\sigma}^2$ are then fed back into Eq. (4.27) to obtain the so-called *concentrated/profile likelihood* [192, 193]. The term is due to the fact that the full likelihood has been further conditioned by setting some of the parameters (in this case σ^2) to a constant (in this case, its maximum likelihood estimates). This procedure eases the numerical difficulty of finding simultaneously the maximum likelihood estimates of all the hyper-parameters in high-dimensional space. Finally, the estimate of $\hat{\Theta}$ is obtained through the maximum of the (profile) likelihood,

Concentrated (profile) likelihood

$$\left(\hat{\Theta} | \hat{\sigma}^2 \right) = \arg \min_{\Theta} \left(-\ln \mathcal{L}(\hat{\Theta}; \hat{\sigma}^2) \right) \quad (4.30)$$

The computation of Eq. (4.30) can then be carried out using an unconstrained optimization algorithm, such as the Newton's, quasi-Newton, or one of the global stochastic (e.g., genetic algorithm) methods. Review of different types of optimization algorithms can be found in Ref. [194].

Having estimated the hyper-parameters, the Kriging predictor is expressed as,

$$\hat{m}_{\text{SK}}(\mathbf{x}_o) = \boldsymbol{\mu}(\mathbf{x}_o) + \mathbf{r}_{o, \hat{\Theta}}^\top \mathbf{R}_{\hat{\Theta}}^{-1} (\mathbf{y} - \boldsymbol{\mu}) \quad (4.31)$$

As before, for compactness, the $N \times 1$ correlation vector between outputs at the test and the training points $\mathbf{R}(\mathbf{x}_o, \mathbf{DM})$ is written simply as

⁵ e.g., for Gaussian kernel there is one hyper-parameter θ for each input while for power-exponential kernel there are two hyper-parameters, p and θ , for each input.

⁶ Logarithm is often taken on the likelihood to avoid underflow error when dealing with a very small number.

\mathbf{r}_o . The subscript $\hat{\Theta}$ appears in \mathbf{r} and \mathbf{R} which implies that the correlation functions are evaluated using the maximum likelihood estimated values of the hyper-parameters.

The variance associated with the predictor is expressed as

$$\hat{\sigma}_{\text{SK}}^2(\mathbf{x}_o) = \hat{\sigma}^2(1 - \mathbf{r}_{o,\hat{\Theta}}^T \mathbf{R}_{\hat{\Theta}}^{-1} \mathbf{r}_{o,\hat{\Theta}}) \quad (4.32)$$

Uncertainty on Ψ

Eqs. (4.31-4.32) are the same as Eqs. (4.25-4.26), except now the hyper-parameters are replaced by their Maximum Likelihood (ML) estimates. This implies that the uncertainties associated with the ML estimates are not incorporated into the Kriging predictor and variance [195]. That is, the uncertainties of the predictor (its variance) given the observed data is also conditional on a particular values of hyper-parameters which underestimate the true Kriging variance [196].

Full Bayesian treatment

Full Bayesian treatment of this problem acknowledges this additional source of uncertainty and considers the hyper-parameters as nuisance parameters. It assumes a prior over the hyper-parameters, compute the posterior based on the training data, and then use the posterior to average (integrate) the hyper-parameters out from the Kriging predictor and variance [83, 88, 93, 195]. This increases the computational cost as well as the complexity of the analysis with mixed results [182]. As noted in Bayarri [93], whose ultimate goal was model calibration against experimental data, the answers provided by either analysis (ML estimates and full Bayesian) are equivalent as the effect of model parameters uncertainties tends to dominate the effect of hyper-parameters uncertainties.

Metamodel fitting estimates the optimal hyper-parameters values relative only to the training data. Its robustness, which depends on the training data, the estimation technique, and the underlying complexity of the simulator, is subjected to the validation process presented in the next section.

4.4.3 Model Validation and Selection

Metamodel validation

Metamodel is always fitted based on a relatively small training data, much smaller than the space of all possible inputs/outputs. As such, its validation is a necessary step in applying the metamodel with confidence at any given input as a surrogate of the original computer simulator. Metamodel validation is defined as a process to determine whether a metamodel has a sufficient range of accuracy within its domain of applicability, consistent with its intended use [68]. As the metamodel is based on a deterministic simulator, the output of running the simulator at an input not used in the fitting process provides the ground truth for assessing the metamodel performance. Different validation metrics can be defined based on this comparison that highlight different inadequacies in line with the intended use. Because ex-

haustively comparing the metamodel prediction and the actual simulator output for all possible inputs is not feasible, a validation strategy is devised, dealing with the approach in generating validation data and in using them to assess the metamodel [197].

The gold standard of validation strategy is by an *independent validation (holdout)* data [178]. In this strategy, a separate validation dataset (samples) is created by generating randomly a new set of validation inputs at which the simulator is evaluated. The metamodel assessment is then made by comparing the prediction made by the metamodel and the output produced by the actual simulator runs. The strategy is straightforward, but because the simulator has to be run at the new validation inputs, the cost of generating the validation dataset is high for an expensive simulator. In addition to that, the results can also be sensitive to the size of validation samples [198].

*Validation (holdout)
samples strategy*

For a computationally expensive simulator, it is not always possible to generate large (if any at all) independent validation samples. The *cross-validation* is an alternative approach to validate a metamodel in this situation [197, 199]. In cross-validation, a batch of samples is removed from the available training samples, used the remaining training samples for fitting and the subsamples for assessing the metamodel (essentially becomes the validation samples). The procedure is repeated by selecting randomly the elements for the removed batch. The most extreme case of this approach is the so-called *leave-one-out* (LOO) cross-validation, where a single training point is removed for validation purpose and exhaustively repeating the procedure. Cross-validation does not require additional simulator runs to generate validation samples. It also incorporates in its results, to a certain extent, the sensitivity due to perturbation in the training samples. However, it can potentially be expensive if numerous metamodel fitting are to be carried out (such as in the case of the LOO approach). Furthermore, if an experimental design with a particular geometrical structure is used, removing one or more points might destroy its property⁷. The fitting, in turn, is carried out in sub-optimal manner [199, 200] and the prediction becomes rather pessimistic (i.e., with larger error).

*Cross-validation
strategy*

The second part of the strategy is to define a validation metric in line with the intended use of the metamodel. Because the metamodel in this thesis is going to be used to explore the posterior probability across the model parameter space, the aim is to construct a metamodel that has a decent global accuracy⁸. Global accuracy measures the metamodel performance over many different input values across its parameter space, on average. The accuracy of the metamodel for a particular input, however, can still be poor and in the case of GP metamodel is defined probabilistically.

Validation metric

⁷ This is indeed the case for an optimized latin hypercube design.

⁸ as opposed to the (local) accuracy in a particular region of input parameter space such as during design optimization.

Predictivity
coefficient Q_2

A particular validation metric that quantifies such global accuracy is the *predictivity coefficient* Q_2 [199]. Assuming that the independent validation samples strategy is adopted, let $\mathbf{DM}^* = \{\mathbf{x}_n^*\}_{n=1}^{N_{\text{valid}}}$ denote the set of validation inputs at which the computer simulator is evaluated, yielding N_{valid} outputs $\mathbf{y}^* = \{f(\mathbf{x}_n^*) = y_n^*\}_{n=1}^{N_{\text{valid}}}$. The predictivity coefficient of a metamodel $\hat{y}(\circ)$ (trained using different set of sample) is given by,

$$Q_2(\hat{\mathbf{y}}^*, \mathbf{y}^*) = 1 - \frac{\sum_{n=1}^{N_{\text{valid}}} (y_n^* - \hat{y}_n^*)^2}{\sum_{n=1}^{N_{\text{valid}}} (y_n^* - \bar{y}^*)^2} \quad (4.33)$$

where \mathbf{y}^* is the outputs at validation inputs produced by the simulator; $\hat{\mathbf{y}}^*$ is the predictions at validation inputs made by the metamodel, i.e., $\{\hat{y}(\mathbf{x}_n^*) = \hat{y}_n^*\}_{n=1}^{N_{\text{valid}}}$; and \bar{y}^* is the sample mean of the simulator output in the validation samples. The predictivity coefficient can be interpreted as the proportion of the output variance explained by the metamodel relative to the variance of the validation samples. Q_2 with values close to 1.0 implies a highly accurate metamodel.

On model selection
and metamodeling
choices

Finally, the validation procedure above assumed the design of experiment for training, the correlation function, and the form of mean function have been chosen. These are additional metamodeling choices an analyst has to made upfront and there is no general rule to select which one for which application. Different studies reported one proper metamodeling choice that is supported by a particular case (or some cases) and presented them as generic advice. The danger of taking such advices at face value, as noted in Ref. [178], is that the results might be anecdotal. More importantly, they might not apply to the particular case being studied. The most pragmatic approach in assessing the appropriateness of such choices is thus to carry out empirical study for the particular case being studied. In other words, different design of experiments (ideally with replications), different correlation functions, and different mean functions (if applied) are to be tested for the same problem and the best choices are selected based on the comparison of validation metrics.

Metamodel promises much faster evaluation of the simulator output at any given input, but for it to be used with confidence, some time has to be invested to properly design, fit, and validate it.

4.5 DEALING WITH MULTIVARIATE OUTPUT

Multivariate
outputs

The previous discussion on GP metamodel dealt with a single output (univariate) case. Many computer simulations produce multivariate outputs⁹. A typical TRACE simulation, for example, produces flow variables as functions of time and space as its *raw* outputs. This is indeed the case for the reflood simulation problem presented in

⁹ In this thesis, the number of outputs are referred to as the *dimension of the output parameter space*.

Chapter 2. As outlined in Chapter 3, some techniques can be used to transform the raw outputs into quantities of interest (the maximum, etc.) that are useful to answer the questions at hand. However, in the calibration setting, some of these outputs have corresponding measurement data and need to be represented by the metamodel in their original form for a direct comparison.

An approach proposed in Ref. [68] is to represent the multiple outputs by separate metamodels. That is, one metamodel is developed to represent each one of the multiple outputs individually. Yet, for a very high-dimensional output (from tens to thousands), this approach is impractical as the numbers of metamodel to train becomes too numerous. Furthermore, the outputs produced by the computer simulation are often highly correlated to each other. As such, developing individual metamodels to represent the correlated outputs separately, especially when they are numerous, is wasteful.

To cope with the problem of high-dimensionality of the outputs, this thesis adopted a linear model of coregionalization (LMC) [201, 202] coupled with a principal component analysis (PCA) [88, 203] to construct a tractable, multivariate version of GP metamodel. The original LMC was formulated to model multivariate data in geostatistics that covary together (over a region) in a linear fashion, while PCA is used here as a data-driven dimensional reduction tool. The resulting model consists of few *independent, univariate* GP metamodels, each of which is the one presented in the previous section.

Separate univariate metamodel

Extension to multivariate case

4.5.1 Linear Model of Coregionalization (LMC)

The function that represents the computer code simulation f is now cast in its multivariate version, $\mathbf{f} : \mathbf{X} \subseteq \mathbb{R}^D \mapsto \mathbb{R}^P$ where P is the dimension of the output parameter space. The LMC of the P -dimensional GP metamodel \mathcal{Y} can be written as,

$$\mathcal{Y}(\mathbf{x}) = \boldsymbol{\mu}(\mathbf{x}) + \boldsymbol{\Phi}\mathbf{w}(\mathbf{x}) + \boldsymbol{\epsilon} \quad (4.34)$$

where $\boldsymbol{\mu}$ is the P -dimensional mean vector of the multivariate process; $\boldsymbol{\Phi}$ is a $P \times Q$ matrix, with $Q \leq P$; $\boldsymbol{\epsilon}$ is a P -dimensional vector of linearization error; and $\mathbf{w}(\mathbf{x}) = (w_i(\mathbf{x}))$ is a Q -dimensional vector with univariate GPs as its elements,

$$w_i(\mathbf{x}) \sim \mathcal{GP}(0, \sigma_i^2 R_i(\mathbf{x}, \mathbf{x}^*)) \quad (4.35)$$

where σ_i^2 and R_i are the process variance and correlation function associated with each element of the vector, respectively. The term $\boldsymbol{\Phi}\mathbf{w}(\mathbf{x})$ describes the covariation between the multivariate outputs as function of model parameters.

Linear model of coregionalization

4.5.2 *Principal Component Analysis*

PCA is then used as a data-driven approach to obtain the components of the LMC in Eq. (4.34). The term data-driven is used as the components are derived directly from the training data. The raw outputs of the training runs are first concatenated row-wise resulting in an $N \times P$ matrix $\mathbf{Y}(\mathbf{DM})$,

$$\mathbf{Y}(\mathbf{DM}) = \begin{pmatrix} & \vdots & \\ \text{---} & \mathbf{y}_n & \text{---} \\ & \vdots & \end{pmatrix} \quad (4.36)$$

$$\begin{aligned} \mathbf{y}_n &= [y_{n,1}, \dots, y_{n,p}, \dots, y_{n,p}] \\ &= [y(\mathbf{x}_n)_1, \dots, y(\mathbf{x}_n)_p, \dots, y(\mathbf{x}_n)_p] \end{aligned}$$

where $y(\mathbf{x}_n)_p$ is the p -th output dimension, evaluated using the n -th training sample. Note that the notation above is similar to Eq. (3.2) but now the dimension of the output y_p is not only restricted to time, nor do they have to be of the same (physical) dimension. In the formulation below, the raw training outputs is always assumed to be dependent on the training samples and thus the notation \mathbf{DM} is suppressed.

The sample mean of the raw outputs is used to substitute the mean in the LMC formulation,

$$\boldsymbol{\mu}(\mathbf{x}) = \bar{\mathbf{y}}^\top \quad (4.37)$$

The sample mean is obtained by taking the column-wise average of Eq. (4.36),

$$\begin{aligned} \bar{\mathbf{y}} &= [\bar{y}_1, \dots, \bar{y}_p, \dots, \bar{y}_p] \\ \bar{y}_p &= \frac{1}{N} \sum_{n=1}^N y(\mathbf{x}_n)_p \end{aligned} \quad (4.38)$$

Note that by the above, the mean of the LMC is a constant vector.

As the PCA deals with the data covariance matrix, the raw outputs in Eq. (4.36) should first be centered,

$$\mathbf{Y}^* = (\mathbf{Y} - \mathbf{j}_N \bar{\mathbf{y}}) \quad (4.39)$$

where \mathbf{j}_N is the N -dimensional vector of ones.

The centered raw outputs \mathbf{Y}^* is then decomposed by means of singular value decomposition (SVD) yielding,

$$\mathbf{Y}^* = \mathbf{U} \mathbf{S} \mathbf{V}^\top \quad (4.40)$$

where \mathbf{U} is the $N \times N$ orthogonal, left singular matrix; \mathbf{S} is the $N \times P$ diagonal matrix of singular values; and \mathbf{V} is the $P \times P$ orthogonal, right singular matrix.

The column vector elements of \mathbf{V} are the principal components (PCs) of the data set which describe the main directions of the data set, along which the variance of the data set is the largest. PCs are sorted in descending order such that the first PC (leftmost in \mathbf{V}) contains the largest variance (Fig. 4.16a). The singular values are related to the explained variance of the eigenvectors (i.e., their respective eigenvalues) by the following,

Principal component

$$\lambda = \text{diag} \left(\frac{\mathbf{S}^2}{N-1} \right) \tag{4.41}$$

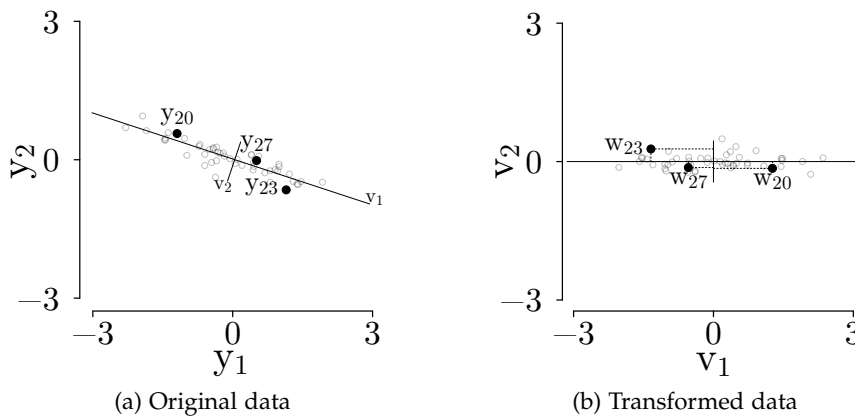


Figure 4.16: PCA of a bivariate data set. A highly correlated bivariate data set can be transformed into a new orthogonal coordinate system according to the principal directions of the data set. The principal directions redistribute the partial variance such that the total variance is preserved in the transformed coordinate. Above, three selected points in the data set in both coordinates.

Projection of the data into the PCs results in *principal component scores* (PC scores),

Principal component score

$$\mathbf{W} = \mathbf{Y}^* \mathbf{V} = \mathbf{U} \mathbf{S} \tag{4.42}$$

where \mathbf{W} is the $N \times P$ matrix of principal component scores. A unique set of P principal component scores are associated with each points in the multivariate data set. The scores describe the locations of the multivariate data points in the new coordinate system as defined by the principal components (Fig. 4.16b).

Often the results of PCA are reformulated in terms of principal component loadings and standardized scores,

$$\mathbf{V}^* = \frac{1}{\sqrt{N-1}} \mathbf{V} \mathbf{S} \tag{4.43}$$

$$\mathbf{W}^* = \sqrt{N-1} \mathbf{U} \tag{4.44}$$

where $\mathbf{Y}^* = \mathbf{W}^* \mathbf{V}^{*\top}$ holds¹⁰. This reformulation gives a magnitude to the unit-norm principal components according to their respective standard deviation (square-root of variance Eq. (4.41)), it also results in standardized PC scores with variance of 1.0. In the context of regression of PC scores used later in metamodeling, the latter property improves the numerical stability of the maximum likelihood estimation of the hyper-parameters.

*Dimension
Reduction*

Dimension reduction takes place when only a small numbers of PCs are kept. That is, only the first Q columns of \mathbf{V}^* are retained with $Q \ll P$. Such selection is justified by a certain amount of partial variance explained by those few first Q principal components. In the illustration of Fig. 4.16, the dimension reduction can be carried out by simply using the first principal component (horizontal axis of Fig. 4.16b) to describe the data set.

Back to the formulation of LMC, the matrix Φ in Eq. (4.34) is substituted by the set of empirical orthogonal basis functions obtained from the first Q PC loadings of the data set, and is defined as

$$\Phi = (\mathbf{v}_1^*, \dots, \mathbf{v}_q^*, \dots, \mathbf{v}_Q^*) \quad (4.45)$$

where \mathbf{v}_q^* is the P -dimensional column-vector taken from the q -th column of matrix \mathbf{V}^* ; and $Q \ll P$. This empirical orthogonal basis functions expansion is obviously related to the ones presented in Chapter 3. The analysis done here, however, used the point-wise data formulation as opposed to functional formulation.

4.5.3 Multivariate Gaussian Process Metamodel

The multivariate output formulation of a GP metamodel based on the previous discussion is summarized as the following equation, where a prediction at an arbitrary input $\mathbf{x}_o \in \mathbf{X}$ is made,

$$\mathcal{Y}(\mathbf{x}_o) = \bar{\mathbf{y}} + \Phi_Q^* \mathbf{w}^*(\mathbf{x}_o) + \Phi_{>Q}^* \mathbf{e} \quad (4.46)$$

where $\bar{\mathbf{y}}$ is the P -dimensional vector of sample mean (Eq. (4.38)). The other elements in the equation are described below.

Φ_Q^* , a $P \times Q$ matrix, is the first Q columns of the PC loadings retained to reconstruct the multivariate output. Specifically, Φ_Q^* is,

$$\Phi_Q^* = (\Phi_1^*, \Phi_2^*, \dots, \Phi_Q^*) = (\mathbf{v}_1^*, \mathbf{v}_2^*, \dots, \mathbf{v}_Q^*); Q \ll P \quad (4.47)$$

where \mathbf{v}_i^* is the P -dimensional column vector of the i -th PC loading taken from Eq. (4.45).

\mathbf{w}^* is the Q -dimensional vector of standardized PC scores for each of the retained PC loading, modeled as a set of univariate, independent, zero-mean GPs,

$$\begin{aligned} \mathbf{w}^* &= [w_1^*, w_2^*, \dots, w_Q^*]; Q < P \\ w_i^*(\circ) &\sim \mathcal{GP}(0, \sigma_i^2 \mathbf{R}_i(\circ, \circ)) \end{aligned} \quad (4.48)$$

¹⁰ $\mathbf{W}^* \neq \mathbf{Y}^* \mathbf{V}^*$ as \mathbf{V}^* is not orthogonal any longer, but $\mathbf{W}^* = (\mathbf{N} - 1) \mathbf{Y}^* \mathbf{V}^{*\top} \mathbf{S}^{-2}$.

where w_i^* is the standardized PC scores of the i -th PC loading; σ_i^2 and $R_i(\circ, \circ)$ are the process variance and the correlation function associated with the GP of w_i^* , respectively.

The observed data from the N training samples is related to each of the w_i by,

$$\mathbf{w}_i^*(\mathbf{DM}) \equiv \mathbf{w}_i^* = \sqrt{N-1} \mathbf{Y}^* \mathbf{S}^{-1} \mathbf{v}_i^*; i = 1, \dots, Q \quad (4.49)$$

That is, the observed data for the i -th standardized PC score is the projection of the standardized data on the i -th PC loading. Conditioning the GP of w_i by the observed data yields,

$$\mathbf{w}_i^*(\mathbf{x}_o) | \mathbf{w}_i^* \sim \mathcal{N}(m_{SK,i}(\mathbf{x}_o), s_{SK,i}^2(\mathbf{x}_o)); i = 1, \dots, Q \quad (4.50)$$

where $m_{SK,i}$ and $s_{SK,i}^2$ are the simple Kriging mean and variance, respectively (Eq. (4.25) and Eq. (4.26)), associated with the i -th standardized PC score. The simple Kriging formulation is used here as the assumed process is already centered (zero-mean).

$\Phi_{>Q}^*$, a $P \times (P - Q)$ matrix, is the unretained columns of the PC loadings,

$$\Phi_{>Q}^* = (\Phi_{Q+1}^*, \Phi_{Q+2}^*, \dots, \Phi_P^*) = (\mathbf{v}_{Q+1}^*, \mathbf{v}_{Q+2}^*, \dots, \mathbf{v}_P^*) \quad (4.51)$$

where Φ_{Q+i}^* is the P -dimensional column vector of the unretained i -th PC loading taken from Eq.(4.43).

Finally, following Ref. [90], \mathbf{e} is the $(P - Q)$ -dimensional vector of independent identically distributed normal random variable with mean 0 and variance 1. In other words, truncation error in Eq. (4.46) due to the unretained PC loadings are modeled as a set of independent normal random variables with the variance given by the PC loadings.

The multivariate output evaluated at \mathbf{x}_o conditioned by the training data is thus distributed as P -variate Gaussian random variable,

$$\begin{aligned} \mathbf{y}(\mathbf{x}_o) | \mathbf{y}(\mathbf{DM}) = \mathbf{Y} &\sim \mathcal{N}_P(\boldsymbol{\mu}_P(\mathbf{x}_o), \Sigma_{P \times P}(\mathbf{x}_o)) \\ \boldsymbol{\mu}_P &= \hat{\mathbf{y}} + \Phi_Q^* \mathbf{m}_{SK}(\mathbf{x}_o) \\ \Sigma_{P \times P} &= \Phi_Q^* \text{diag}(s_{SK}^2(\mathbf{x}_o)) \Phi_Q^{*T} + \Phi_{>Q}^* \mathbf{I} \Phi_{>Q}^{*T} \\ \mathbf{m}_{SK} &= [m_{SK,1}(\mathbf{x}_o), m_{SK,2}(\mathbf{x}_o), \dots, m_{SK,Q}(\mathbf{x}_o)] \\ s_{SK}^2 &= [s_{SK,1}^2(\mathbf{x}_o), s_{SK,2}^2(\mathbf{x}_o), \dots, s_{SK,Q}^2(\mathbf{x}_o)] \end{aligned} \quad (4.52)$$

The model above assumed that the hyper-parameters associated with a selected correlation function are known. In practice, they are not and thus estimated from the data itself using Maximum Likelihood Estimation (MLE) as outlined in Section 4.4.2. Additionally, the theoretical underestimation of the Kriging variance explained in that section also applies here when the ML estimates are plugged into the formulation. In practice, however, Ref. [93] noted that the uncertainties associated with the model parameters often eclipse the uncertainties induced by the metamodel hyper-parameters.

4.6 APPLICATION TO THE TRACE MODEL OF FEBA

In this section, a GP metamodel of the TRACE model of the FEBA facility is constructed and assessed. As before, only the results from analyzing the TRACE model of the FEBA test No. 216 are presented. Following the results of SA from Chapter 3, only the 12 most influential parameters are being considered in the following. The resulting GP metamodel of the TRACE model will then be used for the parameter calibration problem tackled in the next chapter.

4.6.1 Simulation Experiment

The construction of GP PC metamodel of the FEBA TRACE model was carried out in three steps following the recommendation in the statistical/machine learning literature [62, 204]: training, validation, testing as summarized in Fig. 4.17 below.

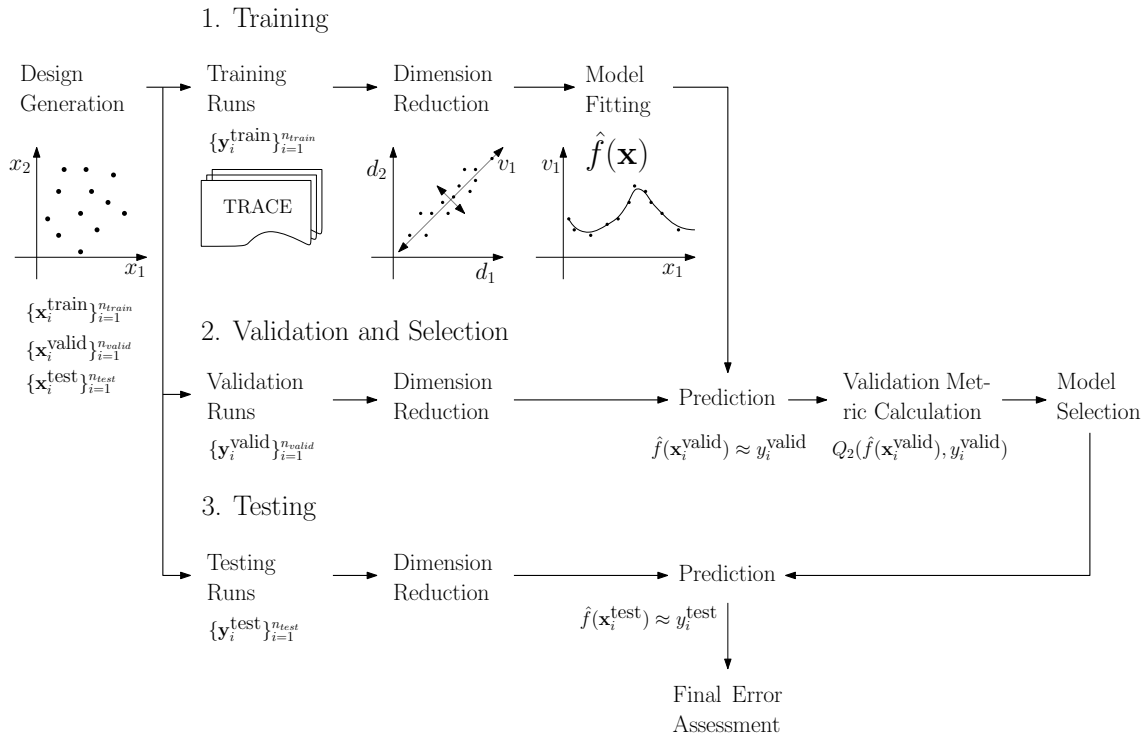


Figure 4.17: Flowchart of the simulation experiment for constructing a GP PC metamodel of the TRACE model of the FEBA facility

In the training step, a metamodel was constructed (i.e. *trained*) based on a training data set. The training data set consists of a set of training inputs and its code output counterparts (from actual code runs). The set of training inputs was generated using an experimental design

algorithm (of which several of them were considered: simple random sampling, latin hypercube, optimized latin hypercube, and Sobol' sequence). Following the recommendation in Ref. [187] the starting sample size was 120 (or $10 \times D$, with D the number of dimensions) and increased in multiples of two (240, 480, and 960). Furthermore, to take into account the effect of random variation in the training data set of each design and size, five replications of each data set were generated for training and five different metamodels were trained.

Metamodels predicting different types of outputs (i.e., clad temperature, pressure drop, and liquid carryover) were constructed separately. These different outputs were themselves of a multivariate nature: liquid carryover was a time-dependent quantity, while clad temperature and pressure drop were time- and space-dependent quantities. As such, PCA was carried out on the raw outputs to reduce their dimensionality and the metamodel was trained with respect to a few retained PC scores.

*Dimension
reduction*

Carrying out PCA for each type of outputs results in pairs of standardized PC scores and PC loadings (see Section 4.5). A GP metamodel was then trained with respect to the standardized PC scores for a selected number of retained PC loadings. Therefore, there were multiple GP metamodels representing the standardized PC scores associated with each of the retained PC loadings for each of the output types.

Several covariance kernel functions were considered for constructing a metamodel: Gaussian, Matérn 3/2, Matérn 5/2, and power-exponential kernels. The hyper-parameters associated with each kernel were estimated (i.e., *fitted*) by MLE as implemented in the R package *DiceKriging* [170]. Following the hyper-parameters estimation for each GP PC metamodel, a metamodel of the TRACE model was fully trained and ready for making prediction in arbitrary inputs. To make prediction back in the original physical space, the full output space had to be first reconstructed using linear combinations of the predicted standardized PC scores and the PC loadings (See Eq. (4.52)).

Model fitting

A validation step was conducted to assess and compare the predictive performance of different metamodels constructed with different experimental designs and covariance functions, taking into account the effects of randomness in the experimental design generation and of training sample size. The validation step was conducted by means of independent validation data sets, a separate set of TRACE outputs from actual runs (preferably large enough) for the metamodels to predict. The predictivity coefficient defined in Eq. (4.33) measured the discrepancy between the output from the validation data set and from the prediction by the metamodel, and thus the quality of the metamodel. However, to have a more intuitive measure of performance directly related to the output in the physical space (as

*Model validation
and selection*

opposed to the reduced space of the principal components), the root-mean-square-error (RMSE) of the reconstructed prediction was also used. It is defined below,

$$\text{RMSE}_{\text{rec.}} = \left[\frac{1}{N_{\text{valid}}P} \sum_{n=1}^{N_{\text{valid}}} \sum_{p=1}^P (y_{n,p} - \hat{y}_{n,p})^2 \right]^{0.5} \quad (4.53)$$

$$\hat{\mathbf{y}}_n = \bar{\mathbf{y}} + \Phi_Q^* \mathbf{m}_{\text{SK}}(\mathbf{x}_n) = [\hat{y}_{n,1}, \dots, \hat{y}_{n,p}, \dots, \hat{y}_{n,p}]$$

where N_{valid} is the number of validation data; P is the number of dimension of the output space; $y_{n,p}$ is the value of the output dimension p at validation input n ; $\hat{y}_{n,p}$ is the predicted value of the output dimension p at validation input n ; and $\hat{\mathbf{y}}_n$ is the mean of the reconstructed multivariate output at validation input n predicted by the GP PC metamodel (see the explanation of Eq. (4.52) for detail). This error combined the error due to the misprediction of the standardized PC scores by the metamodel as well as the error due to the truncation of the PCs. The best setting of the metamodel (the experimental design and the covariance function) was then selected based on the RMSE and one additional metamodel was trained using an increased number of training samples.

Testing Finally, the ultimate performance of the metamodel were assessed in the testing step based on yet another large number of test data set, separately generated. The purpose of this step was to further confirm the previous results on another data set and to give a more robust idea of the expected error of the metamodel in the application setting.

The settings used in the simulation experiment for constructing and assessing the GP PC metamodel are summarized in Table 4.1.

4.6.2 Dimension Reduction by principal component analysis (PCA)

As mentioned, the different outputs considered in this study (i.e., clad temperature, pressure drop, and liquid carryover) were of multivariate nature. For instance, the clad temperature output was defined both in time instance and axial location. Considering eight different axial locations for the thermocouples and 1'000 [s] transient (to ensure that all runs were quenched) with time step size of 0.1 [s], the dimensionality of this output amounted to 80'000. Fig. 4.18 shows three different clad temperature outputs from three different TRACE runs. It shows the contour plot of clad temperature as function of time and axial location in the x - and y - axes, respectively. Though not shown here, the other two types of output were of similar nature: pressure drop was also defined in time instance and in four axial segments (with dimensionality of the output amounting to 40'000), while liquid carryover output was defined only in time (with dimensionality amounting to 10'000).

Table 4.1: Simulation experiment settings for constructing and assessing the GP PC metamodel of the TRACE model of FEBA test No. 216

DESCRIPTION	VALUE
Inputs/Outputs	
Number of inputs	12
Number of outputs:	
Clad temperature	8 (axial locations) \times 10'000 (time-steps) = 80'000
Pressure drop	4 (axial segments) \times 10'000 (time-steps) = 40'000
Liquid carryover	10'000 (time-steps)
Dimension reduction	principal component analysis (PCA)
Training	
Experimental Designs	Simple random (SRS), latin hypercube (LHS), optimized latin hypercube (Opt. LHS), Sobol' sequence
Sample sizes	120, 240, 480, 960, 1'920 (only for testing)
Replication	5
Covariance kernels	Gaussian, Matérn 3/2, Matérn 5/2, power-exponential
Model fitting	Maximum Likelihood Estimation (MLE)
Validation	
Strategy	Independent data set (holdout)
Experimental design	Latin hypercube
Sample size	5'000
Validation metric	Q_2 (Eq. (4.33)) and RMSE (Eq. (4.53))
Testing	
Strategy	Independent data set (holdout)
Experimental design	Latin hypercube
Sample size	5'000

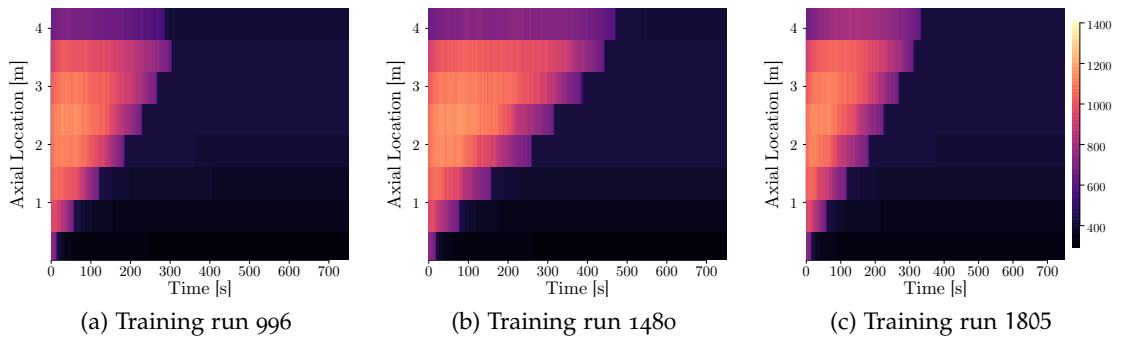


Figure 4.18: Examples of multivariate clad temperature output [K] at eight different locations as function of time, presented as “images”, taken from three different training runs.

PCA was used to reduce this significant number of output dimensions. Fig. 4.19 shows the sample mean surface and the first two PCs (loadings) estimated using 1'920 training samples. These two PCs explained about 83% of the output variance in training samples. This implied that any realization of the training samples could be reconstructed by using the mean surface added by linear combination of the PC multiplied by a unique set of scalars (the standardized PC scores) associated with that realization, such that on average (over many realizations) the reconstruction would be 83% accurate with respect to the RMSE. In other words, "images" in Fig. 4.18 can be reconstructed by overlapping the mean surface and the multiples of PCs "images" shown in Fig. 4.19.

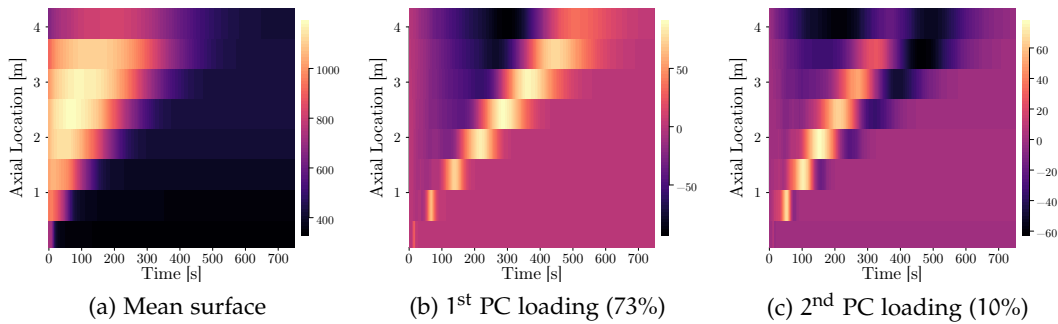


Figure 4.19: PCA results for the clad temperature output

Fig.4.20 summarizes the convergence behavior of the reconstruction error for the three types of output with increasing number of PCs (up to the first 10, out of 80'000, 40'000, and 10'000 PCs obtained for the respective outputs) used for the reconstruction. The plots were obtained from reconstructing the outputs in the validation data set (with sample size of 5'000) using a set of PCs derived from the training data set and by computing the average of the squared error over all realizations in the validation data set.

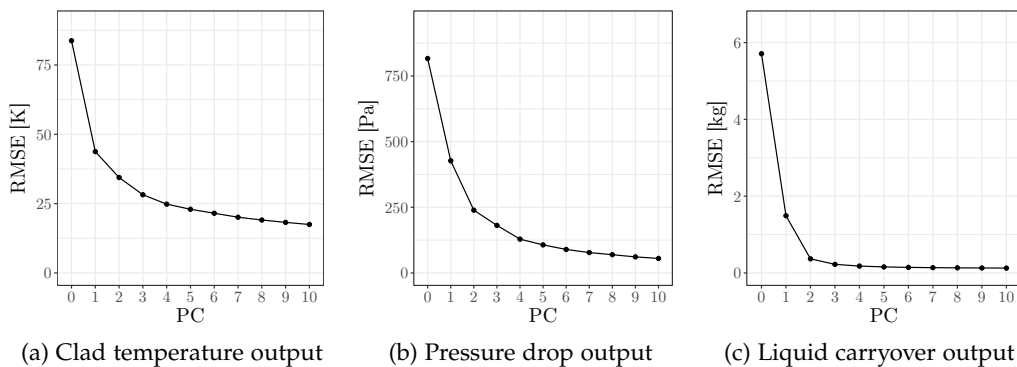


Figure 4.20: The reconstruction error, in terms of RMSE, as a function of the number of PCs used in the reconstruction of the output space for three different output types.

Because the PC scores used in the reconstruction were exact, the plot shows the magnitude of error to be expected from the dimension reduction procedure for each type of outputs. It also shows that the benefit of using larger number of PC is increasingly marginal.

4.6.3 GP PC Metamodel Construction: Training, Validation, and Selection

Following the PCA of the multivariate output, GP metamodels were constructed with respect to the standardized PC scores for each of the output types. The effect of several factors potentially affecting the predictive performance of the GP metamodels were also investigated, including training sample size as well as the choice of experimental design and covariance kernel function.

It was found that higher PCs tends to be harder to fit. That is, more and more training samples were required to have a GP metamodel of good predictive accuracy. Fig. 4.21 shows the predictivity coefficient Q_2 as function of the training sample size for three different PCs GP metamodels with respect to the clad temperature output. The Q_2 was calculated based on the validation data set of 5'000 data points (i.e., independent TRACE runs). The multiple points per training sample size correspond to GP metamodels constructed using different experimental designs, covariance kernel functions, and the five replications.

GP PC metamodel,
clad temperature
output

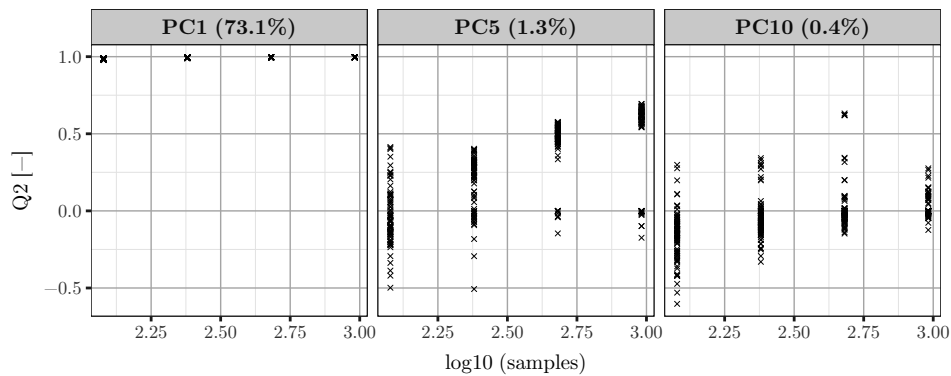


Figure 4.21: Convergence of PC metamodel with increasing number of training samples with respect to the PCs scores associated with the clad temperature output and the Q_2 validation metric. The number inside parentheses is the explained variance of the output by the PC and the cumulative explained variance of the first 10 PCs is about 96%.

The leftmost panel of the plot shows that indeed the GP metamodel for the first standardized PC score is the easiest to fit, requiring only a small size of training sample. On the other hand, the rightmost panel of the plot shows that not even the largest number of training samples considered is enough to have a GP metamodel with decent performance for the 10th PC. Furthermore, the plot also shows that the variations in the performance tends to become smaller with increasing sample size.

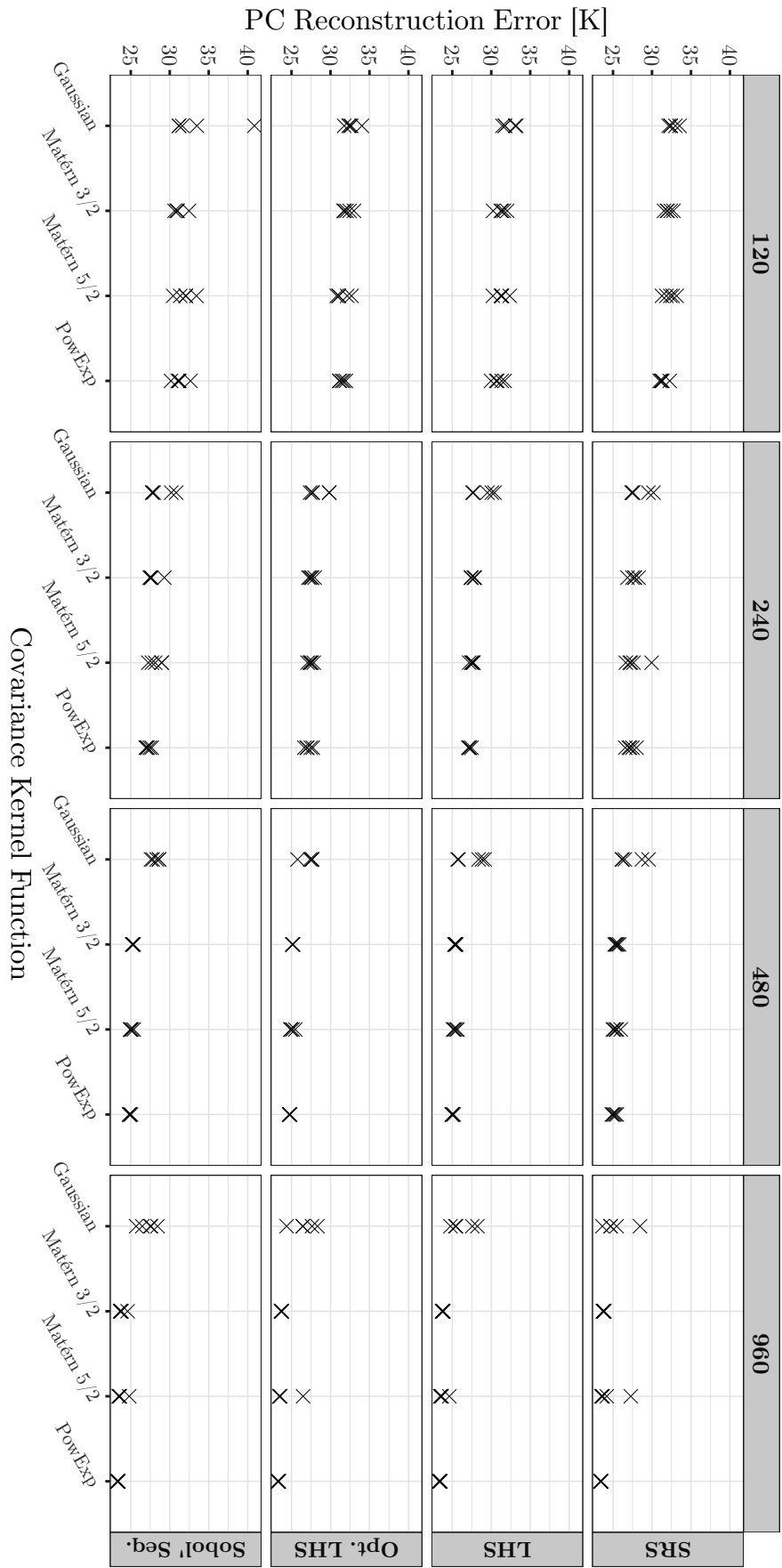


Figure 4.22: The effect of training sample size, experimental design, and covariance function on the predictive performance (in terms of RMSE) of GP PC metamodel with respect to the clad temperature output T.C. 7 PCs were used for the reconstruction.

Fig. 4.22 summarizes the effect of different training sample sizes, types of experimental design, and types of covariance kernel function on the performance of the constructed GP PC metamodels to predict the clad temperature output. The training samples were replicated five times for each size and for each design. The predictive performance was assessed in terms of the RMSE which was computed by retaining the first seven PCs and using the validation data set. Therefore, the RMSE shown in the figure represents the combined error due to PC truncation and misprediction of the PC scores.

The size of the training sample was the most important factor in determining the predictive performance of a GP PC metamodel. The choice of covariance function had some effects on the performance especially between the smoother covariance functions (i.e., the Gaussian and the Matérn 5/2) and the less smooth ones (i.e., the power exponential and the Matérn 3/2). GP metamodel constructed using the Gaussian covariance kernel function, in particular, exhibited significant variation in the performance of training sample replications compared to the other covariance kernel functions. Finally, the choice of experimental design for the training sample had a negligible effect on the predictive performance of the GP metamodel.

The GP PC metamodel to predict the pressure drop output showed the same behavior of being more difficult to fit for the higher PCs (See Fig. 4.23). The GP metamodel for the first standardized PC score remained the easiest to fit. However, the metamodel of the higher PC better converged than that for the clad temperature output. That is, a metamodel with decent predictive performance could be obtained for all first 10 standardized PCs scores using the considered sample sizes. Those 10 PCs carried close to 100% of the total output variance.

GP PC metamodel, pressure drop output

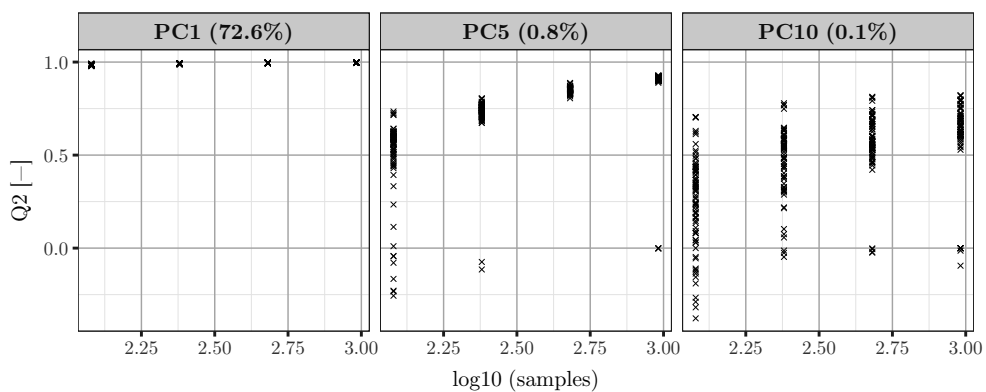


Figure 4.23: Convergence of PC metamodel with increasing number of training samples with respect to the PCs scores associated with the pressure drop output and the Q_2 validation metric. The number inside parentheses is the explained variance of the output by the PC and the cumulative explained variance of the first 10 PCs is about 99%.

*GP PC metamodel,
liquid carryover
output*

Finally, the liquid carryover output was found to be the easiest to construct. While the predictive performance of the GP metamodel with respect to the pressure drop output converged faster across the first five PC scores (Fig. 4.23), those PCs of the liquid carryover output contained almost all of the total output variance (Fig. 4.24).

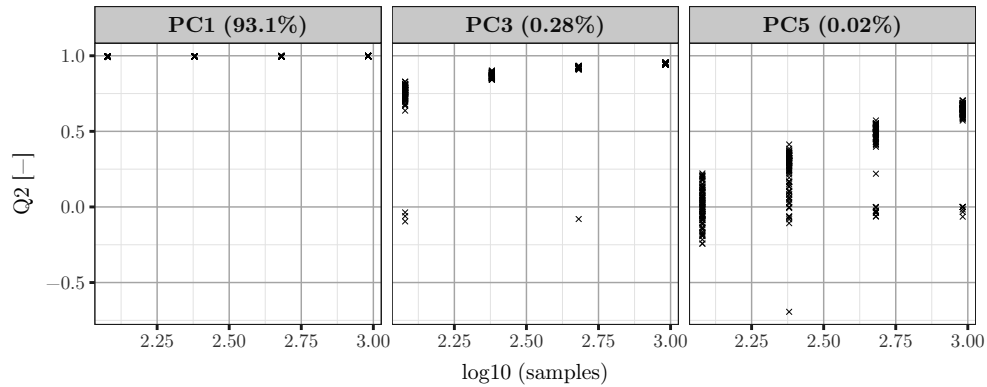


Figure 4.24: Convergence of GP metamodel with increasing number of training samples with respect to the standardized PCs scores associated with liquid carryover output and the Q_2 validation metric. The number inside parentheses is the explained variance of the output by the PC and the cumulative explained variance of the first five PCs is about 99%.

The effect of different training sample sizes, types of experimental design, and types of covariance kernel function on the predictive performance of the constructed GP PC were also investigated with respect to the pressure drop and the liquid carryover outputs. Ten and five PCs were used to compute the predicted reconstruction error for the pressure drop and the liquid carryover outputs, respectively. The findings for these two outputs were also similar to the ones for the clad temperature output: the training sample size was the most important factor in determining the predictive performance, there was a relatively minor effect of the choice of covariance kernel functions especially in terms of the performance variation (with an exception of the Gaussian kernel function which has the worst performance in terms of consistency across replications), and the choice of experimental design was relatively noninfluential. Figs. B.19 and B.20 in the appendix summarize these effects for the two outputs.

4.6.4 GP PC Metamodel Testing

Based on the results presented above, a final set of GP metamodels was constructed with a larger training set of 1'920 samples using a power-exponential covariance kernel function based on a Sobol' sequence (the best options found). Furthermore, 7, 10, and 5 PCs were used in the reconstruction of the clad temperature, pressure drop, and liquid carryover outputs, respectively. As such there were 22 sep-

arate GP PC metamodels. An additional (i.e., *testing*) data set of size 5'000 was then independently generated (with actual TRACE code runs) and used as the basis for testing the predictive performance of the final model. This additional step was done to avoid any possible bias due to the fact that the validation data set was already used to select the final metamodel.

The validation metric Q_2 of the metamodels with respect to the standardized PCs scores computed on the testing data set converged for all types of output (Fig. 4.25). In other words, the size of the testing dataset was found to be (or more than) sufficient to assess the predictive performance of the selected metamodel.

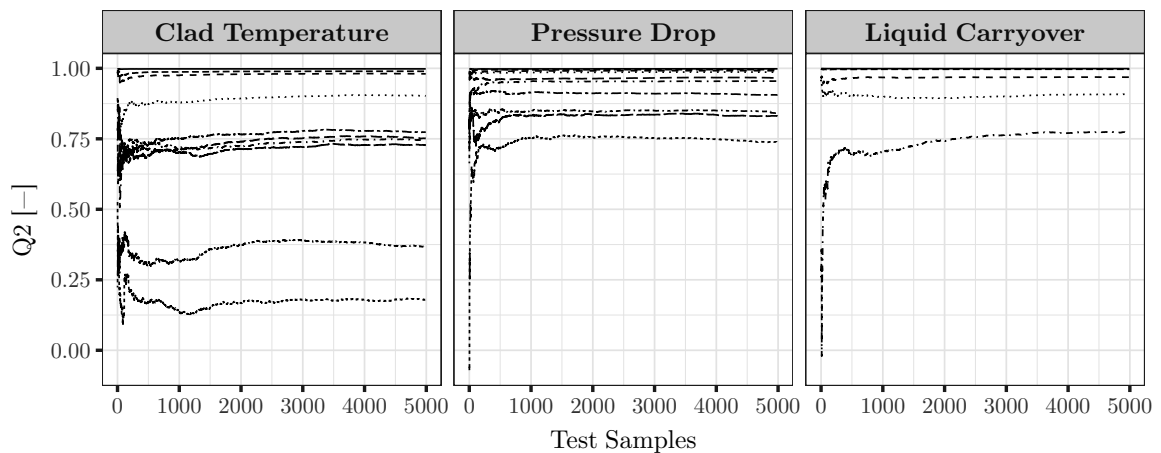


Figure 4.25: Convergence of the predictive performance of the metamodel with respect to the standardized PCs scores for each output type. Shown above are the first 10 PCs for the clad temperature and pressure drop outputs and the first five PCs for the liquid carryover output. For the clad temperature output, the predictivity coefficient falls below 0.75 after the first seven PCs.

There were two main sources of error that dictated the predictive performance of a GP PC metamodel. The first was due to the representation of the full output dimension with only a few selected PCs (i.e., the dimension reduction) and the second was due to the misprediction of the standardized PC scores by the GP metamodel (i.e., the functional approximation). Fig. 4.26 illustrates these errors by presenting the predicted and observed reconstruction error (in terms of RMSE) for each realization in the testing data set. Note that the *observed* reconstruction error was obtained using the reference standardized PC scores of the testing data set, while the *predicted* reconstruction error was obtained using the standardized PC scores as predicted by the GP metamodels.

The extend of the x-axis signifies the range of error due to the PC truncation. The farther a data point is from the left, the larger the error is due to the dimension reduction. On the other hand, the extend of the y-axis, specifically the vertical distance between the data points

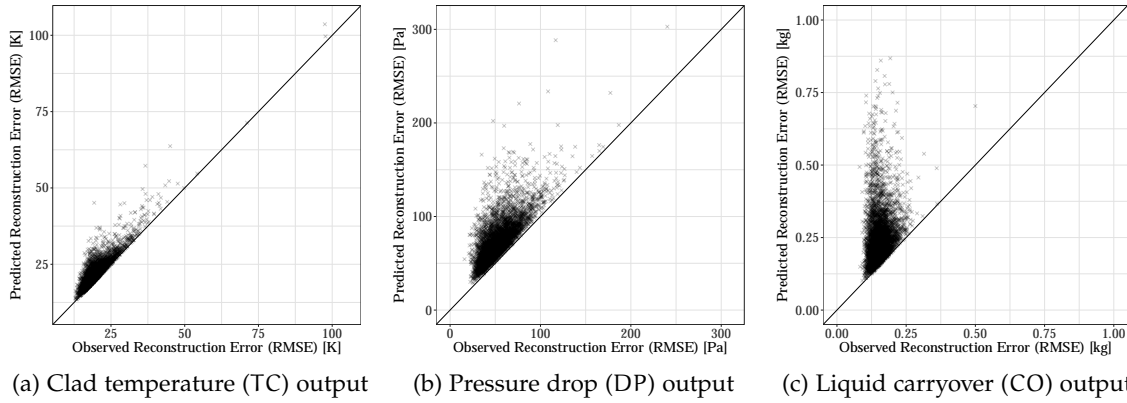


Figure 4.26: Errors, predicted and observed, due to the dimension reduction procedure (PCA) and the functional approximation (GP) for the three types of output.

and the line, signifies the error due to the misprediction of the standardized PCs scores by the GP metamodel. Data points which are located along the line implied a perfect prediction by the GP metamodel. The farther a data point is from the line, the larger the error is due to the metamodel approximation. As can be seen, no data point is located below the line as the truncation error sets the limit of the metamodel predictive performance. Furthermore, though some data points (i.e., realizations) might be mispredicted and lie over a wide range of value, the cloud of the data points is only concentrated around a particular range of value. Table 4.2 numerically summarizes the results of the testing step. For comparison the standard deviation of the testing data set for each output is also given.

Table 4.2: Predictive performance of the selected GP PC metamodel on the testing dataset of size 5'000

Output	PC _{max}	Predictivity Coefficient		Reconstruction Error		Test Data
		Q ₂ PC ₁	Q ₂ PC _{max}	RMSE _{obs}	RMSE _{pred}	Std. Dev.
TC	7	≈ 1.0	0.77	20.17 [K]	22.43 [K]	254.0 [K]
DP	10	≈ 1.0	0.74	55.57 [Pa]	77.95 [Pa]	9200.0 [Pa]
CO	5	≈ 1.0	0.77	0.16 [kg]	0.27 [kg]	30.4 [kg]

4.6.5 Discussion

The selection of the number of PCs to retain is usually done by justifying the amount of total variance explained by the selected PCs. The notion of reconstruction error more intuitively explains the notion of explained variance. The error represents the difference between the original data and reconstructed data using only a small number of

PCs. The series of plots shown in Fig. 4.20 also illustrates the limit of PCA as a dimension reduction tool. The method performs best for the liquid carryover output and worst for the clad temperature output. The latter is mostly due to the fact that the clad temperature output includes a sharp discontinuity (i.e., quenching). PCA, being a linear transformation, deals with this strong non-linearity sub-optimally. That is, a significantly large number of PCs are required to resolve the discontinuity and bring the reconstruction error closer to 0.

However, as indicated in Figs. 4.21, 4.23, and 4.24, constructing a PC metamodel is increasingly difficult for higher PCs for all output types. In other words, as the relationship between model parameters and the standardized PC scores becomes increasingly non-linear, large number of training samples are required to train the GP metamodel to attain a decent predictive performance. At the same time, the benefit of adding PCs becomes increasingly marginal (Fig. 4.20). In addition, some degree of error should be expected in the prediction of the PC scores by the metamodel, especially the higher ones. As such, unless the score is perfectly predicted, this error might offset the potential benefit of adding PCs for the reconstruction.

*GP PC metamodel,
limitation*

A pragmatic approach is thus to choose the number of retained PCs based on some target error or number of TRACE runs that can be afforded. To put the error into context, the reconstruction error of an output can be compared to the standard deviation of the output in the test data itself. This standard deviation, in turn, serves as a measure of the output variation due to the variation in the input parameters. For the clad temperature output, retaining seven PCs for the GP PC metamodel gives a reconstruction error of about 22 [K] (RMSE). This value is small in comparison with the standard deviation of the output in the test data, 254 [K] (less than 9%, Table 4.2). The same is true for the two other outputs. For the pressure drop output, 10 PCs gives a reconstruction error of about 78 [Pa], compared with the test data standard deviation of 9200 [Pa] (less than 0.9%); for the liquid carryover output, five PCs gives a reconstruction error of about 0.27 [kg], compared with the test data standard deviation of 30.4 [kg] (less than 0.9%). Note that those numbers are based on a training sample of size 1'920 and a testing sample of size 5'000. As shown in Fig. 4.25, the size of the testing sample is more than enough to obtain stable estimates for these errors.

*GP PC metamodel,
errors in context*

Another important finding in this study is the major importance of the training sample size on the predictive performance of the GP PC metamodel. The choice of covariance function has some effect in the predictive performance and its variation across replications. Insofar the choice is between the smoother functions (e.g., the Gaussian, the Matérn 5/2) and the rougher ones (e.g., the power-exponential, the Matérn 3/2), with the performance of the smoother kernels tending to be more variable. The Gaussian covariance kernel showed a partic-

*Effects of training
sample size,
experimental design,
and covariance
function*

ularly inconsistent predictive performance over multiple replications for all types of output, and thus should be avoided. The choice of experimental design, on the other hand, has a marginal effect on the predictive performance.

GP PC metamodel, a statistical metamodel

It is also worth noting that a GP PC metamodel is a global statistical metamodel. This implies that its predictive performance is defined over all output space (such as through the use of Q_2 and RMSE as the validation metrics.) and over many realizations. That is, a good metamodel accurately predicts the output for an arbitrary input, *on average*. This also means that the metamodel has to some extent a “hit-and-miss” property: most realizations are accurately predicted, some realizations can be mispredicted, and some small proportion of that can be grossly mispredicted as was illustrated in Fig. 4.26. Fig. 4.27 illustrates this idea further for the clad temperature output.

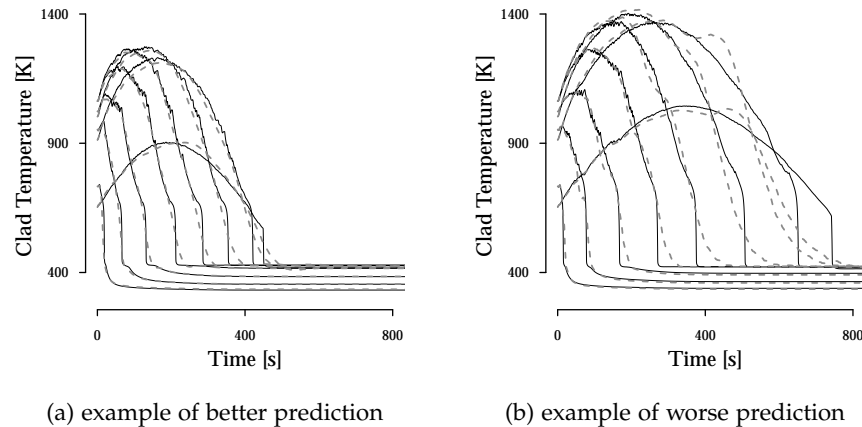


Figure 4.27: GP PC metamodel is a global statistical metamodel which gives global accurate prediction *on average*. Some realizations are better predicted than others, due to both the limitation in the approximations incurred by using PC and GP (e.g., around quenching). Solid and dashed lines are TRACE runs and GP PC predictions, respectively.

Computational cost

A prediction made by a fully specified GP metamodel is, according to Eq. (4.25), a straightforward matrix operation. In R through the package *DiceKriging*, the operation to predict a standardized PC score for an arbitrary input takes about 0.05 [s]. For a full output reconstruction, this operation has to be repeated for multiple PCs scores before being multiplied with the PC loadings. The actual time required for this full reconstruction is specific to a particular implementation and to a particular programming language. Though only rudimentary investigations were carried out, the cost of evaluating the metamodel for an arbitrary input is still expected to be much less than running an actual TRACE simulation (6 – 14 [min]). For instance, the most naive implementation in R takes, on average, less than 5 [s] to

predict and reconstruct all outputs (clad temperature, pressure drop, and liquid carryover) for a given input.

However, it is also important to take into account the computational cost required to train, validate, and test the metamodel. The training, validation, and testing data sets have to be generated from actual TRACE runs. As explained, the size of the training sample data should be as large as the computational budget of running the actual codes allowed. In this regard, it is also worth noting that the model fitting step during training is an optimization problem that becomes computationally expensive for large training sample of large dimensions (large number of input parameters). Again, further study is required to have a more quantitative cost measure with respect to this optimization.

Finally, for any given training sample size, the predictive performance of the metamodel is assessed in the validation and testing steps. The former is aimed for selecting the best metamodel (among metamodels constructed with different settings), while the latter is aimed for estimating the true error expected from using the selected metamodel as a surrogate. The study used a large validation and testing data sets (each with 5'000 data points), but according to Fig. 4.25, that many points might not have been necessary. The size of validation/testing data set can be optimized by first making a plot similar to the one in Fig. 4.25 but with an initial small number of samples to first check the convergence of the error estimate before creating unnecessarily large set upfront.

4.7 CHAPTER SUMMARY

The functional approximation part of the proposed statistical framework has been presented in this chapter. The goal of such an approximation was to evaluate the output of a computer simulation code for an arbitrary input (much) faster. The approximation is based on Gaussian stochastic process resulting in a statistical metamodel. As the dimensionality of the output is large, in the order of tens of thousands, a dimension reduction step is adopted by means of PCA (an approach similar to what was adopted in Chapter 3).

The results obtained on the TRACE model of FEBA is reasonable. Though the prediction error can at times be large, the metamodel gives an overall good performance on average for the three types of multivariate output (clad temperature, pressure drop, and liquid carryover). The metamodels for both pressure drop and liquid carryover outputs have less than 0.9% prediction error (RMSE), while the metamodel for the clad temperature output has less than 9% prediction error (RMSE); these errors are relative to the standard deviation of the respective outputs in the testing data set. The larger error for predicting the clad temperature output highlights the limitation of the

approach for outputs that exhibit strong non-linearity and discontinuity (such as the quenching in the clad temperature transient). This, in turn, is due to the use of PCA as the (linear) dimension reduction tool. As such, a first step of improvement in this regard can be aimed toward replacing PCA with another, more advanced dimension reduction tool.

Using the GP PC metamodel as the surrogate for TRACE run, the prediction for arbitrary model parameters values can be made much faster (< 5 [s] per metamodel evaluation vs. $6 - 15$ [min] per TRACE run). As such the metamodel constructed in this chapter can be used as the basis for Bayesian model calibration which requires tens if not hundreds of thousands function evaluations. However, it is also important to note that the time required for the construction of the metamodel as well as for its convergence study has to be taken into account. The training, validation, and testing data have to be generated from actual code runs. Additionally, the model fitting step to estimate GP metamodel hyper-parameters is an optimization problem that can easily become expensive for large training sample of large dimensions (large number of input parameters).

On a different note, the study confirms that the size of the training data is the main factor in determining the predictive performance of the metamodel. As a result, the size of the training data should be as large as the computational budget allowed. At the same time, the choice of covariance function has some impact especially in relation to the stability of the performance. Regarding this, the power-exponential and Matérn covariance kernel functions are preferred, while the Gaussian kernel should be avoided. Finally, the choice of experimental design has a negligible impact on the predictive performance of the metamodel.

BAYESIAN CALIBRATION OF COMPUTER MODEL: BRIDGING MODEL & DATA UNDER UNCERTAINTY

In Chapter 3, a sensitivity analysis method was employed to better understand the inputs/outputs relationship in a computer simulation model with uncertain inputs. The method was also able to reduce the size of the problem by screening out noninfluential inputs. Chapter 4 then developed a fast approximation to evaluate the output at any given input point, in anticipation of the high cost of the calibration approach presented in this chapter. The respective methods were exemplified by their application to a TRACE reflood simulation model whose inputs were uncertain, as assumed in Chapter 2.

This chapter deals with a statistical framework for calibrating the inputs of a simulation model. The framework casts the calibration problem as a statistical inverse problem where the initial (*prior*) uncertainties of the inputs are updated based on available observed data. It considers the a priori uncertainties in the inputs and in the experimental data, as well as the possible bias of the model. The inputs uncertainties are then coherently updated via the Bayes' theorem resulting in an updated (*posterior*) probability density. The updated uncertainty of the inputs can then be propagated through the simulation model to quantify the prediction uncertainty.

Section 5.1 first presents the statistical framework for the problem of computer model calibration, while Section 5.2 elaborates further the formulation of the calibration problem through probabilistic modeling of the data-generating process. This results in the formulation of the posterior probability density. The posterior density is often a complex highly multi-dimensional function, which makes it difficult to work with. Section 5.3 presents a simulation method (i.e., Markov Chain Monte Carlo (MCMC) simulation) to directly generate representative samples from the posterior density. These samples can be used to approximate the posterior density or for uncertainty propagation. Important aspects of analyzing samples of a Markov chain are presented in Section 5.4. Section 5.5 then discusses the application of the approach to the FEBA TRACE reflood simulation model to constrain the prior uncertainty range of the model parameters based on the available experimental data. To do so, different types of experimental data (i.e., clad temperature, pressure drop, and liquid carryover) are used and their ability to constrain the prior range is investigated. The resulting posterior uncertainty, derived from one set of experimental condition, is verified by propagating it on the other FEBA tests. Finally, Section 5.6 concludes the chapter.

5.1 STATISTICAL FRAMEWORK

Calibration
framework

The calibration framework in this thesis is in line with the seminal work of Kennedy and O’Hagan [83], which is adapted by many in the applied literature [81, 88, 92, 93, 205]. Meanwhile, the explicit formulation here uses a set of notations adapted from different sources [35, 83, 112, 205, 206]. Suppose an experiment on a physical system is being conducted and, in parallel to that, a computer simulator of the system is available. Let y_E be the experimental observation of the system response (i.e., the quantity of interest (QoI)) taken at *controllable* inputs \mathbf{x}_c , then its relationship to the true *unknown* response value y_T is given by

$$y_E(\mathbf{x}_c, \boldsymbol{\lambda}) = y_T(\mathbf{x}_c, \boldsymbol{\lambda}) + \epsilon(\boldsymbol{\lambda}) \quad (5.1)$$

where ϵ is an observation error; and $\boldsymbol{\lambda}$ is an element of an observation layout $\boldsymbol{\Lambda}$ detailed below. The true value, in turn, is linked to the prediction made by the computer simulator y_M by

$$y_T(\mathbf{x}_c, \boldsymbol{\lambda}) = y_M(\mathbf{x}_c, \hat{\mathbf{x}}_m, \boldsymbol{\lambda}) + \delta(\mathbf{x}_c, \boldsymbol{\lambda}) \quad (5.2)$$

Model bias

where δ is the *model bias*, defined as the difference between the true response value and the simulator prediction made by using $\hat{\mathbf{x}}_m$, the best (“true”) value of the *model* parameters. This term, if any, represents the discrepancy in the prediction due to missing physics, numerical approximation, etc. Combining the two relationships yields,

$$y_E(\mathbf{x}_c, \boldsymbol{\lambda}) = y_M(\mathbf{x}_c, \hat{\mathbf{x}}_m, \boldsymbol{\lambda}) + \delta(\mathbf{x}_c, \boldsymbol{\lambda}) + \epsilon(\boldsymbol{\lambda}) \quad (5.3)$$

Goal of model
calibration

The goal of model calibration is, broadly speaking, to learn the true (but unknown) model parameters such that the agreement between the simulator prediction and the experimental observation is improved [83, 86]. The parameters involved in the representations above are *controllable* inputs \mathbf{x}_c , the best value of *model* parameters $\hat{\mathbf{x}}_m$, and an element $\boldsymbol{\lambda}$ of an *observation layout* $\boldsymbol{\Lambda}$. The relationship between elements of Eq. (5.3) is depicted in Fig. 5.1 below.

Observation layout

An observation layout $\boldsymbol{\Lambda}$ is an ordered set and it defines which of the different types of QoI are observed (or predicted) as well as their locations and time points. In this manner, multivariate QoIs can be represented using vectors [205]. For instance, the observation layout $\boldsymbol{\Lambda} = \{(A, z_1, t_1), (B, z_1, t_1), (A, z_1, t_2)\}$ might be used to signify QoIs (observed or predicted) of type A at time t_1 , type B at time t_1 , and type A at time t_2 ; all are taken at location z_1 . The vectors $\mathbf{y}_M(\circ, \boldsymbol{\lambda})$ and $\mathbf{y}_E(\circ, \boldsymbol{\lambda})$ for $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}$ then refer to the model prediction and experimental data given by the element $\boldsymbol{\lambda}$ of the set $\boldsymbol{\Lambda}$, respectively.

Controllable inputs

Departing from the previous chapters, this chapter categorically distinguishes two types of input parameters: *controllable inputs* \mathbf{x}_c and *model parameters* \mathbf{x}_m . Controllable inputs (or design variables) are pa-

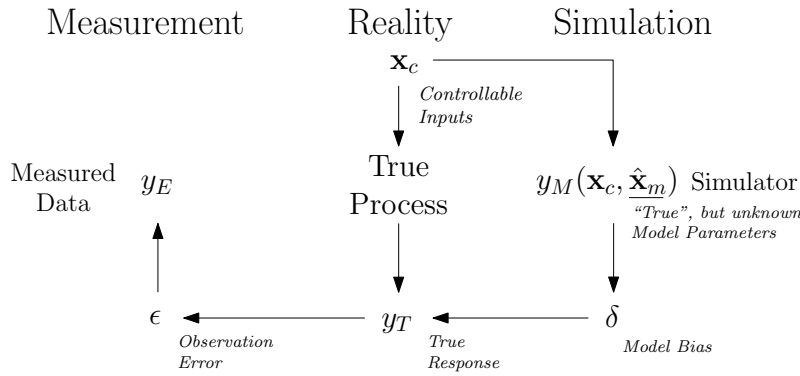


Figure 5.1: Relationships between elements of the calibration formulation (adapted from [206]).

rameters that, in the context of a controlled experiment, can be varied by the experimentalist. Being controllable also implies that the parameters can be observed in the actual experiment. Both in the physical experiment and in the simulation their values are often varied either to investigate the system (respectively, the simulated system) behavior under the change or to find the setting that gives the best system performance. An example of such parameters is the parameters related to boundary conditions of an experiment.

Model parameters refer to parameters that are specific to a particular parametrization of the model in the simulator. As such, they only appear in the term y_M of Eq. (5.3). Model parameters might or might not have a physical meaning; that is, the parameters have interpretation outside the context of the physical model in which the parameters reside, or the parameters are simply used to tune the model such that the prediction agrees better with the observed data (thus become a measure for model inadequacy of a particular model). The parameters are referred to as *physical parameters* in the former case, and as *tuning parameters* in the latter. In the following, however, such distinction is merely conceptual; these parameters are in practice not known a priori and not directly observable with respect to the experiment. The generic goal of model calibration is then to obtain an optimal value of the model parameters \hat{x}_m based on a set of experimental data taken at particular values of x_c and λ . This distinction will be revisited in Section 5.2.1. The notion of the *true* value is usually reserved for the optimal value of a physical parameter [207] and the term *best* or *best-fitting* value is for a tuning parameter [93]. Contrary to the controllable inputs, calibrated model parameters should in principle be applicable for all instances of the simulator application.

The formulation given in Eq. (5.3) contains two unknowns, namely the best value of the model parameters \hat{x}_m and the model bias δ . In the Bayesian statistical framework, any unknown is considered *uncertain* and assigned a prior probability distribution. This prior probability assignment also applies for other terms that might also not be per-

Model parameters

Bayesian statistical calibration

fectly known, such as the controllable inputs \mathbf{x}_c or the observation error ϵ . The goal of calibration is then to update the prior uncertainties on the model parameters based on a comparison between experimental data and simulator predictions. As such, the calibration process becomes a Bayesian inference. Consequently, the process takes into account multiple sources of uncertainty in the result. Appropriately acknowledging multiple sources of uncertainty in the calibration process can provide a hedge against *overfitting*, where the calibrated parameters are overly specific to the data used for the calibration and not applicable in a different setting of controllable inputs.

Bayesian inference

Two steps are involved in conducting a Bayesian inference. The first is a formulation of a probabilistic model for all the terms in Eq. (5.3). In essence, the resulting model represents a data-generating process of the observed experimental data, incorporating the elements of Eq. (5.3) into it. An approach to formulate a probabilistic model is presented in Section 5.2. The probabilistic model is then conditioned on the given experimental data to obtain an updated (*posterior*) probability distribution of the model parameters. Dealing with a high-dimensional arbitrary probability distribution is difficult and most of the computations involving the posterior probability distribution resort to directly generating samples from it. Indeed, the computation of the posterior distribution or any transformation of it (e.g., variance of a function) is the second step of the inference and is presented in Section 5.3.

5.2 BAYESIAN FORMULATION OF CALIBRATION PROBLEM

The Bayesian framework for model calibration begins by constructing a probabilistic model of y_E given in an additive formulation of Eq. (5.3). That is, it aims at formulating the data generating process $\mathcal{Y}_E(\mathbf{x}_c; \lambda)$. This model implies that the experimental data y_E taken at particular \mathbf{x}_c observed at λ is a realization of a stochastic process. Furthermore, this probabilistic modeling entails casting any *uncertain* element in Eq. (5.3) either as random variable or stochastic process.

5.2.1 Probabilistic Model for the Model Bias Term

Recall the relationship between the true system response and its prediction by a simulator (Eq. (5.2)) rearranged below:

$$\delta(\mathbf{x}_c, \lambda) = y_T(\mathbf{x}_c, \lambda) - y_M(\mathbf{x}_c, \hat{\mathbf{x}}_m, \lambda)$$

where the prediction y_M is made using the best but unknown value of the model parameters. As such, the model bias function δ represents a possible systematic difference between the true system response and the simulator prediction that still remains, even from using a simulator with the *best* set of model parameter values. Possible sources for this bias are missing physics in the physical models,

Model bias, possible origins

numerical approximations, or any other simplifications in the simulator whose effects on the prediction are unknown a priori. As such, the bias term tends to be systematic and dependent on the controllable inputs \mathbf{x}_c and the observation layout Λ [205]. Note that, strictly speaking, there is a dependence of $\hat{\mathbf{x}}_m$ on δ , but this dependence is suppressed from the notation; $\hat{\mathbf{x}}_m$, though unknown, should in principle be a unique set of values valid for all \mathbf{x}_c [92, 93].

The unknown model bias function δ can be represented as a random function $\mathcal{D}(\circ)$,

$$(\mathcal{Y}_T - y_M(\mathbf{x}_c, \hat{\mathbf{x}}_m, \lambda)) \equiv \mathcal{D}(\mathbf{x}_c, \lambda) \quad (5.4)$$

Casting the unknown model bias term as a stochastic process is the salient feature of Bayesian calibration framework proposed by Kennedy and O'Hagan [83]. In particular, a stationary Gaussian process (GP) $\mathcal{D}(\mathbf{x}_c, \lambda)$ on $\mathbf{X}_C \subseteq \mathbb{R}^{D_c}$ and on Λ is used to represent the term:

*Gaussian process
formulation*

$$\mathcal{D}(\circ, \circ) \sim \mathcal{GP}(\mathbf{m}_\delta(\circ, \circ; \boldsymbol{\psi}_\delta), \mathbf{K}_\delta((\circ, \circ), (\circ, \circ); \boldsymbol{\psi}_\delta)) \quad (5.5)$$

where \mathbf{m}_δ and \mathbf{K}_δ are the mean function and the covariance function of the GP, respectively; and $\boldsymbol{\psi}_\delta$ is the hyper-parameters associated with the specification of the GP for the model bias function (e.g., its covariance kernel, $\{\sigma, \theta, \rho\}$, see Chapter 4). Under a GP formulation, the notion of *systematic* bias mentioned previously is described statistically in terms of the mean and the covariance of the GP [205].

For a selected values of \mathbf{x}_c and λ , the GP becomes a Gaussian random variable,

$$\mathcal{D}(\mathbf{x}_c, \lambda) \sim \mathcal{N}(\mathbf{m}_\delta(\mathbf{x}_c, \lambda; \boldsymbol{\psi}_\delta), s_\delta^2(\mathbf{x}_c, \lambda; \boldsymbol{\psi}_\delta)) \quad (5.6)$$

where s_δ^2 is the standard deviation at controllable input \mathbf{x}_c observed at λ , under the parametrization $\boldsymbol{\Psi}_\delta$ of the GP. Finally, for observations on multiple combinations of the controllable inputs or the complete observation layout Λ , the GP becomes a multivariate Gaussian random variable, taking into account correlations of the bias at different elements of the observation layout,

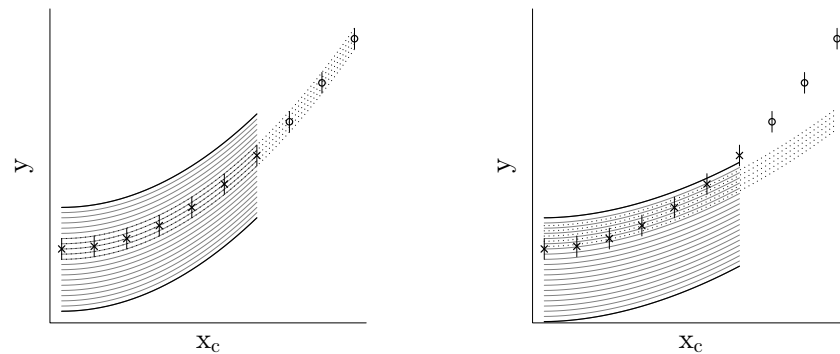
$$\mathcal{D}(\mathbf{x}_c, \Lambda) \sim \mathcal{N}(\mathbf{m}_\delta(\mathbf{x}_c, \Lambda; \boldsymbol{\psi}_\delta), \boldsymbol{\Sigma}_\delta(\mathbf{x}_c, \Lambda; \boldsymbol{\psi}_\delta)) \quad (5.7)$$

where $\boldsymbol{\Sigma}_\delta$ is the (symmetric) covariance matrix of the bias at controllable input \mathbf{x}_c observed on Λ , under the parametrization $\boldsymbol{\Psi}_\delta$ of the GP. The size of the matrix is $P \times P$, with P the product of the number of different combinations of the controllable inputs and the number of elements in the observation layout.

Incorporating a bias term in the calibration procedure is important to avoid overfitting in the model parameters estimates. To illustrate this idea, consider a calibration process for a simulator whose bias is negligible, with a single uncertain model parameter and a single

*Model with
negligible bias,
illustrated*

controllable input x_c as shown in Fig. 5.2a. The thin black lines between the two bounding thick black lines indicate the simulator predictions at different values of the model parameter. As can be seen, the range of the model parameter values can in principle be constrained to match the observed data (crosses) within the observation uncertainty. Furthermore, the range of the model parameters will increasingly become smaller with increasing number of data (such that the associated observation uncertainty becomes increasingly narrow as well). In other words, the calibrated model parameter converges to the “true” value [93, 207, 208]. This parameter value will be valid for prediction outside the calibration domain (i.e., extrapolation at different values of controllable inputs where no data has been observed).



(a) Simulator with negligible bias and calibrated simulator.

(b) Simulator with non-negligible bias and calibrated simulator.

Figure 5.2: Illustration of predictions made by computer simulator with and without non-negligible bias, both with an uncertain model parameter and a controllable input x_c . Crosses are the observed data along with the associated uncertainty taken at different controllable inputs x_c . Bold lines are the simulator prediction using the maximum and minimum of the uncertain model parameter, thin lines are the prediction with different values of the model parameter, and dotted lines are the prediction outside the calibration domain using model parameter calibrated without a bias term. The scales in the axes are arbitrary.

On the other hand, some simulators would have an apparent bias such that their predictions would remain inconsistent with the observed data, regardless of the choice of the model parameter values (Fig. 5.2b). Calibration can still be conducted such that the discrepancy between data and prediction is minimized in some sense (i.e., some kind of *best-fitting* model parameter value). The calibrated parameter would be able to predict calibration data well, but not for prediction outside the calibration domain. The situation becomes more problematic when more precise data becomes available such that un-

Model with non-negligible bias, illustrated

certainty associated with the observed data becomes narrower. In that situation, the uncertainty associated with the calibrated model parameter will also become narrower up to a point value (in this particular example). This illustrates the two symptoms of overfitting the model parameter: the calibrated model parameter is *biased* (i.e., having a wrong value) and its uncertainty is *degenerate* (i.e., increasingly sure on the wrong value with higher precision of the observed data). The latter symptom is particularly troublesome as it inflates the degree of confidence one has on the prediction.

Overfitting

The situation of a biased model is prevalent in complex physics-based simulators, whose constituent physical models were developed using scientific theory and supported by experimental data. This approach forms the scientific basis for making prediction, especially in the region outside the calibration domain [209]. It is hoped that such an approach would be more robust than using purely statistical model of observed data [93, 205]. However, certain degree of simplifications from numerical approximation to ignored physical process due to a lack of knowledge are expected to persist [2]. Furthermore, the strong scientific foundation and the experimental data support of physical models often only apply to the separate constituent models of a complex simulator [48]. In practice, the simulator consolidates numerous models to simulate the behavior of a (more) complex system outside the calibration domain. As such, it can also be expected that the predictions from such simulators would exhibit certain degree of bias (from the true value) that is unknown a priori.

*Physics-based
simulators*

One might argue that if a model is known to be biased it simply requires more developmental effort to correct the bias by putting additional models for the missing physical processes. However, as argued in [11, 209], this approach might not be the best solution as additional models often require even more model parameters to be calibrated and thus call for even more supporting data that cannot be met. Additionally, as noted in [48, 93, 208], it is often impractical (nor realistic) for an analyst to revise the inner workings of a large complex simulator. Yet, to wait until a better simulator is available before making any prediction is simply not constructive.

In a Bayesian framework, the statistical description of the model bias term can potentially alleviate the problem of overfitting. Because the model parameters and the model bias are not fully identifiable according to Eq. (5.2)¹, having more precise data will not make the uncertainty associated with the calibrated model parameters to collapse (i.e., its distribution becomes degenerate) [93, 208]. Whether the calibrated model parameters and the associated uncertainties are applicable for extrapolation outside the calibration domain, however, depends on whether the bias term is modeled properly [86, 92, 93,

*Statistical
description of the
model bias*

¹ that is, without further prior information, arbitrary choice of \hat{x}_m fits the data perfectly well for arbitrary choice of δ . In other words, the two terms are *confounds*.

207–209]. Thus, such a statistical description of the model bias is not a magic bullet in the calibration of a biased model. It does, however, provides additional flexibility in incorporating either prior knowledge or a prior expectation regarding model deficiency.

*Physical parameters,
“true” values*

At this point, it is worth revisiting the meaning of calibrated parameters in a simulator with bias. In a simulator without bias it is straightforward to justify that the calibrated model parameters as the “true” parameter values of the specified model. If the model is physics-based then they also correspond to physical parameters. As argued in [207, 208], physical parameters often have meaning outside the world described by the model where the parameters reside. Moreover, having a true value, such physical parameters would generally be applicable to extrapolation outside the calibration domain.

*Tuning parameters,
best-fitting values*

On the contrary, as illustrated in one of the examples above, calibrated model parameters in a simulator with bias act as best-fitting parameters that allow the simulator to fit, in some sense, the calibration data. Incorporating model bias term might help in alleviating the problem of overfitting, but the a priori arbitrariness of the model bias term confounds with the model parameters itself, making the resulting calibrated model parameters more difficult to interpret [210]. As such, in practice, it is important to emphasize that calibrated model parameters in a simulator with bias will simply be optimal under particular assumptions (e.g., criteria, model bias term) [48]. Ref. [208] went further by arguing that such model parameters (tuning) had limited scientific values and would not help for extrapolation.

A pragmatic view

Regarding this dichotomy, the present thesis takes a more pragmatic view: the distinction is rather irrelevant. It is awkward to discuss the true and wrong values of model parameters if the model itself is considered biased (i.e., to a certain extent *wrong*). In such cases, the notion of true parameter values is hard to justify, the model parameters might not have strict physical meaning and may not be of interest in their own right. And yet, in a complex physics-based simulator (where possible systematic bias cannot be excluded), many of these model parameters are being used in conditions different from their calibration domain, regardless of the conceptual distinction (e.g., Refs. [28, 92]). Thus, the calibration of model parameters based on the available experimental data should be aimed at guaranteeing that the simulator remains applicable outside its calibration domain². The Bayesian framework accommodates this aim of calibration in a flexible manner by taking into account multiple sources of uncertainty through selection of prior uncertainties both for the model parameters and for the model bias term, which eventually results in the associated posterior uncertainties.

² or more eloquently in the words of Leamer [40], the resulting posterior uncertainty associated with the calibrated model parameters is: “...wide enough to be credible and the corresponding interval of inferences is narrow enough to be useful”.

5.2.2 Probabilistic Model for the Observation error

Now recall the relationship between the true system response and its observation through a measurement given in Eq. (5.1),

$$y_E(\mathbf{x}_c, \boldsymbol{\lambda}) = y_T(\mathbf{x}_c, \boldsymbol{\lambda}) + \epsilon$$

The observation error term ϵ represents any possible error during the measurement process, either from the imprecision of the instrument or any other residual variability of the experiment. This variability, in turn, might be due to the inherently stochastic nature of the physical process (irreducible) or unrecognized and uncontrolled variables (reducible) [83].

*Observation error,
possible origins*

Because this term is considered unknown, a stochastic process is defined on the observation layout,

$$\mathcal{E}(\boldsymbol{\lambda}) \sim p(\epsilon | \boldsymbol{\psi}_\epsilon, \boldsymbol{\lambda}) \quad (5.8)$$

where $\boldsymbol{\psi}_\epsilon$ is the parametrization of the PDF describing the observation error ϵ at $\boldsymbol{\lambda}$. That is, it depends on which response is observed, as well as where and when it is observed.

An important assumption made on the distribution of the observation error is that it is independent conditional on the true value of the system response. One can argue that the measurement data points taken from a spatio-temporal physical process would have (perhaps complicated) correlation structure among them. But intuitively, as argued in [211], this structure becomes much simplified once the true value is known; it can mainly be attributed to the residual variability and instrument precision with a simpler description. The true system response itself is already separately formulated in terms of the simulator prediction and a model bias term (Eq. (5.2)). As such, any possible complicated structure of the error (either bias or correlation) is already assigned to the model bias formulation and assuming a simpler measurement error model (i.e., independent) is sufficient [211]. In any case, as noted in [83, 93], it will be difficult to distinguish two correlation structures separately for the model bias term and observation error based on the data alone.

*Conditional
independence*

The particular distribution of the observation error is often assumed to be a Gaussian in the applied literature [83, 93, 209, 211, 212],

$$\mathcal{E} \sim \mathcal{N}(0, \sigma_{\text{obs}}^2(\boldsymbol{\lambda})) \quad (5.9)$$

*Gaussian
observation error*

or equivalently following the conditional independence assumption explained above,

$$(y_E | y_T = y_T(\mathbf{x}_c, \boldsymbol{\lambda})) \sim \mathcal{N}(y_T(\mathbf{x}_c, \boldsymbol{\lambda}), \sigma_{\text{obs}}^2(\boldsymbol{\lambda})) \quad (5.10)$$

where σ_{obs}^2 is the variance of the Gaussian distribution and is the only hyper-parameter of this observation error specification. The value of

the variance depends on the element of the observation layout λ . Eq. (5.9) implies that the observation is taken without bias and the error is independent (but need not be identically distributed) Gaussian random variable.

5.2.3 Probabilistic Model for the Simulator

For a deterministic simulator y_M , the probabilistic modeling of the bias term δ and the observation error term ϵ are enough to formulate a probabilistic model for the experimental observation \mathcal{Y}_E . However, following the development taken in Chapter 4, a Gaussian process (GP) can also be used to represent a deterministic simulator using an explicit formulation of a stochastic process. The prediction made by the simulator at particular values of \mathbf{x}_c , $\hat{\mathbf{x}}_m$, and λ is then given by,

$$\mathcal{Y}_M(\mathbf{x}_c, \hat{\mathbf{x}}_m, \lambda) \sim \mathcal{N}(m(\mathbf{x}_c, \hat{\mathbf{x}}_m, \lambda; \boldsymbol{\psi}_m), s^2(\mathbf{x}_c, \hat{\mathbf{x}}_m, \lambda; \boldsymbol{\psi}_m)) \quad (5.11)$$

where m and s^2 is the kriging mean and the kriging variance, respectively (see Section 4.3); and $\boldsymbol{\psi}_m$ is the hyper-parameters associated with the specification of the GP (e.g., its covariance kernel).

This step is taken especially if the simulator is computationally expensive to evaluate and only a limited number of simulator runs can be afforded [83, 92, 93]. The probabilistic model in Eq. (5.11) then becomes an approximation to the actual simulator (i.e., a GP metamodel). Furthermore, as explained in Chapter 4, the uncertainty associated with a prediction by the metamodel at an arbitrary input point stems from the fact that the simulator itself was not run at that input. This prediction is based on the outputs of which the simulator was run (i.e., the training data)³.

5.2.4 Posterior of the Model Parameters

Data generating process, general

Summarizing the above discussions for a deterministic simulator y_M ,

$$\begin{aligned} \mathcal{Y}_M &\equiv \mathcal{Y}_M \sim p(\mathcal{Y}_M | \hat{\mathbf{x}}_m, \mathbf{x}_c, \lambda) = \delta_d(\mathcal{Y}_M - y_M(\hat{\mathbf{x}}_m, \mathbf{x}_c, \lambda)) \\ (\mathcal{Y}_T - \mathcal{Y}_M) &\equiv \mathcal{D}(\mathbf{x}_c, \lambda) \sim p(\delta | \boldsymbol{\psi}_\delta, \mathbf{x}_c, \lambda) \\ (\mathcal{Y}_E - \mathcal{Y}_T) &\equiv \mathcal{E}(\lambda) \sim p(\epsilon | \boldsymbol{\psi}_\epsilon, \lambda) \end{aligned} \quad (5.12)$$

where δ_d is the Dirac delta function indicating that the simulator prediction is exact (i.e., a *degenerate* density).

Suppose that the form of the densities in Eq. (5.12) are already given, then the stochastic process \mathcal{Y}_E is obtained by adding the terms on the right hand side of Eq. (5.2). Assuming that they are indepen-

³ The statement “conditional on the training data” in Eq. (5.11), i.e., $\mathcal{Y}_M(\mathbf{x}_c, \hat{\mathbf{x}}_m; \lambda) | \mathcal{Y}(\mathbf{DM})$ has been implicitly assumed.

dent, the PDF of y_E is defined as the convolution of the terms [213],

$$p(y_E | \Psi_\delta, \Psi_\epsilon, \hat{\mathbf{x}}_m, \mathbf{x}_c, \boldsymbol{\lambda}) = \dots \\ (p(y_M(\hat{\mathbf{x}}_m, \mathbf{x}_c, \boldsymbol{\lambda})) * p(\delta | \Psi_\delta, \mathbf{x}_c, \boldsymbol{\lambda}) * p(\epsilon | \Psi_\epsilon, \boldsymbol{\lambda}))(y_E) \quad (5.13)$$

where $*$ is the symbol for the convolution operation. This formulation implies that the deterministic simulator is embedded into the probabilistic model y_E .

Following the Gaussian distribution formulations for the model bias, the observation error, and the simulator approximation, a normal likelihood for the calibration problem can be obtained as follows,

Data generating process, Gaussian

$$y_E = y_M + \mathcal{D} + \mathcal{E} \\ y_M(\mathbf{x}_c, \hat{\mathbf{x}}_m, \boldsymbol{\lambda}) \sim \mathcal{N}(m_M(\mathbf{x}_c, \hat{\mathbf{x}}_m, \boldsymbol{\lambda}; \Psi_m), s_M^2(\mathbf{x}_c, \hat{\mathbf{x}}_m, \boldsymbol{\lambda}; \Psi_m)) \\ \mathcal{D}(\mathbf{x}_c; \boldsymbol{\lambda}) \sim \mathcal{N}(m_\delta(\mathbf{x}_c, \boldsymbol{\lambda}; \Psi_\delta), s_\delta^2(\mathbf{x}_c, \boldsymbol{\lambda}; \Psi_\delta)) \\ \mathcal{E}(\boldsymbol{\lambda}) \sim \mathcal{N}(0, \sigma_{obs}^2(\boldsymbol{\lambda})) \quad (5.14)$$

As such, the data generating process y_E under the Gaussian formulation above is

$$y_E \sim \mathcal{N}(m_*, s_*^2) \\ m_*(\mathbf{x}_c, \hat{\mathbf{x}}_m, \boldsymbol{\lambda}; \Psi_m, \Psi_\delta) = m_M(\mathbf{x}_c, \hat{\mathbf{x}}_m, \boldsymbol{\lambda}; \Psi_m) + m_\delta(\mathbf{x}_c, \boldsymbol{\lambda}; \Psi_\delta) \\ s_*^2(\mathbf{x}_c, \hat{\mathbf{x}}_m, \boldsymbol{\lambda}; \Psi_m, \Psi_\delta, \sigma_{obs}^2) = s_M^2(\mathbf{x}_c, \hat{\mathbf{x}}_m, \boldsymbol{\lambda}; \Psi_m) + s_\delta^2(\mathbf{x}_c, \boldsymbol{\lambda}; \Psi_\delta) + \sigma_{obs}^2(\boldsymbol{\lambda}) \quad (5.15)$$

where m_* and s_*^2 are the mean and the standard deviation of the experimental data generating process under the Gaussian formulation, respectively.

Given a set of experimental data \mathbf{y} taken at \mathbf{x}_c and observed on an observation layout $\boldsymbol{\Lambda}$, the likelihood function is then defined as follows

Likelihood function

$$\mathcal{L}(\hat{\mathbf{x}}_m, \Psi_\delta, \Psi_\epsilon; \mathbf{y}, \mathbf{x}_c, \boldsymbol{\Lambda}) \equiv p(y_E = \mathbf{y} | \mathbf{x}_c = \mathbf{x}_c, \hat{\mathbf{x}}_m, \Psi_\delta, \Psi_\epsilon, \boldsymbol{\Lambda}) \quad (5.16)$$

Under the Gaussian formulation, the likelihood function is obtained by using the Gaussian density of Eq. (5.15) for p . Note that if the set of experimental data is simultaneously given on the observation layout $\boldsymbol{\Lambda}$ then the covariance matrix Σ_* is used instead of the standard deviation s_*^2 ,

$$\Sigma_*(\mathbf{x}_c, \hat{\mathbf{x}}_m, \boldsymbol{\Lambda}; \Psi_m, \Psi_\delta, \Psi_\epsilon) = \Sigma_M(\mathbf{x}_c, \hat{\mathbf{x}}_m, \boldsymbol{\Lambda}; \Psi_m) + \dots \\ \Sigma_\delta(\mathbf{x}_c, \boldsymbol{\Lambda}; \Psi_\delta) + \Sigma_{obs}(\boldsymbol{\Lambda}; \Psi_\epsilon) \quad (5.17)$$

where Σ_M , Σ_δ , and Σ_{obs} are the $P \times P$ covariance matrices of the simulator approximation, the model bias term, and the observation error, with P the dimension of the experimental data.

*Joint posterior
density*

According to the Bayes' theorem, the joint posterior probability of the model parameters \mathbf{x}_m and the hyper-parameters associated with the model bias and the observation error is given as,

$$p(\hat{\mathbf{x}}_m, \boldsymbol{\psi}_\delta, \boldsymbol{\psi}_{\epsilon_y} | \mathbf{y}, \mathbf{x}_c, \boldsymbol{\Lambda}) = \dots \frac{\mathcal{L}(\hat{\mathbf{x}}_m, \boldsymbol{\psi}_\delta, \boldsymbol{\psi}_{\epsilon_y}; \mathbf{y}, \mathbf{x}_c, \boldsymbol{\Lambda}) \cdot p(\hat{\mathbf{x}}_m) \cdot p(\boldsymbol{\psi}_\delta | \boldsymbol{\Lambda}) \cdot p(\boldsymbol{\psi}_{\epsilon_y} | \boldsymbol{\Lambda})}{p(\mathbf{y}_E = \mathbf{y} | \mathbf{x}_c = \mathbf{x}_c, \boldsymbol{\Lambda})} \quad (5.18)$$

where $p(\hat{\mathbf{x}}_m)$, $p(\boldsymbol{\psi}_\delta; \boldsymbol{\Lambda})$, and $p(\boldsymbol{\psi}_{\epsilon_y}; \boldsymbol{\Lambda})$ are the prior probabilities for the model parameters, the model bias hyper-parameters, and the observation error parameters, respectively.

*Normalizing
constant*

The denominator of the Eq. (5.18) is a normalizing constant with respect to the model parameters and the hyper-parameters such that Eq. (5.18) is a valid probability density (i.e., integration over the domain yields the value 1.0). As such, it is defined as a multidimensional integral of the following,

$$p(\mathbf{y}_E = \mathbf{y} | \mathbf{x}_c = \mathbf{x}_c; \boldsymbol{\Lambda}) = \int \mathcal{L}(\hat{\mathbf{x}}_m, \boldsymbol{\psi}_\delta, \boldsymbol{\psi}_\epsilon; \mathbf{y}, \mathbf{x}_c, \boldsymbol{\Lambda}) \cdot \dots \cdot p(\hat{\mathbf{x}}_m) \cdot p(\boldsymbol{\psi}_\epsilon | \boldsymbol{\Lambda}) \cdot p(\boldsymbol{\psi}_\delta | \boldsymbol{\Lambda}) d\hat{\mathbf{x}}_m d\boldsymbol{\psi}_\epsilon d\boldsymbol{\psi}_\delta \quad (5.19)$$

The specifications of the likelihood and the associated priors completely specify the Bayesian statistical calibration framework for the model parameters. The full Bayesian formulation for the model parameters calibration given in Eq. (5.18) involves numerous parameters. Besides the model parameters and controllable inputs, the complete formulation above also involves additional parameters associated with the statistical models: $\boldsymbol{\psi}_\delta$, $\boldsymbol{\psi}_{\text{obs}}$, and $\boldsymbol{\psi}_m$ the (hyper-)parameters for the model bias, the observation error, and the simulator approximation, respectively. By simultaneously calibrating them against experimental data, the uncertainties due to the specifications of the model bias, the observation error, and the simulator approximation up to their (hyper-)parametrization are taken into account. In principle, the hyper-parameters are now also part of the calibration problem, increasing the size (in dimension) and the complexity of it. Before presenting a simulation method to make inference on the model parameters, a modularized approach [186] to simplify the problem is introduced beforehand in the following, assuming the Gaussian formulation.

5.2.5 Modularization of the Bayesian Framework

*Modularization,
motivation*

The formulation presented above is naturally compartmentalized into three distinct modules: the GP metamodels for the simulator approximation and the model bias term, and a multivariate Gaussian distribution for the observation error. In a modularized approach, instead of simultaneously calibrating all the parameters involved with

experimental data, the hyper-parameters associated with each of the modules are separately estimated and then fixed in the downstream analysis. The main reason for this simplification is the concern that the parameters involved might be poorly identifiable with respect to the experimental data, especially between parameters (and hyper-parameters) in the different modules. This, in turn, causes difficulty in making inference about the model parameters [186]. Moreover, there is a computational incentive in limiting the number of hyper-parameters in the calibration. Though the simulation method of Section 5.3 tends to have less severe dependence on the problem dimension, more parameters often also increases the complexity of the problem and causes the method to converge slowly. A lack of identifiability is an example of such an increase in complexity not present in the calibration problem of a lower dimension.

Consider first the modularization of the metamodel for the simulator approximation. It is natural to consider the outputs of actual simulator runs (and *not* the experimental data) to be the basis of metamodel construction. Moreover, experimental data tends to be scarce, while the simulator runs would be relatively easier to generate across the range of inputs. Indeed, this was what was done in Chapter 4, where the training of the metamodel was separated from the calibration of the model parameters. The training step, where the hyper-parameters associated with the metamodel were estimated using Maximum Likelihood Estimation (MLE), was done only on the basis of simulator runs (i.e., the training data). Afterward, the metamodel was required to accurately predict the simulator outputs for arbitrary inputs (both model parameters and controllable inputs) within a predefined range via an independent validation step using additional simulator runs. The estimated hyper-parameters of the GP metamodel was kept constant during the validation step and the performance of the metamodel was not assessed with respect to the experimental data.

*Modularization of
the metamodel*

Modularizing the model bias term is more intricate due to inherent confounding between this term and the unknown best model parameters. Model bias is defined as the difference between true response value and the simulator prediction using best, but unknown model parameters (Eq. (5.2)). The model bias itself is unknown (uncertain) a priori. As such, without further information, experimental data alone cannot be used to distinguish between the model bias term and the model parameters. When GP is used to represent the uncertain model bias, this problem typically becomes worse as it introduces multiple hyper-parameters associated with the GP specification.

*Modularization of
the model bias term*

Due to this confounding, modularizing the model bias term is commonly carried out such that the hyper-parameters associated with the GP of the bias term can be fixed prior to the calibration of the model parameters. There are no consensus in the literature on how exactly such fixed values of the hyper-parameters are obtained. How-

ever, Refs. [83, 92, 93, 214] provide a common theme in the modularization of the model bias term through MLE of the hyper-parameters based on an initial fitting the difference between the simulator prediction (evaluated using selected value, or values, from the prior of the model parameters) and the experimental data. Refs. [93, 214], for instance, adopt a pragmatic approach where a GP for the model bias term is fitted (and their corresponding hyper-parameters estimated) based on the differences between simulator prediction using the prior mean of the model parameters and the experimental data across controllable inputs and on the observation layout. This provides an initial estimate of the bias. Afterward, depending on what expectation or assumption are put on the bias term, the estimated associated hyper-parameters can still be allowed to vary in the downstream analysis.

Lastly, one of the main sources for estimating the parameters of the experimental observation error model (i.e., the variance under Gaussian formulation) is the replications under the same experimental condition. If those are not available then alternative sources must be consulted. For instance, experimentalist would have an idea on the extent of the observation error and often, these figures can be found in the experiment report. If these are not to be found and point estimate of these parameters cannot be justified, then prior distribution on the parameters should be assigned. In this case, many analysis in the applied literature assume prior distribution with different degree of informativeness for the scale parameter (e.g., exponential, half-Cauchy, inverse-Gamma, etc.) [88, 93, 209].

The modularization approach represents a series of compromises between having a full uncertainty analysis and having a more tractable formulation of the model parameters calibration problem [83]. Such compromises then become part of the modeling decision in order to simplify a particular calibration problem: fixing the hyper-parameters associated with the GP metamodel Ψ_m at *estimated* values implies that the uncertainty in the simulator approximation is not taken into account completely; fixing the hyper-parameters associated with the GP model for the bias term Ψ_δ at *estimated* values implies that the uncertainty in the bias is not taken into account completely; and fixing the parameters associated with the Gaussian distribution of the observation error Ψ_ϵ at *estimated* values implies that the uncertainty in the observation error is not taken into account completely.

Meanwhile, completely taking into account all sources of uncertainty up to their level of hyper-parameters (and parameters for the observation error model), at the cost of increasing model complexity, might not be necessary. Indeed, as it was reported in Refs. [83, 92, 93, 186, 205], the effect of the additional sources of uncertainty at that level were relatively minor on the calibration results. This is due to the fact that the variation in the simulator prediction is largely determined by the uncertainty about the model parameters and the control-

*Modularization of
the observation error
model*

*Modularization, a
compromise*

*Modularization,
justification*

lable inputs rather than the uncertainty about the hyper-parameters. Hence, it is more important in the simplified analysis to recognize what is being compromised above and to recognize properly the sources of uncertainty in the calibration at the level of metamodel (i.e., simulator approximation), model bias, and observation error in the first place.

5.3 MARKOV CHAIN MONTE CARLO (MCMC) SIMULATION

The formulation of the Bayesian calibration of a computer model presented above results in a joint posterior PDF for all the parameters involved in the resulting probability model $p(\mathbf{x}|\mathbf{y})$ (Eq. (5.18) in Section 5.2.4). This density contains all the information (and consequently, the uncertainties) regarding the model parameters conditioned on the observed data and the assumed data-generating process. The uncertainties associated with the model parameters can then be represented using different summary statistics, many of which involve integration.

*Posterior
uncertainty of the
model parameters*

For example, the uncertainties associated with a model parameter x_d can be represented by its variance, which is defined as

$$\begin{aligned} \mathbb{V}[x_d] &\equiv \mathbb{E}[x_d^2] - \mathbb{E}^2[x_d] \\ &= \int_{\mathbf{X}} x_d^2 p(x_d, \mathbf{x}_{\sim d} | \mathbf{y}) \, d\mathbf{x} - \left(\int_{\mathbf{X}} x_d p(x_d, \mathbf{x}_{\sim d} | \mathbf{y}) \, d\mathbf{x} \right)^2 \end{aligned}$$

where $\mathbf{x} = \{x_d, \mathbf{x}_{\sim d}\}$ stands for the the set of model parameters \mathbf{x}_m in Sections 5.1 and 5.2; and the integrations are carried out over the domain \mathbf{X} . An alternative way to summarize the uncertainties of a model parameter is through its θ -quantile Q_d^θ , which for parameter x_d is defined as

$$Q_d^\theta : \mathbb{P}(x_d \leq Q_d^\theta) \equiv \int_{\inf \mathbf{X}_d}^{Q_d^\theta} \int_{\mathbf{x}_{\sim d}} p(x_d, \mathbf{x}_{\sim d} | \mathbf{y}) \, d\mathbf{x}_{\sim d} \, dx_d = \theta$$

In this manner, the 95% confidence interval of the parameter is written as $Q_d^{0.025} \leq x_d \leq Q_d^{0.975}$.

Though these summaries might be of interest, in an application setting, the model parameters uncertainties are often propagated through the simulation model to obtain the uncertainty in the prediction. Hence, the output from a simulation model $y = f(\mathbf{x})$ is expressed as a random variable \mathcal{Y} from the transformation of a random variable $\mathcal{X} | \mathbf{y}$ by the function f

*Posterior
uncertainty of the
model prediction*

$$\mathcal{Y} = f(\mathcal{X} | \mathbf{y}); p_{\mathcal{X}|\mathbf{y}}(\mathbf{x}) = p(\mathbf{x} | \mathbf{y})$$

where the PDF of $\mathcal{X} | \mathbf{y}$ is the posterior density $p(\mathbf{x} | \mathbf{y})$. The actual PDF of \mathcal{Y} follows the rule of transformation of random variable and it

represents the uncertainty in the output due to the uncertainty in the input parameters conditioned on the data. This uncertainty can also be summarized with various statistics and, as before, many of these involve integration. For instance, the variance of the output:

$$\mathbb{V}[y] = \int_{-\infty}^{\infty} f^2(\mathbf{x})p(\mathbf{x}|\mathbf{y})d\mathbf{x} - \left(\int_{-\infty}^{\infty} f(\mathbf{x})p(\mathbf{x}|\mathbf{y})d\mathbf{x} \right)^2$$

Challenges in dealing with posterior density

The posterior density $p(\mathbf{x}|\mathbf{y})$ and the function $f(\mathbf{x})$, however, are in practice highly multidimensional functions and performing their integration numerically is harder with an increasing number of input parameters. At the same time, conducting MC simulation for estimating the integrals (as was done in Chapter 3 in the estimation of the Sobol' indices) is not straightforward in this case. The multiplication of likelihood $p(\mathbf{y}|\mathbf{x})$ and prior density $p(\mathbf{x})$ will, in general, yield an arbitrary posterior density not available in a closed-form expression ready to be sampled from. As a result, generating independent samples from the posterior density required for the MC estimation becomes a difficult task.

This section presents an approach, the so-called MCMC simulation, to directly generate samples from an arbitrary PDF. These samples are useful for estimating various quantities given as examples above. This technique works with less severe dependence on the dimension of the input parameter space.

Although in the context of Bayesian data analysis the PDF of interest is the posterior PDF [215], generating samples from an arbitrary PDF is a general problem. As such, in the following, a generic notation for an arbitrary PDF $p(\mathbf{x})$ is used instead of $p(\mathbf{x}|\mathbf{y})$.

5.3.1 Motivation

Problem statement

Consider the following problem: Generate a set of samples $\{\mathbf{x}_n\}_{n=1}^N$ from a random variable \mathcal{X} , given the PDF $p : \mathbf{X} \subseteq \mathbb{R}^D \mapsto \mathbb{R}_{\geq 0}$. It is assumed that the PDF can be evaluated at any given $\mathbf{x} \in \mathbf{X}$, at least up to a proportionality constant. That is,

$$p^*(\mathbf{x}) = C p(\mathbf{x}) \tag{5.20}$$

The proportionality constant in the above equation is the normalizing constant such that p is a valid PDF, $\int p(\mathbf{x})d\mathbf{x} = 1 \Leftrightarrow C = \int p^*(\mathbf{x})d\mathbf{x}$. Carrying out the integration of p^* can be problematic in its own right especially for a highly multidimensional function (such as the normalizing constant of a posterior density given in Eq. (5.19)). Therefore, generating a set of samples simply by knowing p^* instead of p is advantageous.

The generated samples can then be used, among other things, to evaluate different summary statistics (such as expectation, variance, etc.) of \mathbf{x} itself or of any function under the PDF.

In the rest of the section, the term model input parameter space \mathbf{X} is replaced by the term *state space*, the range of possible values of the random variable \mathcal{X} ; a more appropriate term in the context of generic problem of generating samples from a distribution.

Generating samples from an arbitrary multidimensional density function is generally a difficult task. Intuitively, for a given sample size, correctly generating samples from a density means that the sample values have to be distributed proportionally to its PDF. There should be more samples in the region where the PDF value is high, and less in the the region where the PDF value is low. For a complex multidimensional density function, these locations are not known a priori and might have to be identified exhaustively [98].

A correct sampling

In one dimension, the most common way of generating sample from a given density is by inverse transform sampling coupled with a random number generator. The approach requires the quantile function of the PDF. To obtain the quantile function, the density has to be integrated and its normalizing constant has to be computed. Appendix D.4 provides a more detail account on the topic. Many univariate random variables are widely studied and the analytical solutions to their quantile functions are available [216]. However, the method is not readily extendable to distributions of higher dimension. Additionally, though sampling algorithms exist for several multivariate densities (notably, the multivariate normal density – See Appendix D.5), this will not be the case for an arbitrary PDF of higher dimension.

Inverse transform sampling

To illustrate the difficulty to generate samples from an arbitrary multivariate probability distribution, consider the following unnormalized bivariate PDF parametrized by the location parameters μ_1, μ_2 and the scale parameters σ_1, σ_2 [217]:

Illustration

$$p^*(x_1, x_2) = \frac{\exp(-(x_1 - \mu_1)/\sigma_1) \exp(-(x_2 - \mu_2)/\sigma_2)}{(1 + \exp(-(x_1 - \mu_1)/\sigma_1) + \exp(-(x_2 - \mu_2)/\sigma_2))^3}$$

$$x_1, x_2 \in \mathbb{R}; \mu_1, \mu_2 \in \mathbb{R}; \text{ and } \sigma_1, \sigma_2 \in \mathbb{R}^+$$

(5.21)

Fig. 5.3 shows the contour plot of the joint density as well as the marginal density for each of the variate, for selected range of values of its parameters.

A straightforward approach to generate samples from a given multivariate density is done by first discretizing the state space of the density function and evaluate the density at the discretized points. Supposed the domain of the density has been discretized uniformly in each dimension with a level Δ resulting in $\{(x_{1,i}, x_{2,j}); i, j = 1, \dots, I\}$ with I the number of discretized points. At the discretized levels, the probability for each value of $(x_{1,i}, x_{2,j})$ is approximated by $p(x_{1,i}, x_{2,j}) = p^*(x_{1,i}, x_{2,j}) / \sum_{i,j} p^*(x_{1,i}, x_{2,j})$ ⁴. The set $\{p(x_{1,i}, x_{2,j}); i, j = 1, \dots, I\}$

Discretized grid approach

⁴ strictly speaking, each density value has to be multiplied by the hypervolume of the grid to obtain the probability mass, but the term cancels out in computing p .

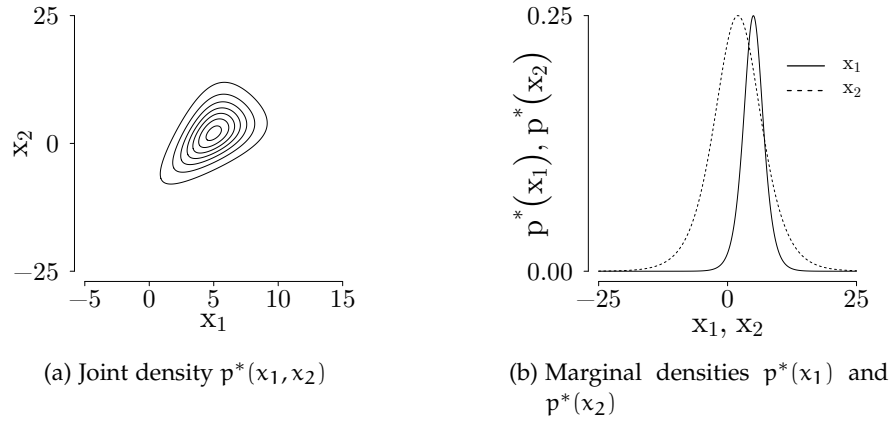


Figure 5.3: Joint and marginal densities plots of the unnormalized PDF in the example. The parameters used in the example are: $\mu_1 = 5$, $\mu_2 = 2$, $\sigma_1 = 1.25$, and $\sigma_2 = 3$.

constitute a complete discrete probability distributions. Generating samples from such a probability distribution is straightforward in a modern computing environment [98].

Discretized grid approach illustrated

Fig. 5.4 illustrates this procedure for the example given above. First, the state is windowed in $\mathbf{X} \in [-25, 25]^2$ before being discretized in $\Delta = 50$ levels. This results in $2'601$ discretized points at which the density is evaluated (Fig. 5.4a). Next, the density values are taken to be the probability for each of the $2'601$ discretized points. Together they make up a complete discrete probability distribution from which samples can be readily generated.

Fig. 5.4a shows 5'000 samples generated from the discrete distribution. Darker points indicate that the values have been sampled multiple times following the actual underlying PDF (the contour of the analytical joint density is overlaid). Figs. 5.4b and 5.4c show the histograms for each of the marginals. The figure shows that the generated samples are indeed approximately distributed as the given PDF.

Curse of dimensionality

The main issue with the discretized grid approach, conceptually simple as it is, is the curse of dimensionality similar to the one mentioned in the previous chapters. The number of density evaluations grows exponentially with the number of dimension. As a rule, for a given discretization level Δ and a given dimension D , the number of density evaluations is $(\Delta + 1)^D$.

Moreover, many of the evaluations on the grid exemplified above are potentially wasteful for carrying out an integration over the density, especially if the cost for evaluating the density is non-negligible. Assuming a well-behaved function of interest inside the integral, some regions of the state space will contribute more to the integration than the others. In fact, this “region of space where it matters” is related

Integration over a density, typical set

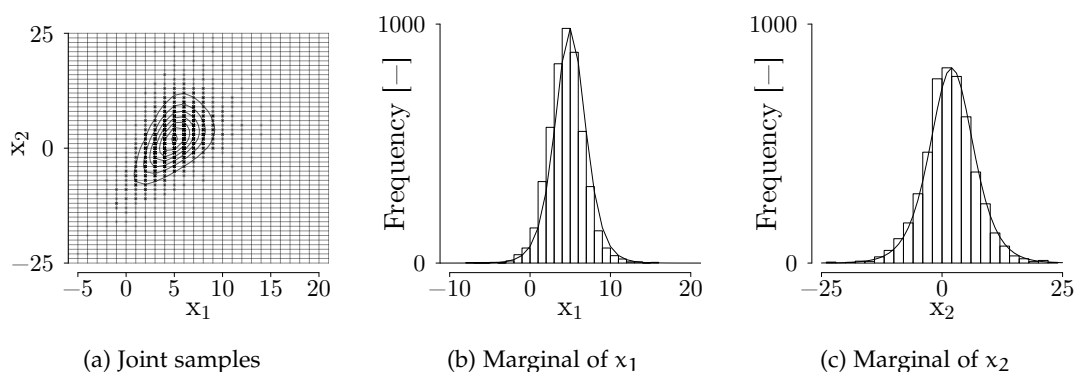


Figure 5.4: Sampling from a multivariate density by discretizing the state space in grids. Each dimension of the state space is discretized into $\Delta = 50$ levels. The density is then evaluated at the discretized points. (Left) 5'000 samples are generated following the resulting discrete probability distribution; (Center and Right) The histograms of the marginals approximately follow the shape of the respective analytical marginal density. The marginal densities have been normalized to match the peak of the histogram.

to the mathematical notion of the *typical set* of a distribution [98]. Loosely speaking, it can be thought of as the region of state space where the probability mass (density times volume) is concentrated. Consequently, any integration over the whole state space of a density can be approximated by an integration over this typical set [98]. Although the location of the typical set for the example above is rather trivial (the region around the center of the density), it will not be the case for an arbitrary high-dimensional PDF (such as a posterior density)⁵. Having samples that are representative of a typical set is a particularly challenging task in conducting MC integration over an arbitrary high-dimensional PDF⁶. At the same time, if such samples can be obtained then the performance of MC integration in high dimension would potentially be more efficient, requiring less function evaluations for the same accuracy level, than that of the grid approach⁷.

To circumvent these issues, a sampling technique based on the theory of stochastic process is adopted [215, 218]. Specifically, by constructing a Markov chain of the input parameters values, the resulting process will eventually converge to a stationary distribution which coincides with the distribution according to the given density (i.e., *target density*). Instead of blindly evaluating the density at every cor-

*Markov Chain
Monte Carlo*

⁵ And in fact, the region around the mode of a distribution becomes less representative of the typical set in high-dimension as it becomes smaller in comparison to other part of the state space where the density value, albeit small, is non-negligible. This is yet another example of the curse of dimensionality.

⁶ Recall that in Chapter 3, the MC integrations for Sobol' indices estimation were conducted over a uniform density thus it was only the property of the function of interest that mattered.

⁷ Deterministic numerical integration is either based on or and improvement of the discretized grid approach.

ner of the state space, such a Markov chain will be directed to explore the typical set of the distribution. As such, this family of techniques potentially has less severe dependence to the dimension of the state space. Generating such samples for the purpose of Monte Carlo (MC) simulation by simulating a Markov chain is termed Markov Chain Monte Carlo (MCMC).

The following briefly presents the basics of Markov chain and its importance in solving the problem of generating samples from an arbitrary PDF. Markov chain in continuous state space as well as some important related concepts and theorems are first presented, without proof and in a somewhat lax manner. A more precise statements of these concepts and theorems are difficult without measure theory which for the purpose of this thesis is irrelevant. Appendix D.8 provides the definitions and illustrations of some of these concepts for a discrete state Markov chain where more intuitive matrix notation and graphical representation are applicable. Finally, two methods to construct a Markov chain for the purpose of MC simulation are introduced and illustrated.

5.3.2 Markov Chain

*Discrete-time
stochastic process*

A Markov chain is a *discrete-time* stochastic process. Recall that from Chapter 4, a stochastic process is a collection of random variables $\{\mathcal{X}^{(i)}; i \in I\}$ where I is an index set. The term *discrete-time* refers to the fact that the possible values of the index set I are discrete. The term *time* is used by convention but by no means exclusively referred to the physical time. In this thesis, a more fitting alternative term would be *step* or *iteration*.

Markov chain

Specifically, a *continuous-state* Markov chain on state space $\mathbf{X} \subseteq \mathbb{R}^D$, D being the dimension of the state space, is defined as a sequence of random variables $\{\mathcal{X}^{(i)}; i \geq 0\}$ where the indices represents successive time, steps, or iterations, *such that the conditional probability of $\mathcal{X}^{(i)}$ given the previous iterations follows the Markov assumption*. That is,

$$\begin{aligned} \mathbb{P}(\mathcal{X}^{(i+1)} \in \mathbf{A} | \mathcal{X}^{(i)} = \mathbf{x}^{(i)}, \dots, \mathcal{X}^{(0)} = \mathbf{x}^{(0)}) = \dots \\ \mathbb{P}(\mathcal{X}^{(i+1)} \in \mathbf{A} | \mathcal{X}^{(i)} = \mathbf{x}^{(i)}), \quad \mathbf{A} \subseteq \mathbf{X} \end{aligned} \quad (5.22)$$

Put differently, the future value depends on the preceding values only through the present one [105, 219].

A Markov chain is fully specified by three components:

- The *state space* $\mathbf{X} \subseteq \mathbb{R}^D$, the set of values which can be taken by the random variables $\mathcal{X}^{(i)}$.
- The *initial distribution* of $\mathcal{X}^{(0)}$, given by the density $\pi^{(0)}(\mathbf{x})$.
- The *transition probability kernel (density)* $T(\mathbf{x}, \mathbf{x}')$ a function $T : \mathbf{X} \times \mathbf{X} \mapsto \mathbb{R}$. For any given \mathbf{x} , $T(\mathbf{x}, \circ)$ defines the conditional

probability density of $\mathcal{X}^{(i)}$ given $\mathcal{X}^{(i-1)} = \mathbf{x}$ with the following properties:

$$\begin{aligned} T(\mathbf{x}, \mathbf{x}') &\geq 0, \quad \forall \mathbf{x}, \mathbf{x}' \in \mathbf{X} \\ \int_{\mathbf{X}} T(\mathbf{x}, \mathbf{x}') \, d\mathbf{x}' &= 1, \quad \forall \mathbf{x} \in \mathbf{X} \end{aligned} \quad (5.23)$$

As $T(\mathbf{x}, \circ)$ is a PDF, then for any given \mathbf{x} it also follows that,

$$\mathbb{P}(\mathcal{X}^{(i)} \in \mathbf{A} \mid \mathcal{X}^{(i-1)} = \mathbf{x}) = \int_{\mathbf{A}} T(\mathbf{x}, \mathbf{x}') \, d\mathbf{x}'; \quad \mathbf{A} \subseteq \mathbf{X} \quad (5.24)$$

The distribution of $\mathcal{X}^{(i)}$ due to the transition from the previous iteration $\mathcal{X}^{(i-1)}$, is given by the transition probability kernel T operated on the density function of the previous iteration $\pi^{(i-1)}$ such that,

$$\pi^{(i)}(\mathbf{x}') = \int_{\mathbf{X}} \pi^{(i-1)}(\mathbf{x}) T(\mathbf{x}, \mathbf{x}') \, d\mathbf{x} \equiv (\pi^{(i-1)}T)(\mathbf{x}') \quad (5.25)$$

where $\pi^{(i)}(\mathbf{x}')$ is the PDF of $\mathcal{X}^{(i)}$. The rightmost definition signifies that the integration of the density function with the transition kernel is taken as an operator on the density function $\pi^{(i-1)}$ resulting in $\pi^{(i)}$ [220, 221]. As such, given the initial distribution of the chain $\mathcal{X}^{(0)}$ and the transition kernel T , the distributions of all the other Markov chain iterations are determined by repeating the integration for each successive iterations. This, in terms of the transition operator, is

$$\begin{aligned} \pi^{(i)}(\mathbf{x}) &= (\pi^{(0)}T^i)(\mathbf{x}) \\ &\equiv \int_{\mathbf{X}} \dots \int_{\mathbf{X}} \pi^{(0)}(\mathbf{x}^{(0)}) T(\mathbf{x}^{(0)}, \mathbf{x}^{(1)}) \dots T(\mathbf{x}^{(i-1)}, \mathbf{x}) \, d\mathbf{x}^{(0)} \dots d\mathbf{x}^{(i-1)} \end{aligned} \quad (5.26)$$

A density π^* is said to be *stationary* with respect to a transition kernel T if the density is invariant under transition. Specifically,

$$(\pi^*T)(\mathbf{x}) = \pi^*(\mathbf{x}) \quad (5.27)$$

Stationary density

In other words, once the chain reaches the stationary density, it will stay there and the chain itself becomes stationary.

The notion of stationary density of a Markov chain is central to the application of Markov chain for generating samples from an arbitrary probability density. Under certain conditions (i.e., irreducible, aperiodic, and Harris recurrent – See Appendix D.8) for the transition kernel T , a stationary density π^* exists, is unique and is the *limiting density* of the stochastic process, such that

Fundamental theorem of Markov chain

$$\lim_{i \rightarrow \infty} |(\pi T^i) - \pi^*| = 0, \quad \forall \pi \in \mathbf{D} \quad (5.28)$$

where \mathbf{D} is the set of all possible PDFs on \mathbf{X} . It implies that regardless of the starting density, some transition kernels will converge to

a unique stationary density. This in turn, is of practical importance when the transition kernel in an MCMC algorithm is designed such that the given target density is the stationary density. This statement of the existence, the uniqueness, and the convergence of a stationary density is the *fundamental theorem* of Markov chain [220, 222].

Central Limit
Theorem

Although the previous theorem provides the mathematical foundation for constructing an MCMC algorithm, it is the property of the sample path⁸ of a finite length, $\{\mathbf{x}^{(i)}\}_{i=1}^I$, that matters when conducting an actual MCMC simulation [222]. By adding additional conditions on T^9 , another theorem states that for a given Markov chain $\{\mathcal{X}^{(i)}\}_{i=1}^I$ with stationary density π^* and for any function $f : \mathbf{X} \mapsto \mathbb{R}$ the following asymptotic result holds,

$$\lim_{I \rightarrow \infty} \frac{1}{I} \sum_{i=1}^I f(\mathcal{X}^{(i)}) - \mathbb{E}_{\pi^*}[f] \sim \mathcal{N}\left(0, \frac{\sigma_f^2}{I}\right) \quad (5.29)$$

That is, the difference between the sample mean and the expected value converges in distribution to the normal distribution with variance σ_f^2/I . σ_f^2 is the variance of the function evaluated under the stationary density of the chain. The theorem implies that for a long enough chain (assuming that the stationarity has been attained), the function f integrated along the target PDF π^* and consequently the QoIs as exemplified in the opening of this section can be estimated with the sample mean of f evaluated at the points of the sample path $\{\mathbf{x}^{(i)}\}_{i=1}^I$. Finally, the theorem provides a basis for estimating the error of an estimate computed by MCMC samples of a finite size.

Though similar to that of MC standard error [223], it is important to note that successive realizations of the Markov chain are not, by construction, independent and identically distributed. The consequence of this will be revisited when the topic of analyzing samples of a Markov chain is discussed in Section 5.4.

Detailed balance
condition

In generating samples from a target density, the engineering is done somewhat in reverse: “Given a target density π_t , construct T such that its stationary density π^* converges to π_t ”. Thus it is worthwhile to note the *detailed balance* condition which is a central condition for an MCMC algorithm. A Markov chain with transition kernel density $T(\circ, \circ)$ satisfies the detailed balance condition if there exists a probability density π such that,

$$\pi(\mathbf{x}) T(\mathbf{x}, \mathbf{x}') = \pi(\mathbf{x}') T(\mathbf{x}', \mathbf{x}) \quad \forall \mathbf{x}, \mathbf{x}' \in \mathbf{X} \quad (5.30)$$

Reversible Markov
chain

As a result, the chain is said to be *reversible*. Formally, for $\mathbf{A} \subseteq \mathbf{X}$,

$$\mathbb{P}(\mathcal{X}^{(i)} \in \mathbf{A} | \mathcal{X}^{(i-1)} = \mathbf{x}) = \mathbb{P}(\mathcal{X}^{(i)} \in \mathbf{A} | \mathcal{X}^{(i+1)} = \mathbf{x}) \quad \forall \mathbf{x} \in \mathbf{X} \quad (5.31)$$

⁸ Recall that from Chapter 3, a sample path is a realization of a stochastic process and in this case, a realization of a Markov chain.

⁹ Namely, irreducible, aperiodic, Harris recurrent, *and* geometrically ergodic. Thus, the chain is called (geometrically) ergodic.

A reversible chain is a stationary chain [222]. Consequently, in an MCMC algorithm, if the transition probability T satisfies the detailed balance condition with respect to the target distribution π_t , it ensures the reversibility of the process and ultimately the stationarity of the chain. Finally, by imposing the conditions (see Footnote 9), the stationary distribution of the chain π^* converges to the target distribution π_t and the quantities of interest exemplified in the opening of this section can be estimated.

5.3.3 Markov Chain Monte Carlo

Consider once more the problem set up at the start of the section: Generate samples from a target probability distribution with a density $p(\mathbf{x})$, known up to a proportionality constant (Eq. (5.20)).

The objective revisited

By acknowledging the theorems above, the task is then to construct a Markov transition kernel such that the target density p becomes the stationary distribution of the Markov chain. Thereafter, based on such kernel, a realization of the chain is generated long enough for the limiting distribution of the chain to be reached, i.e., to obtain samples representative of the target density p . As a result, the samples generated from the realization of the chain converges, in distribution, to the target density. This, in essence, is the objective of MCMC algorithms as defined in Ref. [222].

MCMC algorithms

One might think that the task of constructing a Markov transition kernel would be difficult, especially considering the wide range of possible target distributions which might call for different classes of transition kernels. However, there exists a class of algorithms for generating Markov chains that guarantees its convergence (in distribution) to *any* target distribution as its stationary distribution¹⁰. The Metropolis-Hastings algorithm and its various extensions remains the most universal class of algorithms to generate such Markov chains [224]. The method was first applied for a statistical mechanics problem by Metropolis et al. [101]¹¹ and later generalized by Hastings [102]¹².

Metropolis-Hastings algorithm, origin

The MH algorithm prescribes two main components for constructing transition kernels of a Markov chain that converges to the target distribution: a *proposal probability density* $q(\mathbf{x}^* | \mathbf{x})$ and an *acceptance probability* α . The proposal probability density is responsible for generating a proposal transition or candidate move \mathbf{x}^* for the Markov chain at each iteration and it is in general a density conditional on the previous state. This density is chosen such that it is easier to sample and indeed it is often selected from well-known densities such as

Metropolis-Hastings algorithm, proposal density

¹⁰ see [105, 224] for more rigorous treatment on the convergence properties.

¹¹ There is apparently a controversy surrounding the attribution of the algorithm solely to Metropolis, especially when his role was claimed to be nothing more “other than providing computer time” [225].

¹² There is, to the best of the author’s knowledge, no controversy here.

the Gaussian or uniform densities. This proposal move, in turn, will be accepted with a probability,

$$\begin{aligned}\alpha &= \min\left(\frac{p(\mathbf{x}^*)}{p(\mathbf{x}^{(i-1)})} \times \frac{q(\mathbf{x}^{(i-1)}|\mathbf{x}^*)}{q(\mathbf{x}^*|\mathbf{x}^{(i-1)})}, 1.0\right) \\ &= \min\left(\frac{p^*(\mathbf{x}^*)}{p^*(\mathbf{x}^{(i-1)})} \times \frac{q(\mathbf{x}^{(i-1)}|\mathbf{x}^*)}{q(\mathbf{x}^*|\mathbf{x}^{(i-1)})}, 1.0\right)\end{aligned}\quad (5.32)$$

*Metropolis-Hastings
algorithm,
acceptance
probability*

Where $p(\mathbf{x}^*)$ and $p(\mathbf{x})$ are the values of the target density at the proposed state and the previous state, respectively; and $q(\mathbf{x}^{(i-1)}|\mathbf{x}^*)$ ($q(\mathbf{x}^*|\mathbf{x}^{(i-1)})$) is the value of the proposal density at the previous (proposed) state conditional on the value of the proposed (previous) state. Notice from the ratio, that the proportionality constant in Eq. (5.20) cancels out and only the unnormalized density p^* is required. As such, the potentially difficult multidimensional integral in Eq. (5.20) has been dispensed with by this formulation. The acceptance probability is formulated to satisfy the *detailed balance condition* (Eq. (5.30)) for any valid proposal probability distribution. This, in turn, guarantees the stationarity of the process [226].

If the proposal move is accepted it becomes the current state of the chain, otherwise the chain remains at its current state for the given iteration. To generate a Markov chain of certain length (i.e., certain number of samples), the steps are repeated multiple times until the required length of the chain is met. Algorithm 2 summarizes the steps for constructing a Markov chain with the MH algorithm.

Algorithm 2 Metropolis-Hastings Algorithm

Generate samples from $p(\mathbf{x}) \propto p^*(\mathbf{x})$ given proposal density $q(\mathbf{x}^*|\mathbf{x})$ in I iterations

Require: $I > 0$, $p^*(\mathbf{x})$, and $q(\mathbf{x}^*|\mathbf{x}^{(i-1)})$
 $\mathbf{x}^{(0)} \leftarrow \mathbf{x}_0; \forall \mathbf{x}_0 \in \mathbf{X}$
for $i = 1$ to I **do**
 sample \mathbf{x}^* from $q(\mathbf{x}|\mathbf{x}^{(i-1)})$
 $\alpha \leftarrow \min\left(\frac{p^*(\mathbf{x}^*)}{p^*(\mathbf{x}^{(i-1)})} \times \frac{q(\mathbf{x}^{(i-1)}|\mathbf{x}^*)}{q(\mathbf{x}^*|\mathbf{x}^{(i-1)})}, 1.0\right)$
 sample u from $\mathcal{U}[0, 1]$
 if $u < \alpha$ **then**
 $\mathbf{x}^{(i)} \leftarrow \mathbf{x}^*$
 else
 $\mathbf{x}^{(i)} \leftarrow \mathbf{x}^{(i-1)}$
 end if
end for

*Random walk MH
algorithm*

In the original paper of Metropolis et al., the proposal distribution was chosen to be a symmetric distribution such that $q(\mathbf{x}^*|\mathbf{x}^{(i-1)}) = q(\mathbf{x}^{(i-1)}|\mathbf{x}^*)$. As a result the terms associated with the proposal density in Eq. (5.32) cancel each other. Consequently, any proposal move that yields an “improvement” on the target density evaluation will be accepted, otherwise it will only be accepted according to its acceptance probability. This particularly simple MH algorithm results in a *random walk* Markov chain and it is termed the *random walk* MH [224].

To illustrate the application of the MH, particularly the random walk MH, for generating samples from an arbitrary target distribution, consider again the example of generating samples from the PDF given in Eq. (5.21). For this example, the proposal distribution is chosen to be a bivariate normal with a variance (the scale parameter) of 2.0 equal in both dimensions and without correlation. The initial state of the chain $x^{(0)}$ is set to be at the origin.

Random walk MH algorithm, illustrated

The first three iterations of the random walk MH algorithm are illustrated in Fig. 5.5. At the first iteration (Fig. 5.5a), a proposal move is drawn from the bivariate normal distribution (centered at the origin). The proposal move brings the state closer to the center of the target density, thus it is accepted (Fig. 5.5b). A new proposal move is generated from the bivariate normal centered at the newly accepted state. This time, because the proposal move moves farther away from the center of the target density, it is rejected. The chain remains at the current state and a new proposal move is drawn (Fig. 5.5c). Note that this kind of proposal move will not always be rejected outright but is subject to chance based on the acceptance probability.

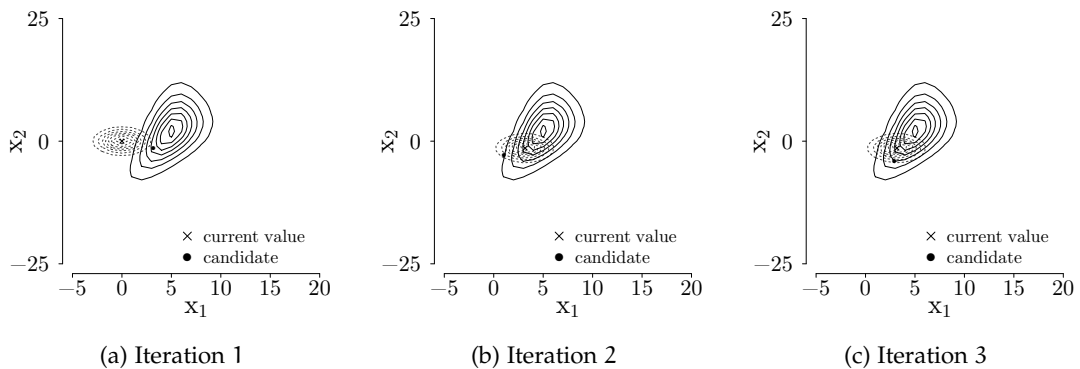


Figure 5.5: Illustration of the first three iterations in Markov Chain simulation by random walk MH algorithm to sample the density given in Eq. (5.21) whose contours showed in solid lines. The proposal density is an independent bivariate normal distribution with $\sigma^2 = 2.0$ whose contours showed in dashed lines, centered at current state.

By repeating those steps multiple times, the chain traverses the state space according to the target distribution. Fig. 5.6a illustrates the chain traversing the 2-dimensional state space of the density of Eq. (5.21) for the first 250 iterations. In the long run, the chain will spend more time in the regions where the density is high and less time where the density is low¹³. Therefore, the resulting samples generated by the chain will be distributed according to the target distribution. Figs. 5.6b and 5.6c are the 1-dimensional *trace* plots for the chain after 50'000 iterations. A trace plot shows the evolution of the chain

trace plot

¹³ The goal of an MCMC algorithm is not, on the other hand, to obtain the parameter value which maximizes the target distribution, at least not only. It seeks to explore the state space in proportion to value of the density function [215].

during the iterations and it is often the first graphical diagnostic tool to spot any possible issue of convergence of a Markov chain [224]. In this particular case, the plots show that the chain seemingly converges to particular region of the input state space and within this region (i.e., the so-called *typical set* [98]) the chain randomly moves from one state to another. It also indicates that x_1 are centered differently than x_2 , and x_2 has a relatively larger dispersion than x_1 .

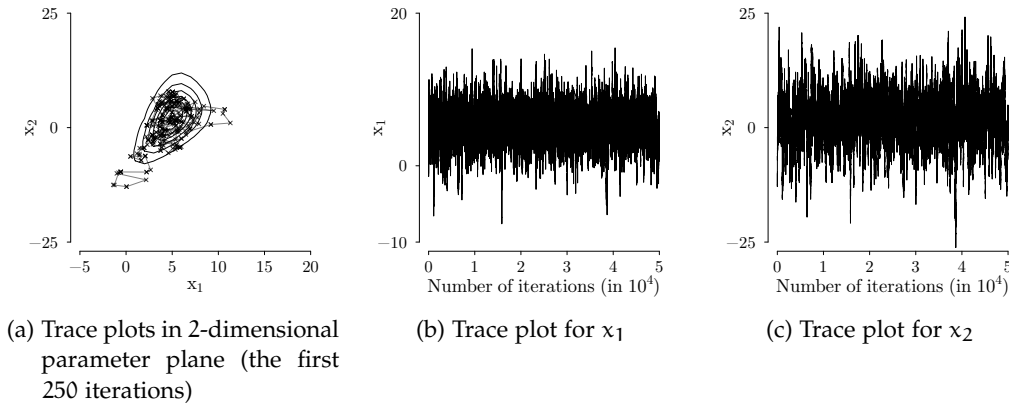


Figure 5.6: Illustration of a Markov chain simulation to generate samples from the target density of Eq. (5.21). (Left) The chain traverses the state space. At each iteration, a move is proposed and accepted in a probabilistic manner. (Center and Right) The trace plots.

After the iterations are completed, the resulting samples should be distributed according to the target distribution. Indeed, this is the case for the MCMC simulation. Fig. 5.7 shows that the resulting samples are distributed according to the target distribution both as the joint as well as its marginal. The joint distribution, in particular, shows that the samples generated by MCMC simulation are correlated according to the correlation contain in the density of Eq. (5.21). Note that, in practice, the correct distribution of the resulting samples cannot be verified by comparing it to the analytical formula. The whole point of generating samples via MCMC simulation is exactly because such arbitrary high-dimensional distributions are hard to characterize.

Although the theorems that underlie the application of MCMC algorithm guarantee the convergence of the chain to the target distribution, its rate of convergence is problem dependent. For many MH algorithms, the choice of proposal distribution is particularly important in determining the convergence rate of the algorithm in reaching the target distribution as its stationary distribution. For instance, Fig. 5.8 illustrates the case where the scale parameter of the proposal distribution is set to be much larger than the actual scale of the target distribution. As shown, because the proposal moves can jump from one side of the state space to another, they are rarely accepted and the chain sticks at the same values for a long period. For the same

*Over-dispersed
proposal
distribution.*

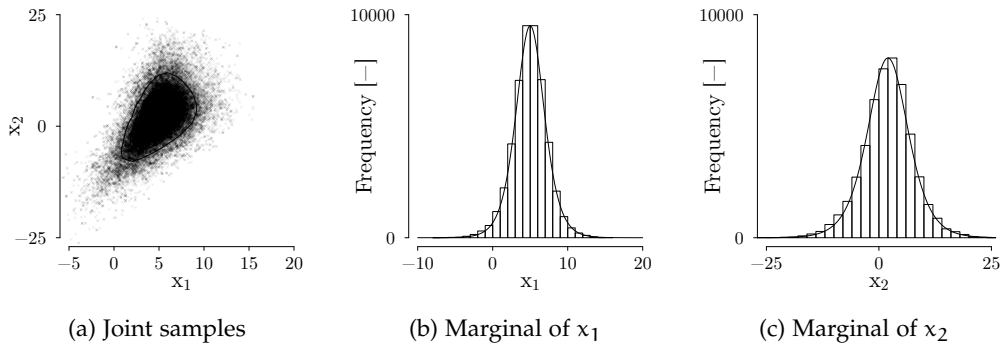


Figure 5.7: Results of samples generated by a Markov chain simulation for the target density given in the example. After 50'000 iterations, the samples resemble the actual shape of the distribution both as a joint (Left) and as marginal distributions (Center and Right). The marginal densities have been normalized to match the peak of the histogram.

length of the chain as before (50'000), the resulting distribution of the samples (Fig. 5.8b) hardly resembles the target distribution.

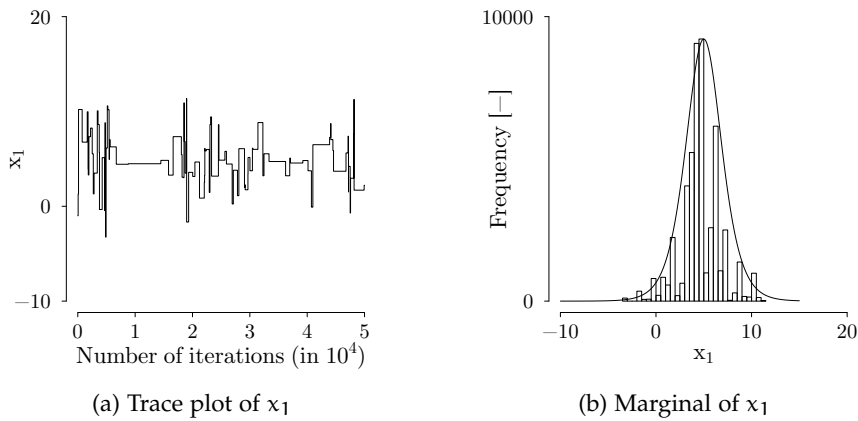


Figure 5.8: Convergence issue due to an over-dispersed proposal distribution ($\sigma^2 = 100.0$).

Fig. 5.9 shows the behavior of the chain in the case of a variance parameter for the proposal distribution that is too small in comparison to the scale of the target density. In this case, any proposal move around the previously accepted state would almost always be accepted and the chain traverses the state space very slowly (see Fig. 5.9a). Consequently, the resulting samples from the chain (with the same length of 50'000 samples in this example) would not be representative of the target distribution as illustrated in Fig. 5.9b. It is important to note that, in both cases, the chain would eventually converge in distribution for both parameters. But this convergence might not be attained for a Markov chain of a practical length.

*Under-dispersed
proposal distribution*

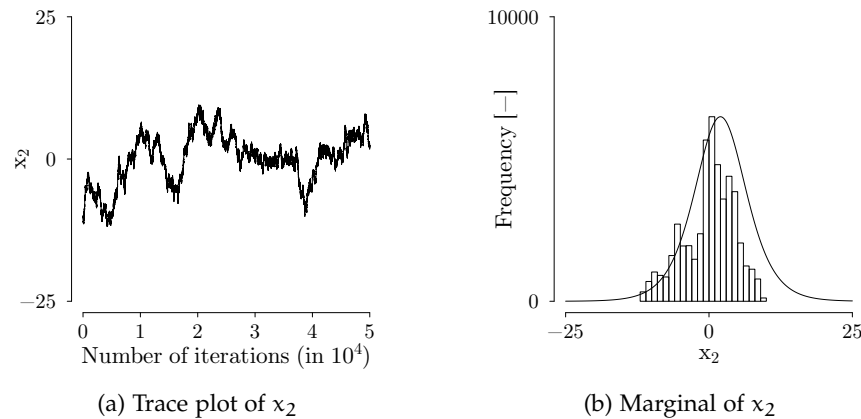


Figure 5.9: Convergence issue due to an under-dispersed proposal distribution ($\sigma^2 = 0.01$).

Tuning and adaptive
MH algorithms

In fact, the optimal choice of the scale parameters for the proposal distribution is closely related to the characteristic length scale of the target distribution [98]. In practice, there would be little information, if any, on the characteristic length scale for each parameter of a given density. As such, some tuning is required regarding the proposal distribution. This is the main motivation for the development of various *adaptive* MH algorithms. In such algorithms, the proposal distribution (and thus the transition kernels) are adapted during the iteration to optimize the performance of the algorithm [224].

Instead of delving into any of the particular improvements on the MH algorithm (such as through the adaptive schemes)¹⁴, this thesis adopted a relatively new MCMC algorithm based on an *ensemble* Markov chain. The algorithm has the potential of requiring minimal tuning to any given particular problem. The main ideas presented above (i.e., proposal move and its acceptance probability) remain central in an ensemble MCMC algorithm.

5.3.4 Affine-Invariant Ensemble Sampler (AIES)

AIES, motivation

Affine-invariant ensemble sampler (AIES) is an MCMC algorithm proposed by Goodman and Weare [108]. Its main motivation is exactly where the previous section left off: the difficulty in tuning or adapting MH algorithm to make them applicable to a wide class of target distribution. The situation is typically worsened for a highly correlated distribution of high dimension where some of the length scales of the target distribution are very small forcing an adaptive algorithm to spend the majority of its time tuning the scale of the proposal dis-

¹⁴ Ref. [106] provides an overview of adaptive MC algorithms, while Refs. [227–229] are examples of adaptive algorithms.

tribution, eventually resulting in a high overhead computational cost for adaptation [109].

Through an affine transformation (to be defined further below), a target distribution with a highly skewed aspect ratio¹⁵ can be made less skewed and thus easier to be sampled from [230]. By ensuring the algorithm to be affine-invariant, the performance of the algorithm would then be equal under all affine transformations of the target distribution. Finally, implementing such a transformation to an ensemble sampler¹⁶ results in an algorithm that requires minimal tuning with respect to each of the state space dimensions (explained below).

AIES belongs to a class of MCMC algorithms that generates a Markov chain on the state space of *ensembles* (i.e., *ensemble samplers*) [108]. An ensemble $\vec{\mathcal{X}}$ is a collection of L random variables $\{\mathcal{X}_l\}_{l=1}^L$ called *walkers*, each of which is in \mathbb{R}^D . That is, $\mathcal{X}_l = [x_{l,1}, x_{l,2}, \dots, x_{l,D}]$. The ensemble of L walkers are independent of each other with respect to the target distribution $p(\circ)$. Specifically,

$$p(\vec{\mathcal{X}}) = p(\mathcal{X}_1) \cdot p(\mathcal{X}_2) \cdot \dots \cdot p(\mathcal{X}_L) \quad (5.33)$$

Eq. (5.33) implies that the target distribution is being independently sampled by L walkers.

A Markov chain of an ensemble, in turn, is a sequence of ensembles, $\{\vec{\mathcal{X}}^{(i)}\}$ for $i \geq 0$ that follows the Markov property while preserving the condition in Eq. (5.33). Consequently, the Markov property lies on the state space of the ensemble¹⁷, while the sequence of each walker $\{\mathcal{X}_l^{(i)}\}$ itself needs not be Markovian [108].

In AIES, the transition between states of an ensemble is conducted by carrying out an affine transformation to the ensemble. The transition (thus the transformation) is carried out at the level of individual walkers one at a time. In other words, an affine transformation f_a is defined such that,

$$f_a : \mathcal{X}_l \mapsto \mathcal{Y}_l = \mathbf{M}\mathcal{X}_l + \mathbf{b} \quad (5.34)$$

When applied to an ensemble of multivariate random variables $\vec{\mathcal{X}}$,

$$\vec{\mathcal{Y}} = [\mathbf{M}\mathcal{X}_1 + \mathbf{b}, \mathbf{M}\mathcal{X}_2 + \mathbf{b}, \dots, \mathbf{M}\mathcal{X}_L + \mathbf{b}] \quad (5.35)$$

where \mathbf{M} is a $D \times D$ invertible matrix; and \mathbf{b} is a D -dimensional vector. In the context of MCMC simulation, $\vec{\mathcal{Y}}$ would represent the proposal move of the chain transition whose acceptance is subject to chance as will be discussed further below.

Ensemble sampler

Affine Transformation

¹⁵ Loosely defined as the ratio between different characteristic length scales of different dimensions.

¹⁶ *Sampler of many particles* creating many paths, as opposed to a sampler of a *single particle* in the conventional MCMC algorithm, such as the previous MH algorithm.

¹⁷ If each walker is in \mathbb{R}^D then an ensemble of L walkers can be thought of to be in $\mathbb{R}^{D \cdot L}$.

Suppose \mathcal{X} is a multivariate random variable with a state space $\mathcal{X} \subseteq \mathbb{R}^D$ and has a multivariate PDF p . Let $\mathcal{Y} = \mathbf{M}\mathcal{X} + \mathbf{b}$, then the PDF of \mathcal{Y} , according to the change of variables rule [213], is given by

$$g(\mathbf{y}) = \frac{1}{|\det \mathbf{M}|} p(\mathbf{M}^{-1}(\mathbf{y} - \mathbf{b})) = \frac{1}{|\det \mathbf{M}|} p(\mathbf{x}) \propto p(\mathbf{x}) \quad (5.36)$$

In other words, barring a proportionality constant, \mathcal{Y} is distributed the same way as \mathcal{X} . An ensemble MCMC algorithm is called *affine-invariant* if the transition kernel of the algorithm follows the same transformation:

*Affine-invariant
algorithm*

$$T(\vec{\mathbf{y}}, \vec{\mathbf{y}}^*) = C_{\mathbf{M}, \mathbf{b}} T(\vec{\mathbf{x}}, \vec{\mathbf{x}}^*) \quad (5.37)$$

where $T(\vec{\mathbf{y}}, \vec{\mathbf{y}}^*)$ is the transition kernel of the transformed variable; $T(\vec{\mathbf{x}}, \vec{\mathbf{x}}^*)$ is the transition kernel of the original variable; and $C_{\mathbf{M}, \mathbf{b}}$ is a normalizing constant of the transition kernel, independent of the variable. This implies that the algorithm sees no difference between sampling the transformed variables or the original variables. An affine-invariant algorithm thus requires no modification under any affine transformation of the variables [108, 109, 231].

AIES, stretch-move

One particular implementation of an AIES MCMC algorithm is the so-called *stretch-move* [108, 109]. As mentioned, the transition between iterations of an ensemble starts at the level of individual walkers. That is, the update is carried out one walker at a time and for stretch-move it proceeds as follows.

*Complementary
ensemble*

Let $\mathcal{X}_l^{(i-1)}$ be the walker l at iteration $(i-1)$ that need to be updated. Let $\vec{\mathcal{X}}_{\sim l}^{(i-1)}$, called a *complementary ensemble*, be the ensemble of walkers at iteration $(i-1)$, complementary to $\mathcal{X}_l^{(i-1)}$. Specifically,

$$\vec{\mathcal{X}}_{\sim l}^{(i-1)} = [\mathcal{X}_1^{(i-1)}, \dots, \mathcal{X}_{l-1}^{(i-1)}, \mathcal{X}_{l+1}^{(i-1)}, \dots, \mathcal{X}_L^{(i-1)}] \quad (5.38)$$

where all the walkers $k < l$ have been updated to their respective new states. Finally, let $\vec{\mathbf{x}}^{(0)} = [\mathbf{x}_1^{(0)}, \dots, \mathbf{x}_L^{(0)}]$ be the initial state of the chain, arbitrarily chosen within the support of \mathcal{X} .

Proposal move

In transitioning the ensemble to $\vec{\mathbf{x}}^{(i)}$, a proposal move is made for one walker at a time and it follows an affine transformation:

$$\mathbf{x}_l^* = \mathbf{x}_j + z(\mathbf{x}_l^{(i-1)} - \mathbf{x}_j) \quad (5.39)$$

where \mathbf{x}_l^* is the proposal move for the walker l at the current iteration; $\mathbf{x}_l^{(i-1)}$ is the walker to be updated (i.e., walker l at the previous state $(i-1)$); \mathbf{x}_j is the *complementary walker*, randomly selected from the complementary ensembles of $\vec{\mathcal{X}}_{\sim l}^{(i-1)}$; and z is the scaler of the transformation (i.e., the *stretcher*), randomly generated from,

$$q(z) \propto z^{-0.5}, \quad z \in [a^{-1}, a] \quad (5.40)$$

where a is a free positive parameter, and where the value of 2.0 is widely used as default for many applications [104, 108, 109, 231, 232].

Fig. 5.10 illustrates how a move is proposed in stretch-move for a single walker in an ensemble of 10 walkers, in a 2-dimensional state space. First, a walker in the ensemble will be updated while the rest of the walkers becomes its complementary walkers (Fig. 5.10a). Secondly, a complementary walker is randomly selected among the ensemble of complementary walkers (Fig. 5.10b). Thirdly and finally, a proposal scaler is randomly generated according to g and a proposed move is made according to Eq. (5.39) (see Fig. 5.10c).

Stretch-move, illustrated

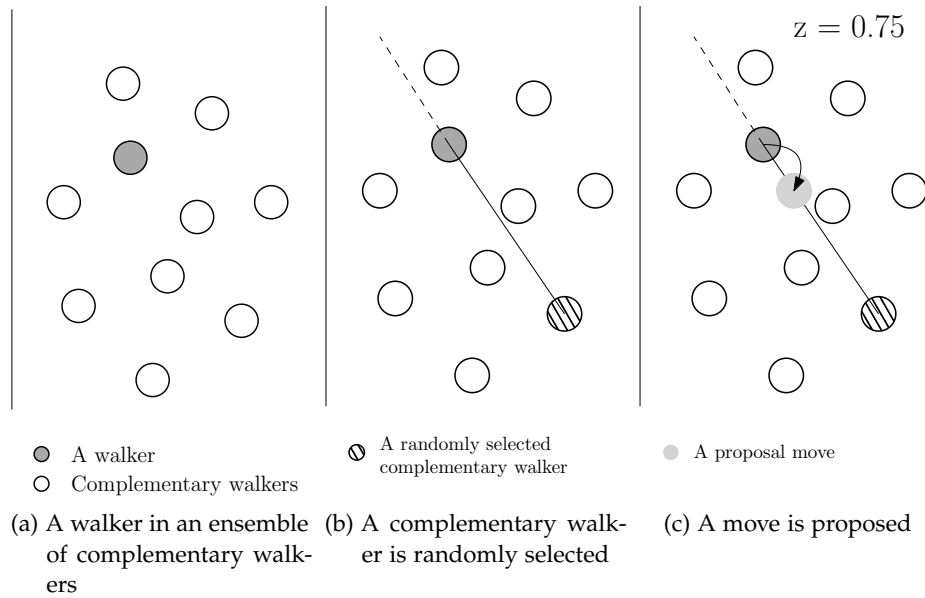


Figure 5.10: Illustration of a *stretch move* update for a single walker in a 2-dimensional state space.

The probability of accepting this proposal move is given by,

Acceptance probability

$$\alpha = \min \left(\frac{p^*(\mathbf{x}_l^*)}{p^*(\mathbf{x}_l^{(i-1)})} \times z^{D-1}, 1.0 \right) \quad (5.41)$$

where $p^*(\mathbf{x}_l^*)$ and $p^*(\mathbf{x}_l^{(i-1)})$ are the values of the (unnormalized) target density at the proposed and previous states, respectively; and D is the dimension of the state space. As with the MH algorithm, if accepted, the proposal move becomes the current state of the walker; otherwise, it remains in the previous state. The steps are then repeated for the current iteration until all the walkers in the ensemble have been updated. Algorithm 3 summarizes the steps in the stretch-move AIES MCMC algorithm.

To illustrate the application of the AIES algorithm for generating samples from an arbitrary target distribution, consider again the example of generating samples from the unnormalized PDF given in Eq. (5.21). The number of walkers is set to be 100 and the algorithm is run for 500 iterations. In other words, there is 50'000 target density evaluations. The initial state of the chain $\bar{\mathbf{x}}^{(0)}$ is set to be at the origin for all walkers.

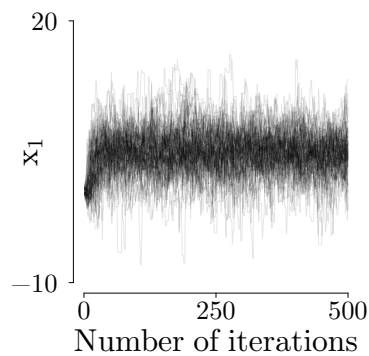
AIES algorithm, illustrated

Algorithm 3 Affine-Invariant Ensemble Sampler (Stretch Move)
 Generate samples from $p(\mathbf{x}) \propto p^*(\mathbf{x})$ using L walkers in I iterations.

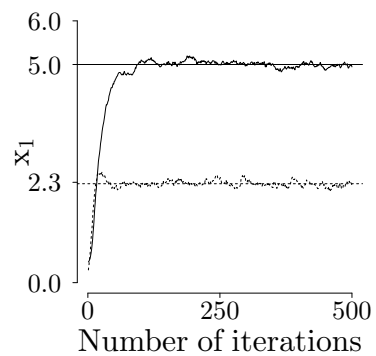
Require: $I > 0$, $L \geq I + 1$, $p^*(\mathbf{x})$, and $q(z)$
 $\bar{\mathbf{x}}^{(0)} = [\mathbf{x}_1^{(0)}, \dots, \mathbf{x}_L^{(0)}] \leftarrow \bar{\mathbf{x}}_0; \forall \bar{\mathbf{x}}_0 \in \mathbf{X}^L$
for $i = 1$ to I **do**
 for $l = 1$ to L **do**
 pick randomly \mathbf{x}_j from $\bar{\mathbf{x}}_{\neq l}^{(i-1)}$
 sample z from $q(z)$ (e.g., Eq. (5.41))
 $\mathbf{x}^* \leftarrow \mathbf{x}_j + z(\mathbf{x}_l^{(i-1)} - \mathbf{x}_j)$
 $\alpha \leftarrow \min\left(\frac{p^*(\mathbf{x}^*)}{p^*(\mathbf{x}_l^{(i-1)})} \times z^{D-1}, 1.0\right)$
 sample u from $\mathcal{U}[0, 1]$
 if $u < \alpha$ **then**
 $\mathbf{x}_l^{(i)} \leftarrow \mathbf{x}^*$
 else
 $\mathbf{x}_l^{(i)} \leftarrow \mathbf{x}_l^{(i-1)}$
 end if
 end for
end for

*ensemble samples,
 trace plot*

Fig. 5.11a shows the evolution of each individual walker traversing the state space of variable x_1 . The Markov property of an ensemble is not guaranteed for each individual walker. Thus, it is more suitable to graphically diagnose the chain by plotting all the individual walkers together. It is shown here that after an obvious initial phase, the ensemble seems to converge and to stay around a particular region of the state space. The width of the darker region, indicating region of the state space visited more often, shows the dispersion of the variable. This is confirmed by plotting the running empirical mean and standard deviation of the ensemble (Fig. 5.11b). The values 5.0 and 2.3 shown in the plot are the analytical mean and standard deviation of x_1 , respectively. Though not shown, x_2 has a similar behavior.



(a) Trace plot of all walkers



(b) Running empirical mean and standard deviation

Figure 5.11: Trace plots of individual walkers and the running mean and standard deviation for x_1 .

After the iterations are completed, the resulting samples should be distributed according to the target distribution. Fig. 5.12 shows that the resulting samples are distributed according to the target distribution both as a joint as well as its marginal. Shown here are the samples with the aforementioned “initialization phase” removed from the final tally. It was estimated from the plot to last for about 100 iterations. Thus the final number of samples presented below is $(500 - 100) \times 100 = 40'000$ samples.

*ensemble samples,
marginal*

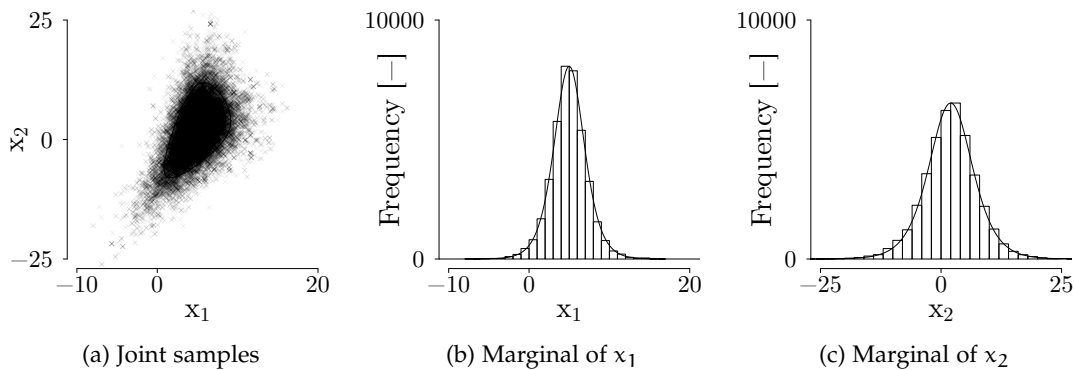


Figure 5.12: Results of samples generated by AIES. After 500 iterations of an ensemble of 100 walkers (for a total 50'000 target density evaluations), the samples resembles the actual shape of the distribution both as the joint distribution (Left) and as marginal distributions (Center and Right). The marginal densities have been normalized to match the peak of the histogram.

The AIES algorithm requires minimal tuning to generate samples from an arbitrary target distribution. The intuition behind this is that instead of trying to adapt the proposal distribution during the iterations (possibly with several associated tuning parameters), a proposal move relies on the information carried by the previous positions of the complementary ensemble. With a large number of walkers, thus with larger computational cost per iteration, more information will automatically be available per iteration regarding the landscape of the distribution. And although the parameter α in Eq. (5.40) is one potential tuning parameter, most applications work well with the default value of 2.0. As such, the algorithm required mainly the decision on the number of walkers L and total number of iterations I (these two yields the total number of target density evaluations $L \times I$).

*AIES algorithm,
intuition*

The specific distribution of the ensemble at the beginning, however, might cause a relatively long “initialization phase” noted above (compared to the total length of the iterations). This is especially the case if the chain starts at very atypical values of the distribution. Having a large ensemble implies a larger “inertia” for each of them to move and settle to a more typical region of the state space. The algorithm was also recently reported to badly scale with the number of dimensions and to eventually fail in a high dimension ($D \geq 50$) either by a

*AIES algorithm,
possible problems*

very slow convergence or, more importantly, by a biased convergence (i.e., convergence to a wrong value) [230, 233]. The number of dimensions concerned in this thesis, however, is still far below the value cited above.

There are two important issues not discussed in detail in either of the illustrations above, namely the convergence and the required length of the chain. These issues are of practical importance for any MCMC algorithm. For instance, the “initialization phase” of the AIES algorithm is an indication of a particular lack of convergence to stationarity in the initial part of the chain. It needs to be detected and removed lest it would bias the estimation using its samples. Rigorous proof of stationarity is difficult to obtain in practice and instead, analysis is often based on empirical diagnostic with heuristics. Afterward, the main question is on how long the chain needs to be to obtain a small enough statistical error on the MCMC estimate. Analyzing a realization of a Markov chain for Monte Carlo application is the subject of the next section.

5.4 DIAGNOSING MCMC SAMPLES

Consider once more Eq. (5.29), the central limit theorem (CLT) for the MCMC. Let $\{\mathcal{X}^{(i)}\}_{i=1}^I$ be a stationary Markov chain of length I with a stationary distribution π_t (i.e., the target distribution). The chain is used to estimate the expectation of a function f under π_t such that $\hat{f} = 1/I \sum_i f(\mathcal{X}^{(i)})$. The asymptotic error of the estimator is as follows

$$\lim_{I \rightarrow \infty} \hat{f} - \mathbb{E}_{\pi_t}[f] \sim \mathcal{N}\left(0, \frac{\sigma_f^2}{I}\right)$$

where σ_f^2 is the variance of a given function f evaluated under the stationary distribution of the chain. As mentioned, by construction, the successive realizations in a Markov chain are not independent. Before discussing in Section 5.4.1 the implication of this correlation on the statistical error $\frac{\sigma_f^2}{I}$, several important concepts are introduced below¹⁸.

Autocovariance The autocovariance of the function f is defined as the covariation between the function f evaluated using the chain at the iteration i and iteration j . That is,

$$\text{Cov}\left[f(\mathcal{X}^{(i)}), f(\mathcal{X}^{(j)})\right] \equiv \mathbb{E}\left[\left(f(\mathcal{X}^{(i)}) - \mathbb{E}[f]\right) \cdot \left(f(\mathcal{X}^{(j)}) - \mathbb{E}[f]\right)\right] \quad (5.42)$$

Assuming a stationary chain, the covariance becomes only a function of the separation between the two iterations irrespective at which particular point in the chain the function is evaluated. In other words,

$$\text{Cov}\left[f(\mathcal{X}^{(i)}), f(\mathcal{X}^{(i+t)})\right] \equiv C_f(t) \quad (5.43)$$

¹⁸ Note that the dependence on π_t in the discussion below is implicitly assumed and suppressed from the notation.

where C_f is the autocovariance function of f and t is the *lag* of the covariance function. Note that according to the notation above the covariance function C_f is always defined with respect to a given function f , as indicated in the subscript. Such function includes function that returns a particular dimension of a multivariate chain, e.g., $f(\mathcal{X}^{(i)}) = \mathcal{X}_d^{(i)}$, with d a dimension of the multivariate chain.

The autocorrelation function ρ_f of f is the normalized autocovariance function defined as,

$$\rho_f(t) \equiv \frac{C_f(t)}{C_f(0)} \tag{5.44}$$

Autocorrelation function

where $C_f(0)$ is the lag-0 autocovariance and is equal to the process variance $\mathbb{V}[f]$ for a stationary process.

Autocovariance (and autocorrelation) measures the strength of the covariance (and correlation) between samples in a Markov chain. By definition, both functions are symmetric about the origin. They also tend to decay with increasing lag as illustrated in the figure below for three different Markov chains with different autocorrelations.

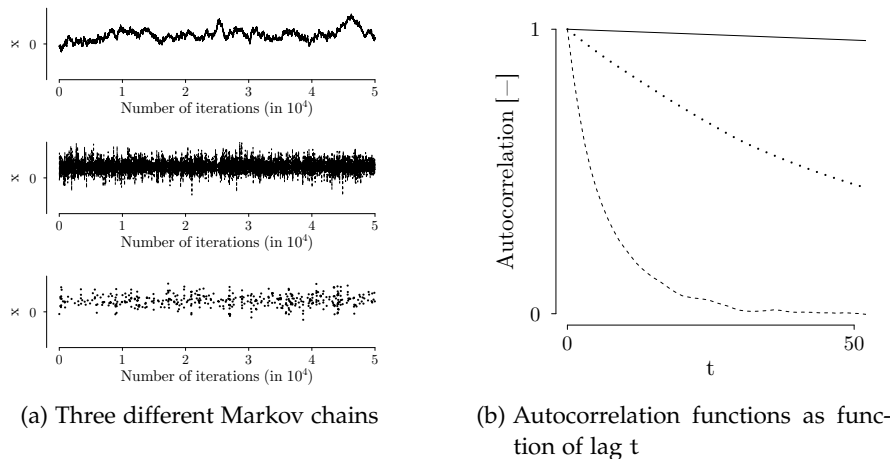


Figure 5.13: Illustration of autocorrelation functions for three different Markov chains. The Markov chain at the top has the strongest autocorrelation shown as solid line in the right plot. The correlation between samples dies off much slower than the other two chains.

5.4.1 Autocorrelation in Equilibrium and Thinning

Assuming that the stationary chain has been attained, the asymptotic results of Eq. (5.29) applies. To derive the expression for $\sigma_{\hat{f}}^2$ in the equation, consider that the variance of the MCMC estimator \hat{f} for the expected value of f computed by a Markov chain $\{\mathcal{X}^{(i)}\}$ of length I is,

$$\mathbb{V}[\hat{f}] = \mathbb{E} \left[\left(\frac{1}{I} \sum_{i=1}^I \left(f[\mathcal{X}^{(i)}] - \mathbb{E}[f] \right) \right)^2 \right] \quad (5.45)$$

By using nested sum, the definition can be rewritten as,

$$\begin{aligned} \mathbb{V}[\hat{f}] &= \mathbb{E} \left[\frac{1}{I^2} \sum_{i=1}^I \sum_{j=1}^I \left(f[\mathcal{X}^{(i)}] - \mathbb{E}[f] \right) \cdot \left(f[\mathcal{X}^{(j)}] - \mathbb{E}[f] \right) \right] \\ &= \frac{1}{I^2} \sum_{i=1}^I \sum_{j=1}^I \mathbb{E} \left[\left(f[\mathcal{X}^{(i)}] - \mathbb{E}[f] \right) \cdot \left(f[\mathcal{X}^{(j)}] - \mathbb{E}[f] \right) \right] \\ &= \frac{1}{I^2} \sum_{i=1}^I \sum_{j=1}^I \text{Cov} \left[f(\mathcal{X}^{(i)}), f(\mathcal{X}^{(j)}) \right] \end{aligned}$$

where the definition of covariance between two random variables have been applied to arrive to the last line above.

Assuming that the chain $\{\mathcal{X}^{(i)}\}$ is stationary then the covariance function is simply a function of the separation between the two iterations i and j ,

$$\begin{aligned} \mathbb{V}[\hat{f}] &= \frac{1}{I^2} \sum_{i=1}^I \sum_{j=1}^I \text{Cov} \left[f(\mathcal{X}^{(i)}), f(\mathcal{X}^{(j)}) \right] \\ &= \frac{1}{I^2} \sum_{i=1}^I \sum_{j=1}^I C_f(|i-j|) \\ &\approx \frac{1}{I^2} \sum_{i=1}^I \sum_{t=-\infty}^{\infty} C_f(|t|) = \frac{1}{I} \sum_{t=-\infty}^{\infty} C_f(|t|) \end{aligned}$$

where C_f is the (stationary) autocovariance function associated with function f . The approximation in the last line above is valid assuming that C_f decays as the separation $t = i - j$ becomes larger [219]. Noting that the covariance function C_f is a symmetric function about zero,

$$\begin{aligned} \mathbb{V}[\hat{f}] &= \frac{1}{I} \left(C_f(0) + 2 \sum_{t=1}^{\infty} C_f(t) \right) \\ &= \frac{2C_f(0)}{I} \left(\frac{1}{2} + \sum_{t=1}^{\infty} \rho_f(t) \right) \end{aligned}$$

where ρ_f is the autocorrelation function of f defined in Eq. (5.44). Rearranging the term, the variance of the MCMC estimator \hat{f} is,

$$\mathbb{V}[\hat{f}] = \frac{2\tau_{\text{int},f}}{I} C_f(0) \quad \tau_{\text{int},f} \equiv \frac{1}{2} + \sum_{t=1}^{\infty} \rho_f(t) \quad (5.46)$$

Integrated autocorrelation time

where $\tau_{\text{int},f}$ is the *integrated autocorrelation time* (or, in this thesis, simply the autocorrelation time).

Comparing with Eq. (5.29), Eq. (5.46) can be interpreted in two ways. First, the use of a Markov chain of length I to estimate the expectation of f inflates the estimator variance by a factor $2\tau_{\text{int},f}$. In other words, $\sigma_f^2 \equiv 2\tau_{\text{int},f} C_f(0)$. Or, equivalently, the number of independent samples required in the computation of the MC sampling variance is only a factor of $\frac{1}{2\tau_{\text{int},f}}$ of the total number of MCMC samples I . In other words, $N \equiv \frac{I}{2\tau_{\text{int},f}}$ with N the number of independent samples (or *effective sample size*). From this latter interpretation, the autocorrelation time gives a measure of the number of MCMC iterations required to generate a single independent sample.

Effective sample size

In either interpretation, the statistical error associated with the MC estimation is larger for an estimation using MCMC samples than using independent samples due to the inherent correlation. Moreover, in the case of MCMC, the autocorrelation time $\tau_{\text{int},f}$ directly affects the statistical error. As such, this quantity is useful in either determining the required length of the chain in a MCMC simulation or assessing the statistical error of a chain of a given length.

Determining the length of a chain

That is, the minimum length of the chain can be determined such that the statistical error of the estimator of f $\mathbb{V}[\hat{f}]^{0.5}$ is smaller than a fraction ϵ of the true standard deviation $\mathbb{V}[f]^{0.5}$ [232],

$$\left(\frac{\mathbb{V}[\hat{f}]}{C_f(0)} \right)^{0.5} = \left(\frac{\mathbb{V}[\hat{f}]}{\mathbb{V}[f]} \right)^{0.5} = \left(\frac{2\tau_{\text{int},f}}{I} \right)^{0.5} \leq \epsilon \Leftrightarrow I \geq \frac{2\tau_{\text{int},f}}{\epsilon^2} \quad (5.47)$$

where I is the number of MCMC samples. For a single particle MCMC this number is also the number of MCMC iterations¹⁹. For an ensemble sampler, the total number of samples is a multiplication between the number of iterations I and the number of walkers L such that,

$$I \geq \frac{2\tau_{\text{int},f}}{\epsilon^2 L} \quad (5.48)$$

The autocorrelation time gives a measure of the required length of the chain to reach a target statistical error. For instance, 10'000 MCMC iterations are relatively long for a single particle chain having an autocorrelation time of 5 ($\epsilon \approx 3\%$), but much shorter in comparison to a chain having an autocorrelation time of 100 ($\epsilon \approx 14\%$). In the latter case, in such a relatively short chain, the reliability of the estimation of the autocorrelation time itself can actually become suspect.

¹⁹ hence the notation I .

Estimating
autocorrelation time

The autocovariance function of f can be estimated from a realization of a Markov chain $\{\mathbf{x}^{(i)}\}_{i=1}^I$ using the following estimator [219, 232],

$$\hat{C}_f(t) = \frac{1}{I-t} \sum_{i=1}^{I-t} (f(\mathbf{x}^{(i)}) - \bar{f}) (f(\mathbf{x}^{(i+t)}) - \bar{f}) \quad (5.49)$$

where $\bar{f} = 1/I \sum_i f(\mathbf{x}^{(i)})$ is the sample mean of f . The autocorrelation function estimator follows,

$$\hat{\rho}_f(t) = \frac{\hat{C}_f(t)}{\hat{C}_f(0)} \quad (5.50)$$

Finally, the estimator of the autocorrelation time is given by,

$$\hat{\tau}_{t,f} = \frac{1}{2} + \sum_{t=1}^I \hat{\rho}_f(t) \quad (5.51)$$

Direct estimation of the autocorrelation time using the above estimator is usually unstable due to the statistical noise associated with large lag t . For a large t , the data of iterations that are far separated becomes too sparse to have a reliable estimate of the autocovariance function value. A “windowing” technique is proposed in [219] to stabilize the estimation and is implemented in the routine ACOR [234, 235] used in this thesis.

On thinning the
chain

The notion of independent samples (or effective sample size) mentioned above are coupled with the practice of *thinning* or sub-sampling the MCMC samples [236]. Thinning the chain means that only 1 MCMC sample is kept for every k iterations, with k an integer. Often, the integer k is chosen to be close to, or at least the same order of magnitude as, twice the autocorrelation time $\tau_{\text{int},f}$. Historically, thinning was mainly motivated by the limited storage and memory to store all the samples, especially for a long running MCMC simulation. As the end purpose of conducting MC estimation requires the use of independent samples, it was argued that by thinning, only samples that matters (i.e., independent) were kept for further analysis.

On the other hand, the practice of thinning is questioned by several authors [236, 237]. The argument against thinning is relatively easy to intuit: there can be much a smaller number of samples left after thinning that often worsen the accuracy of the estimator.

Thinning can be justified in the case of expensive “post-processing” of the MCMC samples. That is, if the function f is expensive to evaluate. For example, in the context of Bayesian data analysis, the posterior samples obtained via MCMC simulation might need to be propagated through a computationally expensive code for the purpose of forward uncertainty quantification by an ordinary MC simulation. In this case, only calculations of a few samples might be afforded and, as required by MC simulation, such selection of samples should be independent of each other which, in turn, can be achieved by thinning the chain first.

5.4.2 Initialization Bias and Burn-in

The results above hold for MCMC samples obtained from a stationary chain. As illustrated in Fig. 5.11a, the MCMC samples at the beginning of the chain are taken from a lower probability region of the target probability distribution and thus are less representative of the distribution. After some period, however, the apparent “initial transient” eventually dies off and the chain settles in a more typical region of the parameter space. The colloquial term *burn-in* period is used to characterize this initial part of the chain. By *burn-in* the chain, it is meant that the samples of that period are discarded.

Burn-in

As opposed to the previous discussion on autocorrelation in equilibrium that causes inflation of MC *statistical error*, the presence of burn-in period in an MCMC simulation can cause a *systematic bias* on the MC estimator [219]. Therefore, for a short chain, it is important to remove the burn-in period lest the estimator will be heavily affected by initialization bias. In a single particle MCMC algorithm discarding the samples corresponding to the burn-in period is generally not necessary as the total number of iterations tends to be much larger than this period and the simulation can be extended to wash out that initialization bias.

Initialization bias

On the other hand, in an ensemble sampler, burn-in period might contain much larger biased samples relative to the total number of samples due to the use of multiple walkers within an iteration [109, 231, 232]. Multiple walkers can be initialized from low probability regions. Furthermore, although an iteration computationally costs more, the number of iterations in an ensemble sampler is relatively shorter than in a single particle sampler. As such, determining the length of burn-in period – and discarding the corresponding samples – is more important for ensemble samplers.

Burn-in period can be associated with the relaxation period of an infinitely long Markov chain [219]. In that case, the autocorrelation function of Eq. (5.44) can often be expressed as an exponential law, i.e., $\rho_f(t) \approx \exp(-t/\tau_{\text{exp},f})$. Under this assumption, the *exponential* autocorrelation time of a function *f* can be defined as follow,

Exponential autocorrelation time

$$\tau_{\text{exp},f} = \lim_{t \rightarrow \infty} \frac{t}{-\ln|\rho_f(t)|} \tag{5.52}$$

The time represents the rate of convergence of a Markov chain with respect to the given function *f*, starting from arbitrary initial values to its stationary distribution [219]. It can be used to determine the number of iterations to discard before the chain is considered stationary.

The relationship between exponential correlation time and stationarity only strictly holds for exponential correlation functions. Fortunately, most infinitely long chains have exponential correlation functions [219]. In addition, although the exponential autocorrelation time

is different from (twice) the integrated autocorrelation time²⁰, in practice, the two are often assumed to be the same [109, 232] or at least having the same order of magnitude [219].

Determining the number of iterations to discard

In this thesis, they are both estimated using the definition of the integrated autocorrelation time (Eq. (5.46)). That is, the autocorrelation time is first estimated for the whole chain and is used, after multiplication by a conservative factor of 20 [219], to determine the length of the burn-in period. Afterward, assuming that the stationary chain has been attained, the autocorrelation time is re-estimated and is used as the basis for assessing the autocorrelation between successive realizations (and if applies, for thinning).

Burn-in, a heuristic

In the end, it is important to acknowledge that the burn-in period as determined by the autocorrelation time is a heuristic. It is useful to avoid initialization bias from a Markov chain of a finite length especially in the case of ensemble samplers. Moreover, a suspiciously long burn-in period can give an indication of a more serious underlying problem either in the posterior formulation or the sampler. It cannot, however, establish the fact that a Markov chain has indeed settled in its stationary (and simultaneously, its target) distribution [105, 219].

Burn-in, alternatives

Finally, note that the initial transient is mainly due to selecting values with low posterior probability to initialize the chain. In a simple problem, perhaps it can be straightforward to determine a value that lies in a high probability region of the posterior. This is not the case for a more complex high-dimensional problem and some arbitrary values are often selected instead. Thus, some authors [232, 236] proposed instead to tune the initial values such that they are more representative of the posterior PDF and make do without burn-in. This tuning, however, requires additional preliminary calculations.

5.5 APPLICATION TO THE TRACE MODEL OF FEBA

In this section, a Bayesian calibration on the parameters of the TRACE reflow model is conducted and assessed on the basis of the data from a test series of the FEBA reflow experiment. Following the results of Chapter 3, only the eight most relevant reflow model parameters are considered in the calibration (while the uncertainties of the four parameters related to boundary conditions are optionally taken into account). Furthermore, following the developments in Chapter 4, a GP metamodel is used to substitute the code run of the TRACE model. There were six experimental runs conducted in the test series that corresponded to different experimental boundary conditions. The calibration is conducted solely on test No. 216, the base experimental run. The calibration results are then assessed by means of propagating the resulting model parameters posterior uncertainties and com-

²⁰ except in the case of exponentially decaying autocorrelation function.

paring the prediction uncertainties with the data from the other five experimental runs.

5.5.1 *Simulation Experiment*

The application of the Bayesian calibration framework on the TRACE reflood model parameters against the FEBA experimental data is based on six different statistical formulations, in the following referred to as *calibration schemes*. These schemes are distinguished by their respective assumptions:

- *w/ Bias, All*. The first calibration scheme assumes that the TRACE model is an imperfect simulator of the reflood phenomena in the FEBA experiment. As such it considers a model *bias* term (as described further below) in the calibration process. Furthermore, in this scheme, *all* available types of experimental data are considered. The data includes the clad temperature measurements at different time points and at different axial locations (will be succinctly referred to below as the TC output or data), the pressure drop measurements at different time points and at different axial segments (referred to as the DP output or data), and the collected liquid carryover measurement at different time points (referred to as the CO output or data). As mentioned, following the results of the previous chapter, only the eight most influential reflood model parameters are considered for the calibration.
- *w/ Bias, TC; w/ Bias, DP; and w/ Bias, CO* are three variants of the scheme *w/ Bias, All* in which only one type of experimental data (respectively, output) is considered at a time for the calibration. The purpose of these schemes is to investigate the effect of using different types of data from the same test to constrain the model parameters prior uncertainties. The calibration is still conducted for the eight reflood model parameters and considering the model bias term.
- *w/o Bias* scheme is similar to the scheme *w/ Bias, All*; it uses all available types of experimental data to calibrate the eight reflood model parameters, except that no model bias term is included in the formulation. In essence, this scheme assumes that the TRACE model perfectly describes the reflood phenomena in the FEBA test No. 216.
- *w/ Bias, no dffbVIHT* is the last calibration scheme considered; it is conducted to investigate the effect of excluding, from the calibration process, an influential parameter (*dffbVIHT*) that is later found from the scheme *w/ Bias, All* to be strongly correlated. Except for calibrating only the remaining seven reflood

model parameters, this scheme used similar assumptions as the first scheme.

The six calibration schemes above aim to update the prior uncertainties of the model parameters using the experimental data from FEBA test No. 216. The six posterior PDFs are then directly sampled using an ensemble MCMC sampler to obtain six different sets of posterior samples. To avoid the excessive computational cost of having to run TRACE hundreds of thousands of times, the GP metamodel developed in Chapter 4 is used to substitute the TRACE model.

These different sets of samples are then analyzed to assess the effect of using different calibration schemes in constraining the prior uncertainties of the model parameters. Finally, the same posterior samples are used in forward uncertainty quantification (UQ) on the TRACE model of different FEBA tests corresponding to different boundary conditions, namely system pressures and reflood rates. This final exercise aims to assess how the posterior uncertainties from the different calibration schemes perform under boundary conditions different from that of the calibration data.

In the following, the important terms of Eq. (5.3) will be discussed in the context of the present application to the TRACE model before detailing each calibration scheme. Afterward, the MCMC sampler as well as a method to evaluate and compare different posterior prediction uncertainties are presented.

5.5.1.1 *Experimental Data and Observation Layout*

The experimental data of FEBA test No. 216 used for the calibration was extracted from the experimental report [123], which was provided to the participants of the PREMIUM benchmark [124].

The experimental data provided for the clad temperature (TC) consists of 33 time points for each of the eight different axial locations of the thermocouples along the test section. Recall that by convention in the experiment, TC1 corresponds to the thermocouple measurement at the top of the test section (≈ 4.1 [m]), while TC8 corresponds to the measurement at the bottom (≈ 0.3 [m]) (see Table 2.2).

Due to the strong discontinuity of the clad temperature around the point of quenching, the model bias term cannot be modeled using a stationary GP (see Section 5.5.1.3) as it severely violates the constant variance assumption as function of time and axial location (at the very least, before and after the quenching occurs). To keep using a simple stationary GP formulation, the model bias term is modeled only for the part of the transient before the quenching. Thus, the calibration is also conducted using only the data prior to quenching. This is further justified by the fact that after quenching there is almost no relevant variation in the temperature transient.

Because of the different timing of quenching along the test section, the number of data points available for calibration changes per axial

*Clad temperature
(TC) data*

location. Based on these data points, an observation layout for the TC data can be defined,

$$\begin{aligned} \Lambda_{TC} = \{ & (z_1, t_1), (z_1, t_2), (z_2, t_1), \dots, (z_2, t_7), \\ & (z_3, t_1), \dots, (z_3, t_{12}), (z_4, t_1), \dots, (z_4, t_{17}), \\ & (z_5, t_1), \dots, (z_5, t_{21}), (z_6, t_1), \dots, (z_6, t_{24}), \\ & (z_7, t_1), \dots, (z_7, t_{25}), (z_8, t_1), \dots, (z_8, t_{27}) \} \end{aligned} \quad (5.53)$$

where z denotes the axial location and t denotes the time point. The total number of data points associated with the TC data is 133.

The reported experimental uncertainty associated with the clad temperature measurement is $\pm 0.5\%$ of the measured value in $[\text{°C}]$. In this thesis, this statement of uncertainty is translated into a Gaussian probability distribution such that the uncertainty covers the 99.7% probability (i.e., $3\text{-}\sigma$ level). Let $\mathbf{y}_{E,TC}$ be the vector of TC data observed at Λ_{TC} , then the experimental uncertainty is given as,

$$\begin{aligned} \mathcal{E}(\Lambda_{TC}) & \sim \mathcal{N}(0, \Sigma_{E,TC}) \\ \Sigma_{E,TC} & = \text{diag} \left(\left(\frac{0.005}{3} \mathbf{y}_{E,TC} \right)^2 \right) \end{aligned} \quad (5.54)$$

The dimension of the multivariate Gaussian random variable above is 133, i.e., the length of the observation layout Λ_{TC} . The random variable is independent but not identically distributed as the variance changes for each measurement point.

The experimental data provided for the pressure drop (DP) consists of 18 time points for each of the 4 different axial segments of the pressure drop measurements. Recall that in the experiment, the *bottom* segment corresponds to the segment $0.0 - 1.7$ [m], the *middle* to $1.7 - 2.3$ [m], the *top* to $z = 2.3 - 4.1$ [m], and the *total* to $0.0 - 4.1$ [m]. In the following, the bottom, middle, top, and total segments are simply indices of the DP output; z_1, z_2, z_3, z_4 , respectively. The observation layout for the DP data is then defined as follows,

$$\Lambda_{DP} = \{(z_1, t_1), \dots, (z_1, t_{18}), (z_2, t_1), \dots, (z_4, t_{18})\} \quad (5.55)$$

where z denotes the axial segment and t denotes the time point. The total length of the observation layout Λ_{DP} is 72.

The reported experimental uncertainty associated with the pressure drop measurement is $\pm 10\%$ of the measured value in [Pa]. As before, this statement of uncertainty is translated into a Gaussian probability distribution covering the 99.7% probability (i.e., $3\text{-}\sigma$ level). Let $\mathbf{y}_{E,DP}$ be the vector of DP data observed at Λ_{DP} , then the experimental uncertainty is given as a multivariate Gaussian,

$$\begin{aligned} \mathcal{E}(\Lambda_{DP}) & \sim \mathcal{N}(0, \Sigma_{E,DP}) \\ \Sigma_{E,DP} & = \text{diag} \left(\left(\frac{0.1}{3} \mathbf{y}_{E,DP} \right)^2 \right) \end{aligned} \quad (5.56)$$

Pressure drop (DP)
data

where the random variable is a 72-dimensional multivariate Gaussian.

*Liquid carryover
(CO) data*

Finally, the experimental data provided for the liquid carryover (CO) initially consisted of 16 time points. However, because the collecting tank was saturated at 10 [kg] only the transient up to that mass is of interest. By excluding the data points where the tank has been saturated, only 7 data points are available for calibration. Based on these data points $\mathbf{y}_{E,CO}$, the observation layout for the CO data is defined as,

$$\Lambda_{CO} = \{(t_1), \dots, (t_7)\} \quad (5.57)$$

where t denotes the time point.

A large uncertainty was indicated for the liquid carryover measurement that possibly includes biased measurement as the measured mass in the collecting tank does not always correspond to the liquid carryover of the reflood transient [21]. The suggested level of uncertainty for the benchmark was ± 0.5 [kg]. To cover the reported uncertainty and the possible bias, the reported level is assumed to be $1\text{-}\sigma$ level of an independent identically distributed multivariate Gaussian,

$$\mathcal{E}(\Lambda_{CO}) \sim \mathcal{N}(0, \mathbf{I}\sigma_{E,CO}^2) \quad (5.58)$$

where \mathbf{I} is an identity matrix of size 7, i.e., the length of the observation layout Λ_{CO} ; and $\sigma_{E,CO}$ is the standard deviation of the distribution, taken to be 0.5 [kg].

Finally, the observation layout for each output (data) type can be combined into a single long vector of the full observation layout, $\Lambda = \{\Lambda_{TC}, \Lambda_{DP}, \Lambda_{CO}\}$. The total number of data points and the length of the observation layout Λ used in the calibration are thus 212.

5.5.1.2 Gaussian Process Approximation for TRACE Simulations

Following the results of Chapter 4, three separate multivariate GP metamodels are used to approximate the TRACE predictions for each type of output (TC, DP, and CO). The hyper-parameters associated with these metamodels are separately estimated using actual TRACE runs \mathbf{Y} based on a design of experiment \mathbf{DM} (see the details in Section 4.6). After being estimated, the hyper-parameters of the GP metamodel are kept constant in the application of the metamodel.

Under the GP formulation, the simulator prediction for a given input \mathbf{x}_o (containing both the controllable inputs \mathbf{x}_c and the model

parameters \mathbf{x}_m) becomes a probabilistic model. The prediction of the TC output at the observation layout Λ_{TC} is formulated as follows,

$$\begin{aligned}
\mathcal{Y}_{M,TC}(\mathbf{x}_o)|\mathbf{Y} &\sim \mathcal{N}(\boldsymbol{\mu}_{M,TC}(\mathbf{x}_o), \boldsymbol{\Sigma}_{M,TC}(\mathbf{x}_o)) \\
\boldsymbol{\mu}_{M,TC} &= \bar{\mathbf{y}}_{TC} + \boldsymbol{\Phi}_{Q_{TC},TC}^* \mathbf{m}_{SK,TC}(\mathbf{x}_o) \\
\boldsymbol{\Sigma}_{M,TC} &= \boldsymbol{\Phi}_{Q_{TC},TC}^* \text{diag}(\mathbf{s}_{SK,TC}^2(\mathbf{x}_o)) \boldsymbol{\Phi}_{Q_{TC},TC}^{*T} + \boldsymbol{\Phi}_{>Q_{TC},TC}^* \mathbf{I} \boldsymbol{\Phi}_{>Q_{TC},TC}^{*T} \\
\mathbf{m}_{SK,TC} &= [m_{SK,TC,1}(\mathbf{x}_o), m_{SK,TC,2}(\mathbf{x}_o), \dots, m_{SK,TC,Q_{TC}}(\mathbf{x}_o)] \\
\mathbf{s}_{SK,TC}^2 &= [s_{SK,TC,1}^2(\mathbf{x}_o), s_{SK,TC,2}^2(\mathbf{x}_o), \dots, s_{SK,TC,Q_{TC}}^2(\mathbf{x}_o)]
\end{aligned} \tag{5.59}$$

where the notations above follow the convention of Section 4.5.3. According to Section 4.6, the number of retained principal components for the TC output Q_{TC} is selected to be 7.

Recall that the SVD was conducted on the full TRACE simulation output (in the case of the temperature output: at eight axial levels and at 10'000 time-steps) for the dimension reduction. However, some points of the full simulation output do not have a corresponding experimental data. As such, for the calibration, the observation layout Λ_{TC} is used to select the elements of the output mean vector $\bar{\mathbf{y}}_{TC}$, the eigenvectors $\boldsymbol{\Phi}_{Q_{TC},TC}^*$, and the unretained eigenvectors $\boldsymbol{\Phi}_{>Q_{TC},TC}^*$ such that they contain only the points in time where data are actually observed. The resulting dimension of the Gaussian distribution is thus 133.

Similar formulations are used for the GP metamodels with respect to the DP and CO outputs. According to Section 4.6, the numbers of retained principal components are 10 and 5 for the DP and TC outputs, respectively. Again, only the points in time which coincide with the observed data are selected according to the respective observation layout.

5.5.1.3 Modeling the Model Bias Term

According to Section 5.2.1, model bias term is represented using a Gaussian process (GP). A model bias term is formulated for each type of data (or output). The formulation of a Gaussian process (GP) for the model bias term is adapted from [93, 112, 238]:

1. Generate N realizations of the TRACE simulation for FEBA test No. 216 (randomly) varying only the 4 parameters related to the boundary conditions (namely, `breakP`, `fillT`, `fillV`, and `pwr`), while keeping the other 8 model parameters at their respective nominal values. Each output type of the TRACE simulations are selected at its respective observation layout (Λ_{TC} , Λ_{DP} , and Λ_{CO}). For each simulation the vectors of values are denoted $\hat{\mathbf{y}}_{M,TC}$, $\hat{\mathbf{y}}_{M,DP}$, and $\hat{\mathbf{y}}_{M,CO}$ for the TC, DP, CO output, respectively.

2. Assume the vectors $(\mathbf{y}_{E,TC} - \hat{\mathbf{y}}_{M,TC})$, $(\mathbf{y}_{E,DP} - \hat{\mathbf{y}}_{M,DP})$, and $(\mathbf{y}_{E,CO} - \hat{\mathbf{y}}_{M,CO})$ to be realizations from stationary GPs on the observation layouts (i.e., as function time and space). Note that as the observation layouts comprise discrete points in time and space the GPs collapse to multivariate Gaussian distributions. The power-exponential covariance function is selected for the covariance kernel and its hyper-parameters (i.e., σ^2 , \mathbf{p} , θ) are estimated using the R package `DiceKriging`.
3. The mean of the bias term of each output is taken to be the difference between the data $\mathbf{y}_{E,o}$ and the nominal prediction $\hat{\mathbf{y}}_{M,o}$, while the covariance matrix of the bias term is taken to be the covariance matrix constructed at the observation layout using the estimated hyper-parameters above. The constructed covariance matrix takes into account correlations of the model bias in time and space. In the present analysis the values of the hyper-parameters estimated in the previous step are kept constant during the subsequent phases of the calibration process.

The above model bias term formulation is only partially Bayesian as it uses the data to make the initial estimation. But as argued in Ref. [93] it is a pragmatic way to carry out the analysis as there is no independent data to formulate the bias. Several additional assumptions are made in this thesis to fully formulate the bias term. By varying the parameters related to the boundary conditions in the construction of the term implies that the experimental uncertainty in the boundary conditions is included in the model bias term. In the FEBA experiment, the effect of these uncertainties on the experimental data was not directly observed as there was no replication of the experiment at the same controllable inputs and the experimental uncertainty with respect to each type of data was taken as given from the benchmark specification.

By using the mean for the bias term defined above, any difference between the nominal TRACE prediction and the experimental data is corrected. This is an indirect way of putting a strong prior preference for the TRACE nominal prediction such that the model parameters should not dramatically be shifted to correct the mismatch between the experimental data and the TRACE prediction. In other words, it is a way to keep as much as possible the nominal TRACE prediction, which was already based on a long running V&V activities. Ref. [93] also recommends to allow the variance of the process to vary, but as the calibration conducted here is based only on the data from a single FEBA test, the variance is kept constant. In fact, this represents a pessimistic assumption in the sense that the data is not allowed to reduce the bias by altering the model parameters. This is in line with the definition of the the mean used before.

All in all, the calibration using this proposed model bias term thus aims to update the prior uncertainties of the model parameters as-

suming that the nominal prediction is centered around the data while keeping the variance unchanged.

Based on the above discussion, the model bias term for the TC output is expressed as,

$$\begin{aligned} \mathcal{D}_{TC} &\sim \mathcal{N}(\mathbf{m}_{\delta,TC}, \Sigma_{\delta,TC}) \\ \mathbf{m}_{\delta,TC} &= (\mathbf{y}_{E,TC} - \hat{\mathbf{y}}_{M,TC}) \\ \Sigma_{\delta,TC} &= \sigma_{TC}^2 R_{TC}(\Lambda_{TC}, \Lambda_{TC}) \end{aligned} \tag{5.60}$$

where $\mathbf{m}_{\delta,TC}$ and $\Sigma_{\delta,TC}$ are the mean vector and the covariance matrix of the model bias term with respect to the TC output, respectively; while σ_{TC}^2 and R_{TC} are the process variance and correlation function of the process representing the bias term, respectively. The bias term is a multivariate Gaussian random variable with the dimension of 133, equals to the length of the observation layout Λ_{TC} .

Similar formulations apply for the model bias terms with respect to the DP and CO outputs. As a final note, the formulation of the model bias term is supposed to include explicitly different controllable inputs \mathbf{x}_c to take into account possible change in the bias as function of the inputs. However, the present study considers for the calibration only the data from FEBA test No. 216 corresponding to a single combination of controllable inputs. Therefore, the parametrization of \mathbf{x}_c is dropped from the following notation.

5.5.1.4 Calibration Schemes

Having defined the elements of the generic calibration formula of Eq. (5.3) within the context of the present problem, the explicit formulation for each calibration scheme introduced at the beginning of this section can now be presented.

The calibration scheme *w/ Bias, All* combines the formulation of the calibration schemes *w/Bias, TC*, *w/Bias, DP*, and *w/Bias, CO*. As such, in the following the latter three calibration schemes are first presented. Combining the terms of the above according to Eq. (5.3) gives a similar formulation as Eq. (5.15), but specifically for the TC data generating process. That is, the process corresponding to the scheme *w/ Bias, TC* is,

w/ Bias, TC

$$\begin{aligned} \mathcal{Y}_{E,TC}|\mathbf{x}_m &\sim \mathcal{N}(\boldsymbol{\mu}_{TC}(\mathbf{x}_m), \Sigma_{TC}(\mathbf{x}_m)) \\ \boldsymbol{\mu}_{TC}(\mathbf{x}_m) &= \boldsymbol{\mu}_{M,TC}(\mathbf{x}_m) + \mathbf{m}_{\delta,TC} \\ \Sigma_{TC}(\mathbf{x}_m) &= \Sigma_{M,TC}(\mathbf{x}_m) + \Sigma_{\delta,TC} + \Sigma_{E,TC} \end{aligned} \tag{5.61}$$

where $\boldsymbol{\mu}_{TC}$ and $\Sigma_{TC}(\mathbf{x}_m)$ are the 133-dimensional mean vector and the 133×133 covariance matrix associated with the TC output/data, respectively. The mean vector $\boldsymbol{\mu}_{TC}$ consists of the mean vector of the GP metamodel prediction $\boldsymbol{\mu}_{M,TC}$ (Eq. (5.59)); and the mean vector of the model bias term $\mathbf{m}_{\delta,TC}$ (Eq. (5.60)). The covariance matrix $\Sigma_{TC}(\mathbf{x}_m)$ comprises the covariance matrix of the GP metamodel

prediction (Eq. (5.59)); the covariance matrix of the model bias term $\Sigma_{\delta,TC}$ (Eq. (5.59)); and the covariance matrix of the experimental uncertainty for TC data (Eq. (5.54)).

The data generating processes for the DP (respectively, CO), corresponding to the calibration scheme *w/ Bias, DP* (respectively, *w/ Bias, CO*), can be defined in a similar manner.

w/ Bias, All The data generating processes for the TC, DP, and CO data above are combined to arrive at the process corresponding to the calibration scheme *w/ Bias, All*. The main assumption in combining the data generating processes is independence between types of data [205]. That is, no a priori relationship between different types of data is assumed. This assumption greatly simplifies the problem and thus the joint process becomes a concatenation of the Gaussian random vector

$$\begin{aligned} \mathbf{y}_{E,\{TC,DP,CO\}}|\mathbf{x}_m &\sim \mathcal{N}(\boldsymbol{\mu}_{TC,DP,CO}(\mathbf{x}_m), \Sigma_{TC,DP,CO}(\mathbf{x}_m)) \\ \boldsymbol{\mu}_{\{TC,DP,CO\}}(\mathbf{x}_m) &= [\boldsymbol{\mu}_{TC}(\mathbf{x}_m), \boldsymbol{\mu}_{DP}(\mathbf{x}_m), \boldsymbol{\mu}_{CO}(\mathbf{x}_m)] \\ \Sigma_{\{TC,DP,CO\}}(\mathbf{x}_m) &= \text{diag}(\Sigma_{TC}(\mathbf{x}_m), \Sigma_{DP}(\mathbf{x}_m), \Sigma_{CO}(\mathbf{x}_m)) \end{aligned} \quad (5.62)$$

where $\boldsymbol{\mu}_{\{TC,DP,CO\}}$ is a 212-dimensional vector from the concatenation the mean vectors of TC, DP, and CO; and $\Sigma_{\{TC,DP,CO\}}$ is the 212×212 corresponding covariance matrix, which is the block diagonal matrix $\text{diag}(\Sigma_{TC}(\mathbf{x}_m), \Sigma_{DP}(\mathbf{x}_m), \Sigma_{CO}(\mathbf{x}_m))$.

w/o Bias The calibration scheme *w/o Bias* has a similar data generating process to that of the scheme *w/ Bias, All*, except that the mean vectors and the covariance matrices of the model bias term for each types of data have been removed from the formulation. Specifically, for the TC data, the vector $\mathbf{m}_{\delta,TC}$ and the covariance matrices $\Sigma_{\delta,TC}$ are removed from Eqs. (5.61). Similar approach applies for the DP and CO data.

w/ Bias, no dffbVIHT Lastly, the calibration scheme *w/ Bias, no dffbVIHT* has the same data generating process as Eq. (5.62), except that the parameter *dffbVIHT* is not part of \mathbf{x}_m . In discussion below, this also implies that the parameter is assigned no prior probability.

Likelihood functions Given the experimental data $\mathbf{y}_{E,TC}$, $\mathbf{y}_{E,DP}$, and $\mathbf{y}_{E,CO}$ for the TC, DP, and CO, respectively, the likelihood functions with respect to each of the calibration schemes above can be defined following Eq. (5.16). The likelihood is from the Gaussian density (the formula for the density is given in Appendix D.3). Note that the model parameters are embedded inside the likelihood function through the mean and the covariance of the GP metamodel prediction.

Posterior PDFs The posterior PDF is then formulated by assigning the prior PDF for the eight (respectively seven for the scheme *w/ Bias, no dffbVIHT*) important reflood model parameters \mathbf{x}_m from Table 2.5. The posterior PDF is defined for each of the calibration schemes using the respective likelihood functions. For instance, the posterior PDF for the

model parameters under the calibration scheme w/ Bias, TC up to a constant is written as,

$$p_{TC}(\mathbf{x}_m | \mathbf{y}_{E,TC}) \propto \mathcal{L}_{TC}(\mathbf{x}_m; \mathbf{y}_{E,TC}) \cdot p(\mathbf{x}_m) \quad (5.63)$$

where \mathcal{L}_{TC} is the likelihood function associated with the data generating process in Eq. (5.61). The five other likelihood functions and posterior PDFs are defined similarly. Table 5.1 summarizes the different calibration schemes considered in this study.

Table 5.1: Bayesian calibration schemes conducted for the TRACE reflow model parameters against data from FEBA test No. 216.

No.	Calibration Scheme	Model Bias Term	Types of Output			Reflow Model Parameters (total number)
			TC	DP	CO	
1	w/ Bias, All	✓	✓	✓	✓	All (8)
2	w/ Bias, TC	✓	✓			All (8)
3	w/ Bias, DP	✓		✓		All (8)
4	w/ Bias, CO	✓			✓	All (8)
5	w/o Bias		✓	✓	✓	All (8)
6	w/ Bias, no dffbVIHT	✓	✓	✓	✓	Excluding dffbVIHT (7)

5.5.1.5 MCMC Simulation using Ensemble Sampler

Each calibration scheme above results in a likelihood function, which when combined with the prior PDFs of the model parameters, yield a posterior PDFs. The 8-dimensional (respectively seven for the w/ Bias, no dffbVIHT scheme) posterior PDFs contain all the information on the model parameters conditional on the experimental data and the assumed prior uncertainties, under the assumed respective calibration scheme. To characterize the posterior uncertainties of the model parameters, samples are directly generated from the respective posterior PDF by means of MCMC simulation.

Although the use of GP metamodels alleviate the burden of having to run TRACE directly, evaluating the likelihood function requires an inversion of the covariance matrix. The computational cost of matrix inversion is still not negligible, especially considering the expected number of evaluations. Furthermore, although the AIES MCMC algorithm (Algorithm 3) is straightforward to implement, it is not readily applicable for using multiple CPU [109].

A parallelization of the AIES sampler was originally developed and implemented in the python package `emcee` [109]. The main design philosophy of `emcee` (and its ported R package `rgw` [239] used in this thesis) is that of a *portable* sampler. That is, the user simply has to code the posterior formulation (the likelihood and the prior) in the

respective generic computing environment (R or python), without the need to put the probabilistic model within a new framework²¹.

In the present study, 2'000 iterations are carried out for an ensemble of 1'000 walkers. The initial state of the ensemble is a tight random scatter around the nominal model parameter values. The total number of iterations depends on the convergence of the MCMC simulation (discarding the initialization bias) and the required level of statistical error as detailed in Section 5.4. For the present study, they are assessed after-the-fact and the results is indeed found to be sufficient. Meanwhile, there is no clear cut rule for choosing the number of walkers L [109]. Larger number of walkers requires a higher computational cost per iteration but yields more independent samples per iteration. At the same time, larger number of walkers might cause more of the initial calculations to be discarded as more calculations are required to settle the ensemble in the typical region of the posterior distribution. A thousand walkers were selected considering the available computational resources at the time of the analysis.

The MCMC simulation for each calibration schemes results in 2×10^6 posterior samples of the model parameters. These samples are then further post-processed to remove the initialization bias and to reduce the autocorrelation among successive iterations.

5.5.1.6 Evaluating Calibration Results

The results of the MCMC simulations are sets of samples directly drawn from the respective posteriors. These multivariate samples are visually represented as *corner plot* depicting the joint posterior samples as a set of 1-dimensional (univariate) and 2-dimensional (bivariate) marginals of the posterior distribution (see Section 5.5.3). From the univariate marginal of each model parameter posterior uncertainties, the constraining ability of the data and the calibration scheme can be quickly, if not rigorously, assessed. From the bivariate marginals, the correlation structure between the model parameters, if any, can also be quickly assessed.

To investigate the implication of the different calibration schemes on the TRACE predictions, simulation campaigns of FEBA test No. 216 are conducted using samples from the posterior PDFs. Furthermore, to assess if the posterior uncertainties are applicable for the simulations of reflood experiments with different boundary conditions, the simulation campaigns are extended to the five additional FEBA tests. In other words, these five additional FEBA tests becomes the validation data sets. Finally, the campaigns are conducted both

²¹ Such as the approach adopted in the more established WinBugs [240], Jags [241], and Stan [242]. These samplers, however, has more extensive capabilities for conducting a Bayesian data analysis and tends to be faster as they port the user-specified probabilistic models to a lower level language (e.g., C++). Furthermore, being older, they have a larger and more diverse user base.

with and without considering the correlation structure in the posterior samples to investigate the effect of the model parameter correlations.

The uncertainty propagation campaigns therefore consists of the campaigns on each FEBA test using model parameters posterior uncertainties derived from different calibration schemes with and without consideration of the correlation among model parameters. For each of these campaign, actual TRACE runs are carried out using 1'000 posterior samples. These samples are directly drawn from the pool of posterior samples obtained from different calibration schemes. The results of the propagation are represented in series of plots of prediction with the associated uncertainty bands for the three output types (TC, DP, and CO) similar to the ones presented in the result section of Chapter 2. From these plots the different propagation campaigns can be compared.

At the same time, the numerous plots are unwieldy to deal with. To circumvent this issue, a more quantitative means of aggregating the results of different prediction uncertainties is required. Although formal Bayesian approaches are available to assess the quality of the prediction using the model parameters posterior uncertainty²², this thesis adopts a more pragmatic assessment method based on the possibilistic theory proposed in [246].

The aim of the method is to quickly compare the applicability of the different posterior uncertainties in making prediction. Loosely speaking, the applicability is measured by the width of the prediction uncertainty as well as its coherence (in the most general sense of the word) with experimental data. The method was applied to synthesize the results of different participants in the context of benchmarking, namely the BEMUSE [246] and the PREMIUM [21] projects. The method consists of three steps: *information modeling*, *information evaluation*, and *information synthesis*. For the comparison purpose in this thesis, only the first two steps above are applied and discussed in the following.

In the information modeling, the information in a prediction uncertainty of a QoI Y is represented by an interval (lower and upper uncertainty bounds: LUB_Y and UUB_Y , respectively) and a reference value y_{ref} . A *source* of information src for a particular QoI Y consists of such an interval and, optionally, a reference value. If for the QoI Y only the bounds LUB_Y and UUB_Y are supplied then a rectangular model can be defined,

Rectangular model

$$\pi_{src,Y}(y) = \begin{cases} 1.0, & LUB_Y \leq y \leq UUB_Y \\ 0.0, & \text{otherwise} \end{cases} \quad (5.64)$$

²² Formal computer model validation metrics, Bayesian or otherwise, is a research topic in its own right, see for instance the validation metrics proposed in [243–245]. Their application is outside the scope of this thesis.

where π is the possibility measure, whose minimum and maximum are 0.0 and 1.0, respectively.

Triangular model The presence of a reference value $y_{\text{ref.}}$ between two bounds LUB_Y and UUB_Y allows the triangular model to be defined,

$$\pi_{\text{src},Y}(y) = \begin{cases} \frac{y - \text{LUB}_Y}{y_{\text{ref.}} - \text{LUB}_Y}, & \text{LUB}_Y \leq y < y_{\text{ref.}} \\ \frac{\text{UUB}_Y - y}{\text{UUB}_Y - y_{\text{ref.}}}, & y_{\text{ref.}} \leq y \leq \text{UUB}_Y \\ 0.0, & \text{otherwise} \end{cases} \quad (5.65)$$

These two information models are illustrated in Fig. 5.14.

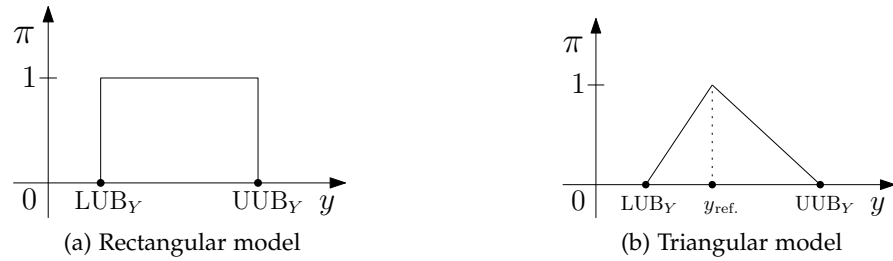


Figure 5.14: Information modeling to represent uncertainty propagation results for a QoI Y .

Ignorance In the following, the rectangular model is used to represent a complete ignorance with respect to the QoI Y , $\pi_{\text{ign},Y}$ using a minimum lower uncertainty bound $\text{LUB}_{Y,\text{min.}}$ and a maximum upper uncertainty bound $\text{UUB}_{Y,\text{max.}}$. That is,

$$\pi_{\text{ign},Y}(y) = \begin{cases} 1.0, & \text{LUB}_{Y,\text{min.}} \leq y \leq \text{UUB}_{Y,\text{max.}} \\ 0.0, & \text{otherwise} \end{cases} \quad (5.66)$$

Informativeness The information evaluation part of the method comprises two indices to evaluate the quality of information in a given source. *Informativeness*, associated with a source src and a QoI Y is defined as,

$$\text{Inf}_Y(\text{src}) = \frac{|\pi_{\text{ign},Y}| - |\pi_{\text{src},Y}|}{|\pi_{\text{ign},Y}|} = 1 - \frac{1}{2} \frac{\text{UUB}_Y - \text{LUB}_Y}{\text{UUB}_{Y,\text{max.}} - \text{LUB}_{Y,\text{min.}}} \quad (5.67)$$

where $|\circ|$ denotes the area under an information model. Informativeness measures the precision of the uncertain prediction, regardless of the position of the reference value within the bound; it takes value between 0.5 (the widest uncertainty range of a source, the same as the maximum and minimum bounds) and 1.0 (the narrowest uncertainty range of a source, practically 0). Fig. 5.15 illustrates Inf calculation.

Calibration score Calibration score between a source src and the observed value $y_{\text{obs.}}$ for a QoI is defined as,

$$\text{Cal}_Y(\text{src}) = \pi_{\text{src},Y}(y_{\text{obs.}}) \quad (5.68)$$

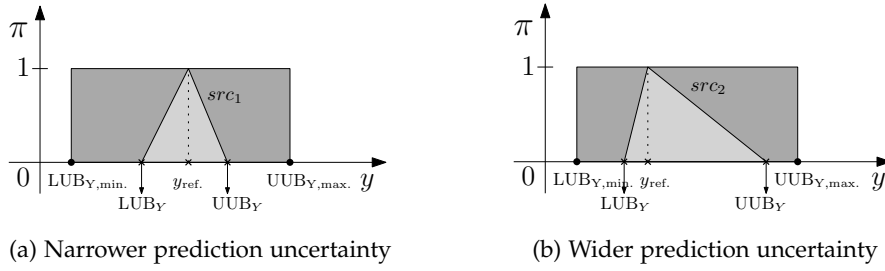


Figure 5.15: Informatives of two different information sources. Wider uncertainty interval gives lower informativeness, and vice versa. Thus $\text{Inf}_Y(\text{src}_1) > \text{Inf}_Y(\text{src}_2)$

that is, the calibration score is the possibility of the observed value y_{obs} under the information model of the source src . It measures the discrepancy between the observed value and the uncertain prediction represented by an interval and a reference value. Following Eq. (5.65), the score severely penalizes the observed data that falls outside the prediction interval; assigning calibration score of 0.0. The score is at maximum of 1.0 for $y_{\text{obs}} = y_{\text{ref}}$. It does not, however, takes into account the possible uncertainties associated with y_{obs} .

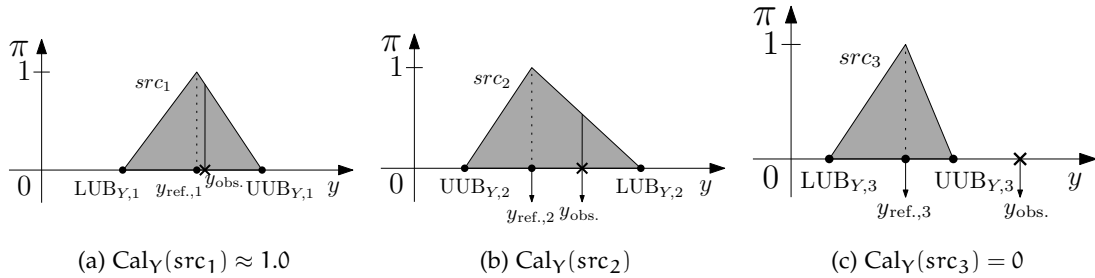


Figure 5.16: Calibration scores of three different sources with the same observed data y_{obs} : $\text{Cal}_Y(\text{src}_1) > \text{Cal}_Y(\text{src}_2) > \text{Cal}_Y(\text{src}_3)$

For multiple QoIs $Y = [Y_1, \dots, Y_D]$ of the same source, both informativeness and calibration score can be aggregated by,

$$\text{Inf}_Y(\text{src}) = \frac{1}{D} \sum_{d=1}^D \text{Inf}_{Y_d}(\text{src}) \quad \text{Cal}_Y(\text{src}) = \frac{1}{D} \sum_{d=1}^D \text{Cal}_{Y_d}(\text{src}) \tag{5.69}$$

Translating from the Bayesian (probabilistic) framework, the bounds of the prediction interval of the above LUB and UUB can be taken to be two percentiles of the prediction probability distribution that cover a selected probability. For instance, the criteria used in the present study, selecting a symmetric 95% probability interval implies the LUB and UUB to be the 2.5-th and 97.5-th percentiles, respec-

tively. The choice is rather arbitrary but as long as the criteria is applied consistently across different prediction uncertainties, the value of the method for comparison is preserved. Furthermore, the 2.5-th and 97.5-th percentiles of the prior prediction distribution are taken to be the LUB_{\min} and UUB_{\max} and the reference value y_{ref} is taken to be the median value of each posterior prediction distribution. Finally the experimental data is taken to be the observed value y_{obs} without considering the associated uncertainty following the original paper [246]. Once again, this lack of consideration is less of an issue for comparing between different uncertain predictions.

5.5.2 MCMC Convergence

The calibration schemes explained above were each run for a total of 2'000 iterations using 1'000 walkers. This results in a total of 2'000'000 posterior samples. These initial samples require further post-processing to remove the initialization bias and the autocorrelation in the samples. In the following, only the results from the calibration scheme with model bias term and considering all types of output (scheme w/ Bias, All) are discussed. Though not shown, the results from other schemes are similar.

Fig. 5.17 shows the trace plots for each of the 8 model parameters in the calibration scheme w/ Bias, All. To avoid over-plotting, the plot only shows the trajectories for the last 100 iterations (out of 1'240 post-burn-in iterations) and for 400 walkers (out of 1'000 walkers). As can be seen, the walkers traverse the model parameter space and spend more time during the iterations in the region where the values of the model parameters allows the simulator to best reproduce the experimental data (thus the region becomes darker in the plots). Furthermore, it can be inferred that some parameters are more constrained by the data (e.g., $gridHT$) than the others (e.g., $tQuench$).

To check the convergence of an ensemble samplers, it is a common practice to investigate the running statistics of the ensemble (i.e., statistics over all walkers per iteration) instead of the individual walkers [109, 232]. The running average and standard deviation for each model parameter are shown in Fig. 5.18. From the figure, it is clear that after some initial transient (i.e., the burn-in period), the running statistics converge for all parameters. Note also that the number of iterations spent in the burn-in period for an ensemble sampler can be large (here up to 760 iterations).

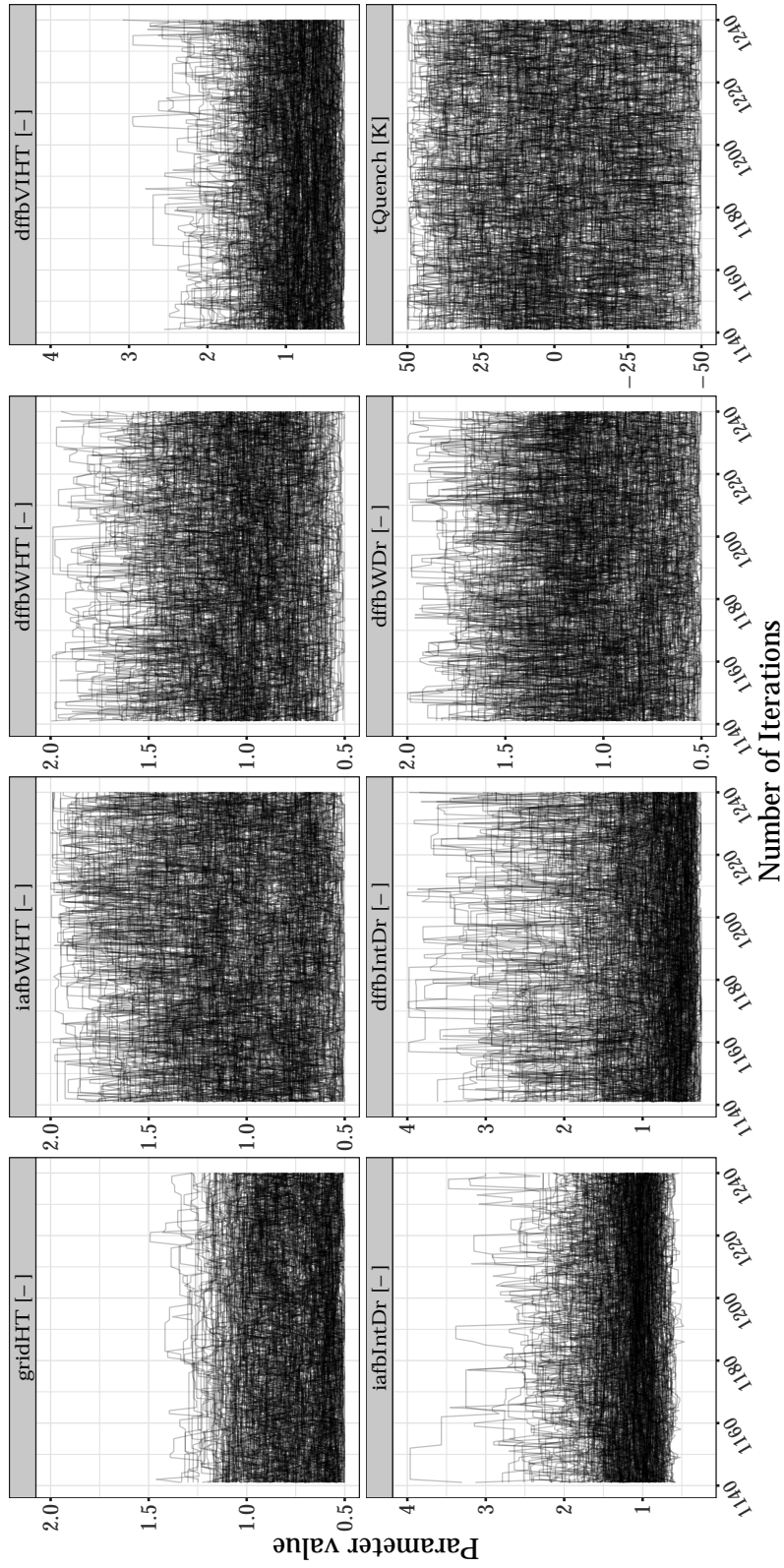


Figure 5-17: Ensemble trace plots for each model parameter of calibration with model bias term. Shown here are the last 100 iterations (out of 1'240 post-burn-in iterations) of 400 walkers (out of 1'000 walkers).

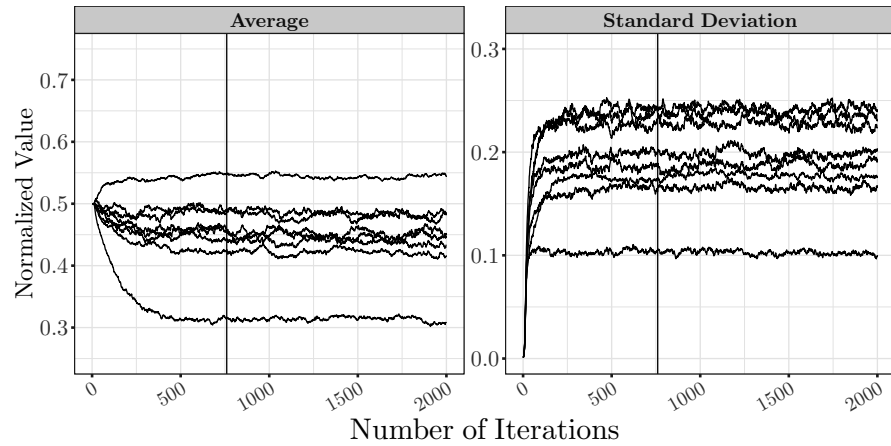


Figure 5.18: Ensemble average and standard deviation as function of the number of iterations for calibration with model bias term. Vertical lines indicate the iterations for burn-in (i.e., approximately 20 times the autocorrelation time).

Although the length of the burn-in period can be inferred directly from Fig. 5.18, a more rigorous criteria can be obtained via the autocorrelation time of the running statistics. Table 5.2 summarizes the estimated autocorrelation time for the running statistics and for each model parameter. The autocorrelation times of the average tend to be longer than the ones of the standard deviation. The longest autocorrelation time of all the parameters (shown in the table in bold) becomes the basis for determining the length of the burn-in period. As recommended in [219] a multiple (in this case 20) of the autocorrelation time is deemed enough to remove the initialization bias. The obtained length of the burn-in period is shown as vertical lines in Fig. 5.18 (at iteration 760); those iterations are subsequently discarded.

After such period, the autocorrelation time is re-estimated to assess if the sampler faces difficulty in sampling the posterior PDF. A particularly long autocorrelation time, even after burn-in, gives an indication of a sampler that is trapped in a particular region of the parameter space [231, 232] and thus requires longer iteration to have representative samples. As can be seen in Table 5.2 ($\tau_{\text{post-burn-in}}$) the times are smaller after burn-in.

Additionally, the remaining samples are to be used for forward UQ. The MC simulation for forward UQ requires independent (specifically, iid) samples. To obtain a set of independent samples, the remaining samples are thinned on the basis of the re-estimated autocorrelation time (see Section 5.4.1. The largest autocorrelation time of all parameters is used for thinning and 32'000 independent posterior samples are obtained for forward uncertainty propagation. Though it results in a much smaller sample size, the associated statistical error relative to the standard deviation of the model parameter estimates are at most 0.6% (Eq. (5.48)).

Table 5.2: Estimated autocorrelation times for the eight model parameters with respect to the ensemble running average and standard deviation, for the calibration scheme w/ Bias, All. The bold term indicates the largest autocorrelation time used to determine the length of burn-in period.

No.	Parameter	Average		Standard Deviation	
		$\tau_{\text{pre-burn-in}}$	$\tau_{\text{post-burn-in}}$	$\tau_{\text{pre-burn-in}}$	$\tau_{\text{post-burn-in}}$
1	gridHT	32.2	3.4	15.1	13.3
2	iafbWHT	15.9	11.3	12.2	10.9
3	dffbwHT	25.6	15.0	13.4	8.4
4	dffVIHT	33.2	11.2	14.1	7.2
5	iafbIntDr	28.6	19.4	10.7	8.0
6	dffIntDr	37.8	8.8	14.3	11.4
7	dffbwDr	13.6	9.4	25.1	3.3
8	tQuench	26.7	6.6	14.1	8.0

5.5.3 Calibration Results

For each calibration scheme, the posterior samples can be analyzed to investigate the constraining power of the data and the possible correlation structure of the parameters. The posterior samples of the calibration scheme w/ Bias, All is presented in this section, while the graphical representation of the posterior samples from the other calibration schemes can be found in Appendix B.6.

The results of calibrating the 8-parameter model with the calibration scheme w/ Bias, All is presented in a corner plot shown in Fig. 5.19. A corner plot [247] depicts the univariate (1-dimensional) and the bivariate (2-dimensional) marginals of the posterior samples and it provides information on the possible correlation structures between pairs of model parameters. That is, it projects the multi-dimensional posterior distribution into each of the 1-dimensional and 2-dimensional subspaces (see the illustration for Gaussian marginals in Chapter 4).

The univariate marginals are shown as the diagonal elements of the plot. Solid lines indicate the 95% symmetric credible intervals computed from the univariate samples²³, while dashed and dotted lines indicate nominal parameter values and posterior median parameter values, respectively. Note that the range for each of the model parameters in the plot corresponds to the respective prior uncertainty range.

Corner plot

*Corner plot,
diagonal element,
univariate marginal*

²³ That is, the interval between the 2.5-th and the 97.5-th percentiles of the posterior samples.

*Corner plot,
off-diagonal element,
bivariate marginal*

The bivariate marginals of the posterior distribution are shown as the off-diagonal elements of the plot. Because correlation is symmetric, only the lower half portion of the plot is shown. In this adaptation of corner plot, hexagonal binning [248] is used to represent the large number of posterior samples. In the off-diagonal elements of Fig. 5.19, lighter color shading indicates the region of the parameter space that is denser with sampled points. The correlation between each pairs of parameters can be preliminary inferred from the shape of the bivariate marginals while the exact number for the color scale is unimportant.

In Fig. 5.19, the most constrained parameters (from either side of the prior range) are the `iafbIntDr`, `dffbIntDr`, `dffbVIHT`, `dffbWHT`. Some parameters are mostly constrained on one side (most notably `gridHT`), while `tQuench` is the least constrained by the data and the calibration scheme.

Although most pairs are largely uncorrelated, the parameter `dffbIntDr` is strongly correlated with `dffbWHT`, `dffbVIHT`, and `iafbIntDr`. The parameter `iafbWHT` is also correlated with `tQuench`. Because of the strong correlation between `dffbVIHT` and `dffbIntDr`, the calibration scheme w/ Bias, no `dffbVIHT` was conducted to investigate the effect. Some correlations (like the one between `dffbWHT` and `dffbVIHT`) are approximately elliptical while the others (like the one between `iafbIntDr` and `dffbIntDr`) appear more nonlinear. However, for most of the strongest correlated parameters, regions of high sample density can be identified. This, in turn, implies that the posterior parameters values that are consistent with the experimental data are largely contained within a small, bounded region of the parameter space, much smaller than the prior parameter space.

Table 5.3 summarizes the prior and posteriors model parameters uncertainties. The posteriors presented are from all the calibration schemes considered. The three numbers inside the brackets correspond to the 2.5-th percentile, the median (for the prior also the the nominal values), and the lower 97.5-th percentile, respectively. That is, the interval constructed by the percentiles corresponds to the symmetric 95% credible interval covering 95% probability.

From the table, it can be seen that considering additional outputs in the calibration tends to constrain even more the posterior range of the parameters (see also Figs. B.21, B.22, and B.23). Furthermore, the calibration scheme in which the parameter `dffbVIHT` was excluded tends to have a tighter posterior uncertainty range for the parameters that were correlated with the parameter `dffbVIHT` (i.e., `dffbIntDr` and `iafbIntDr`), while the range for the `dffbVIHT` remains at its initial prior range (see Fig. B.24). Finally, the posterior of the calibration scheme without model bias term tends to have a tight range and to be concentrated at either end of the prior range. Additionally, the median posterior values of the parameters are shifted far away from

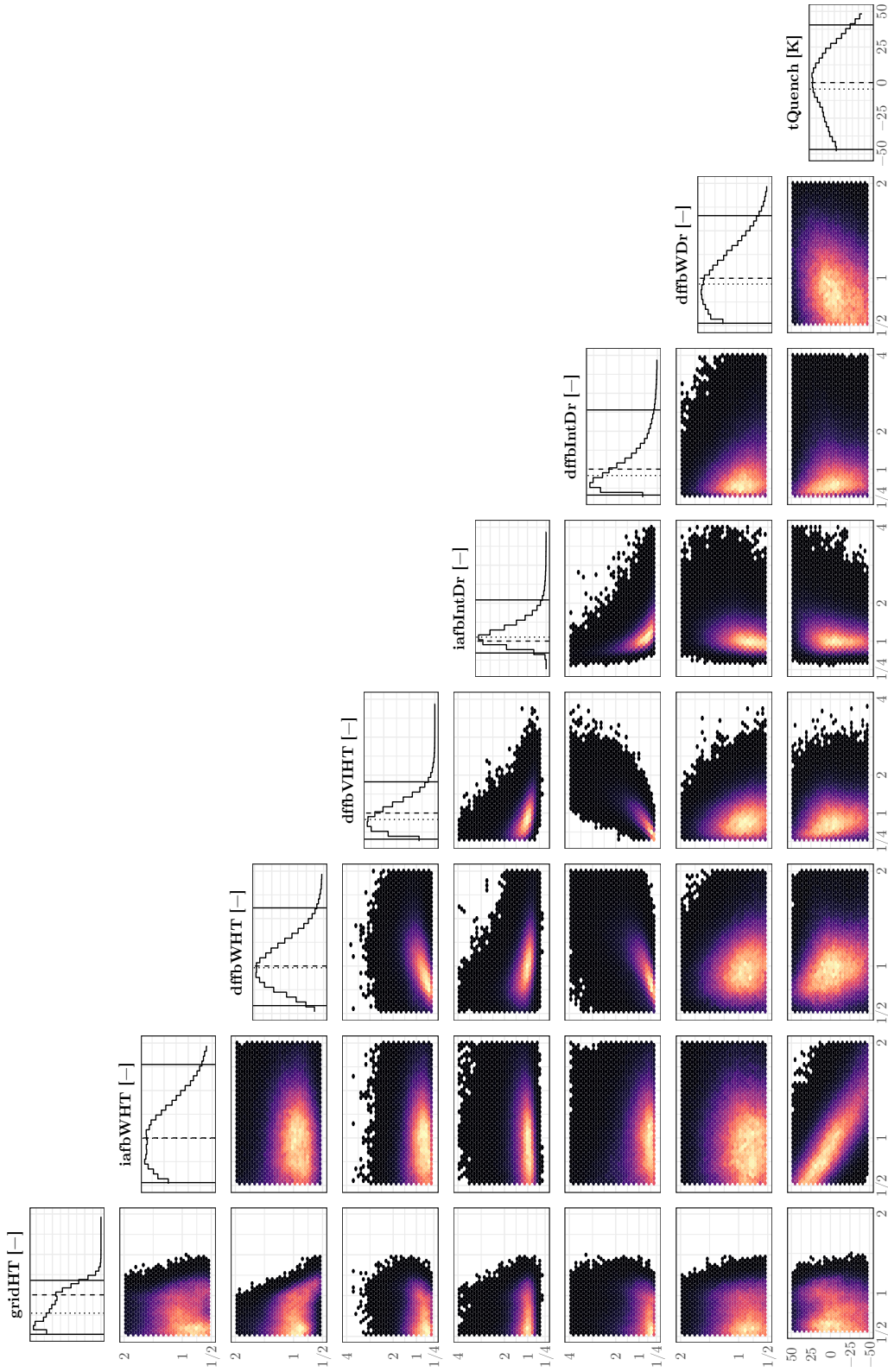


Figure 5.19: Univariate and bivariate marginals of the posterior samples for each of the 8 model parameters. Solid, dashed, and dotted lines indicate the credible intervals, the nominal parameter values, and the posterior median parameter values. Calibration with model bias term.

Table 5.3: Summary of calibration results. The three numbers in brackets are the lower 95% credible interval, the median, and the upper 95% credible interval, respectively.

No.	Parameter	Prior	Posterior Summaries						
			ID	Summaries	w/ Bias, All	w/ Bias, TC	w/ Bias, DP	w/ Bias, CO	w/ Bias, no dffbVIHT
1	gridHT	[0.52, 1.00, 1.93]		[0.51, 0.77, 1.18]	[0.51, 0.78, 1.28]	[0.52, 0.92, 1.82]	[0.52, 0.95, 1.91]	[0.51, 0.80, 1.18]	[0.50, 0.94, 1.06]
2	iafbWHT	[0.52, 1.00, 1.93]		[0.53, 1.00, 1.77]	[0.52, 0.92, 1.83]	[0.53, 1.06, 1.92]	[0.52, 0.97, 1.92]	[0.53, 1.02, 1.78]	[0.50, 0.52, 1.99]
3	dfbWHT	[0.52, 1.00, 1.93]		[0.58, 0.98, 1.61]	[0.56, 0.97, 1.71]	[0.51, 0.85, 1.88]	[0.52, 0.95, 1.93]	[0.62, 1.06, 1.64]	[0.50, 0.52, 0.71]
4	dfbVIHT	[0.27, 1.00, 3.73]		[0.31, 0.83, 1.82]	[0.30, 0.90, 2.10]	[0.27, 0.92, 3.39]	[0.27, 0.75, 3.10]	[0.27, 1.00, 3.73]	[3.36, 3.90, 4.00]
5	iafbIntDr	[0.27, 1.00, 3.73]		[0.69, 1.10, 2.09]	[0.36, 1.27, 3.75]	[0.61, 1.11, 3.09]	[0.27, 1.03, 3.74]	[0.70, 1.02, 1.76]	[0.46, 0.57, 0.71]
6	dfbIntDr	[0.27, 1.00, 3.73]		[0.32, 0.83, 2.56]	[0.30, 0.97, 3.33]	[0.31, 1.17, 3.62]	[0.30, 1.22, 3.74]	[0.57, 1.03, 1.96]	[0.39, 0.96, 2.23]
7	dfbWDr	[0.52, 1.00, 1.93]		[0.53, 0.94, 1.66]	[0.52, 1.01, 1.93]	[0.53, 0.97, 1.72]	[0.52, 1.01, 1.93]	[0.52, 0.92, 1.62]	[0.50, 0.52, 0.62]
8	tQuench	[-47.5, 0.0, 47.5]		[-47, -4, 6, 40.5]	[-47.8, -8, 42.5]	[-47, -3.5, 44.8]	[-47.6, -1.9, 47.2]	[-47.2, -7.7, 36.9]	[-49.7, 48.4, 50.]

their initial nominal values. Most of the initial nominal values actually fall outside the 95% credible intervals (see Fig. B.25).

5.5.4 Calibration Evaluation

The implication of the model parameters posterior uncertainty on the prediction is investigated by propagating the uncertainty through the TRACE simulations of FEBA test No. 216 (the calibration data) as well as the other five FEBA tests. Samples of size 1'000 are picked directly from the joint posterior samples and are used to execute the TRACE simulations. Furthermore, the uncertainties related to the boundary conditions (4 additional parameters, namely *breakP*, *fillV*, *fillT*, and *pwr*) are also propagated alongside the posterior samples from each calibration scheme. Finally, for the selected calibration schemes, the uncertainty propagation is also conducted without taking into account the correlation structure of the model parameters posterior uncertainties. In other words, only the information from the posterior univariate marginals is used for the propagation and the parameters are considered independent of each other. However, the bias term used in some of the calibration schemes is not included in the propagation for the present comparison purpose.

Figs. 5.20, 5.21, and 5.22 show the propagation of the uncertainties for the clad temperature (TC), the pressure drop (DP), and the liquid carryover (CO) outputs, respectively. The model parameters posterior uncertainties used in these figures are the ones obtained from the calibration scheme with model bias term and considering all types of output (i.e., *w/ Bias, All* in Table 5.1). The dark gray band corresponds to the model parameters prior uncertainties propagation, while the two lighter bands correspond to the posterior uncertainties, with and without taking into account the correlation structure of the posterior samples. Finally, solid lines, dashed lines, and crosses correspond to the simulation with the nominal parameters values, the median of the posterior runs, and the experimental data, respectively.

Fig. 5.20 shows the uncertainty propagation for the time-dependent TC outputs at all axial levels with the posterior samples generated by the calibration scheme *w/ Bias, All*. The posterior uncertainties of the clad temperature prediction are narrower as compared to the prior uncertainties across all axial elevations and at all time points. All uncertainty bands, however, show similar behavior regarding their inflation going from the bottom part of the assembly to the top of the assembly, and from the start of the transient to the time of quenching. While it is true that the bias term is not included in the propagation, there is also an apparent “rigidity” associated with the TRACE re-flood curve which cannot be arbitrarily bend. This is well illustrated by the panel of TC2 in Fig. 5.20 which shows that having the uncertainty band to envelop the experimental data points in the early phase

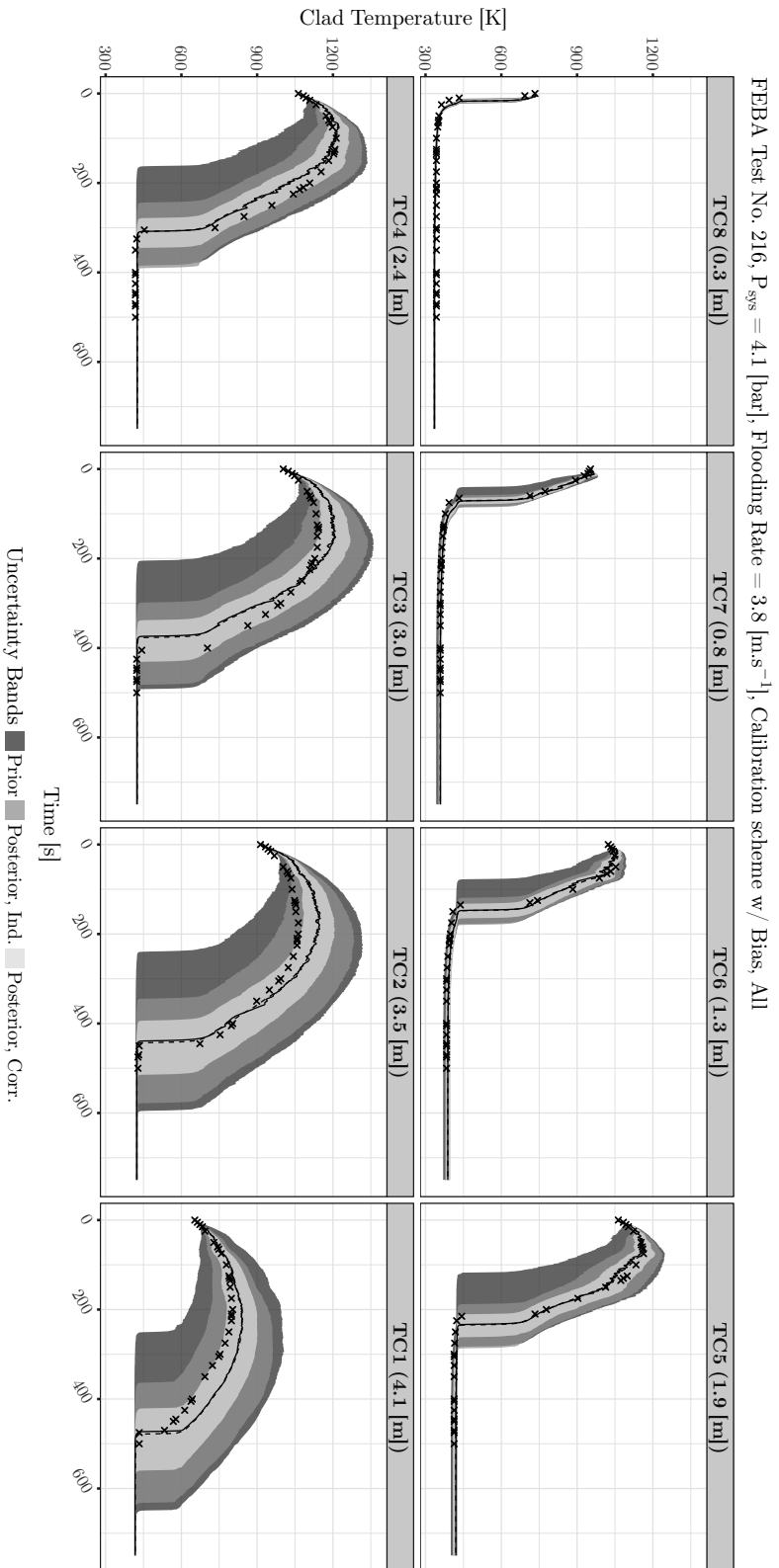


Figure 5.20: Propagation of the model parameters uncertainty on FEBA test No. 216 for the clad temperature output (TC) at different axial locations. The uncertainty bands refer to the symmetric 95% probabilities. Solid lines, dashed lines, and crosses indicate the simulation with the nominal parameters values, the median of the posterior, and the experimental data, respectively. The posterior samples are from the calibration with model bias term and considering all types of output (w/ Bias, All).

of the transient would require a realization of the reflood curve that increases the discrepancy in the later phase of the transient, especially near the time of quenching. Lastly, the median of the posterior predictions (dashed lines) coincides almost perfectly with the prediction of the nominal TRACE run (solid lines).

Not considering the correlation between model parameters in the posterior samples results in wider uncertainties in the clad temperature prediction. While the prediction lower uncertainty bound in this case is much narrower than that of the prior, the prediction upper uncertainty bound is closer to the upper uncertainty bound of the prior. That is, the prediction upper bound is less constrained.

Fig. 5.21 shows the uncertainty propagation for the time-dependent DP outputs for each of the axial segments. The posterior samples correspond to the calibration scheme w/ Bias, All. Once again the posterior uncertainties propagation result in narrower uncertainty bands as compared to that of the prior. However, the difference between taking and not taking into account correlation between model parameters is less striking for this type of output. Moreover, all uncertainty bands cover most of the experimental data.

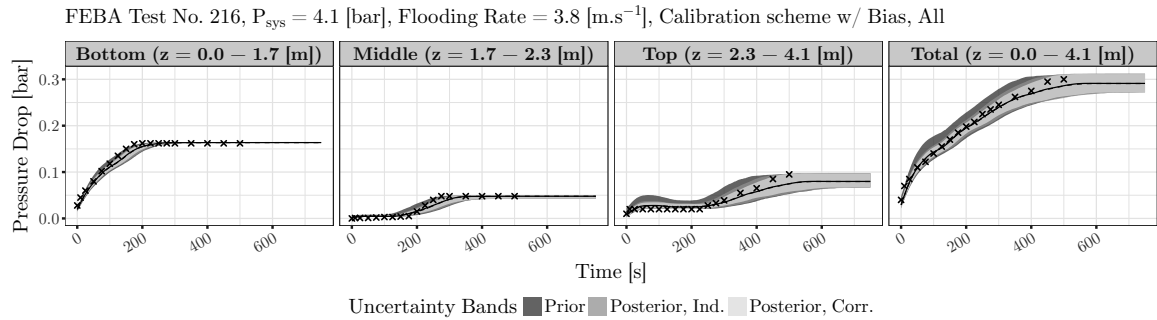


Figure 5.21: Propagation of the model parameters uncertainty on FEBA test No. 216 for the pressure drop output (DP) at different axial segments. The uncertainty bands refer to the symmetric 95% probabilities. Solid lines, dashed lines, and crosses indicate the simulation with the nominal parameters values, the median of the posterior, and the experimental data, respectively. The posterior samples are from the calibration scheme w/ Bias, All.

Fig. 5.22 shows the propagation for the time-dependent CO output up to the saturation of the measurement tank at 10 [kg] with the posterior samples corresponding to the calibration scheme w/ Bias, All. Unlike the previous two types of output, the nominal TRACE prediction exhibits a large bias compared to the experimental data. While the large prior uncertainty manages to cover the experimental data points, all of the points fall outside the posterior uncertainty bounds both with and without taking into account correlation among parameters. As shown in Section 2.6, the TRACE prediction for this particular output was shown exhibited strong bias. Hence, because

the bias term is not included in the propagation, the parameters uncertainty cannot cover the initially large discrepancy due to the bias.

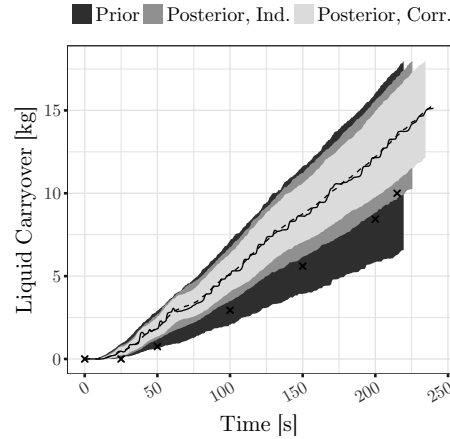


Figure 5.22: Propagation of the model parameters uncertainty on FEBA test No. 216 for the liquid carryover output (CO). The uncertainty bands refer to the symmetric 95% probabilities. Solid lines, dashed lines, and crosses indicate the simulation with the nominal parameters values, the median of the posterior, and the experimental data, respectively. The posterior samples are from the calibration scheme w/ Bias, ALL.

Similar plots for the propagation of the model parameters posterior uncertainties obtained from all calibration schemes are presented in Appendix B.7. The appendix includes the propagation on all the FEBA tests. Fig. 5.23 summarizes the effect of the uncertainty propagation by plotting the *calibration score* vs. *informativeness* (see Section 5.5.1.6).

In each panel, the vertical lines correspond to the informativeness of the prior relative to a rectangular (i.e., representing a state of ignorance) model, which is 0.5 uniformly across output types and FEBA tests. The horizontal lines correspond to the calibration score of the prior uncertainty bands and the nominal TRACE run as its reference simulation value. As can be seen the scores are slightly different from test to test and from output type to output type. Finally, the results of propagating the posterior samples obtained from each calibration scheme to the TRACE FEBA model are plotted. Increasing the informativeness is equivalent to narrowing the uncertainty band; while increasing the calibration scores indicates that the prediction and its uncertainty are closer to the experimental data.

For the TC output, and except for FEBA test No. 216 (the calibration data), there is an apparent linear relationship between calibration score and informativeness. That is, the propagation that results in narrower uncertainty band (high informativeness) tends to have a higher

failure in enveloping the experimental data (low calibration score)²⁴. The results of the scheme w/o Bias, in particular, have among the lowest calibration score with respect to the TC output and the highest informativeness. On the other hand, for the same level of informativeness, the results of the scheme w/ Bias, All have higher calibration scores across all the FEBA tests.

This relationship does not hold for the DP output. There, all the calibration scores fall near the initial calibration score, while having a higher informativeness. Furthermore, there is less variation in informativeness; the points tend to be clustered together especially in comparison with that of TC.

Finally, most calibration schemes result in much lower calibration score with respect to the CO output, some even fall to zero. That is, the resulting uncertainty bands completely fail to envelop a single experimental data points. At the same time, the results of the scheme w/o Bias (and with the exception of FEBA test No. 218), manage to improve the prediction of the output, both in terms of calibration score and informativeness;

Considering correlation between the model parameters in the propagation affects the calibration score and the informativeness. It consistently increases the informativeness (tightening the uncertainty band) and lowers the calibration score across all the FEBA tests. The effect can be observed across outputs, calibration schemes, and FEBA tests; though it is especially strong for the scheme w/ Bias, All and for the TC output.

Comparing results across FEBA tests shows that the informativeness and the calibration score of each calibration scheme remain similar. In particular, the maximum informativeness with respect to the TC, DP, CO outputs are about 0.7, 0.5, 0.5, respectively, for all the FEBA tests. This indicates that the uncertainty band in the prediction due to the posterior uncertainties (obtained on the basis of a single test) are relatively insensitive to the boundary conditions of the tests.

Lastly, for FEBA test No. 216, Fig. 5.23 also shows the results of the calibration schemes with bias where only the TC, DP, CO outputs were considered separately. As expected, these schemes have a lower informativeness than when considering all types of output together. And using only a particular type of output causes the informativeness with respect to the other outputs to be particularly lower. Moreover, the increase in informativeness from taking into account multiple types of output is not followed by a large decrease of the calibration score. This is especially true for the DP (and TC) output where the improvement of the informativeness is significant when compared to the results of calibration using experimental data other than the DP output (and TC output, respectively) itself. This con-

²⁴ Recall that a failure in enveloping experimental data points is assigned to have a zero calibration score.

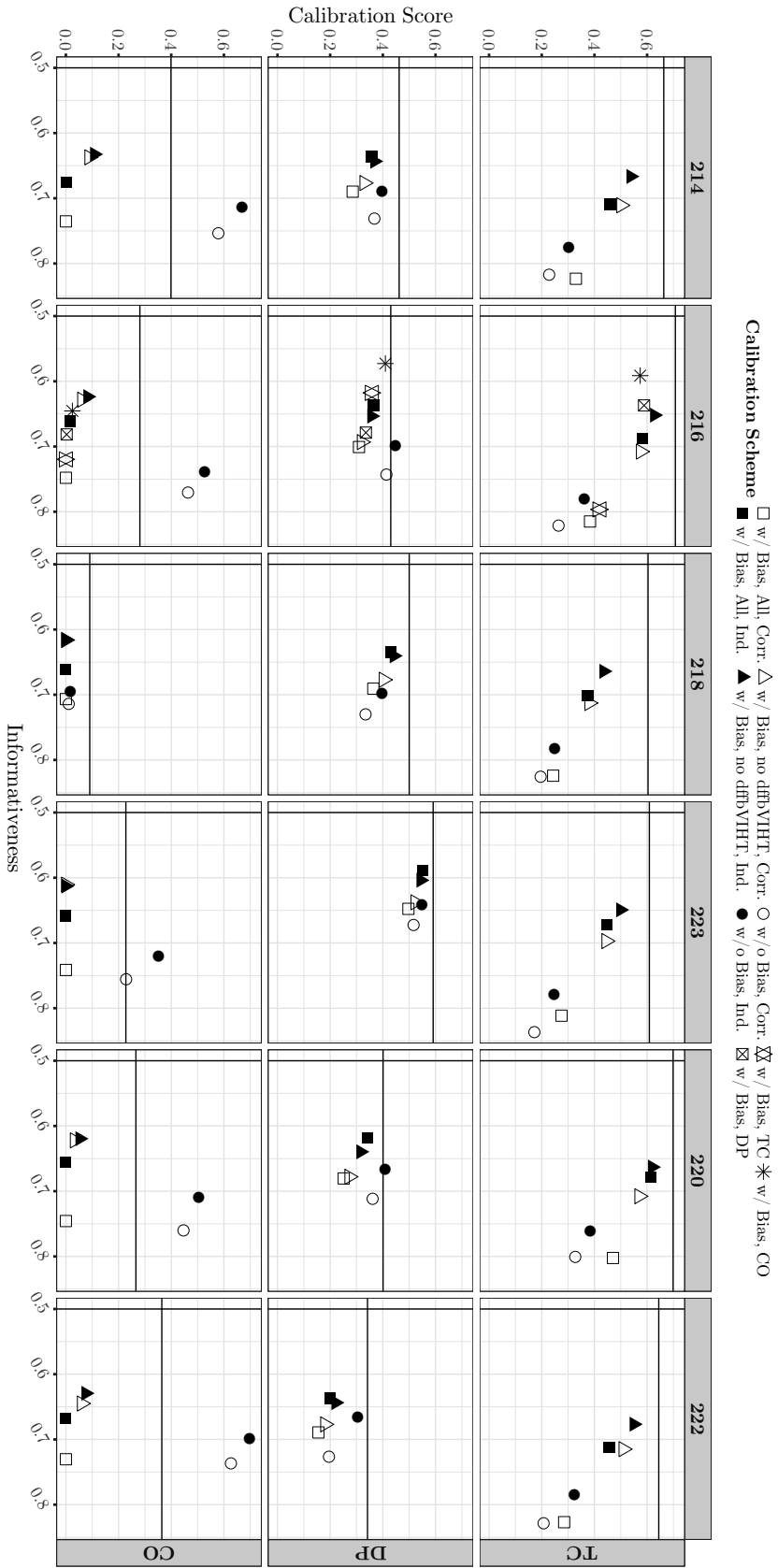


Figure 5.23: Calibration score vs. Informativeness for different posterior samples propagated on all the FEBBA tests. Vertical lines indicate the informativeness of the prior uncertainty (defined as 0.5) while the horizontal lines indicate the initial calibration score (i.e., that of the prior).

firms that considering different model parameters is responsible for the improvement with respect to each output type as was previously showed by sensitivity analysis.

5.5.5 Discussion

5.5.5.1 On the Convergence of the MCMC Simulation

It is not uncommon to run a single particle MCMC simulation up to 100'000 iterations (or beyond) for computer model calibration [80, 249]. In such case, the length of the burn-in period is generally much smaller than the total number of iterations and the corresponding samples do not need to be discarded²⁵.

In the case considered here, the total number of iterations is only 2'000 and the length of the burn-in period was estimated at 40% of the total number of iterations²⁶. The relatively long burn-in period with respect to the total number of ensemble iterations is consistent with the observations of Refs. [109, 231, 232], each of which applied an AIES ensemble sampler to conduct a Bayesian calibration of a computer model. Therefore, determining the burn-in period was indeed mandatory; if the samples associated with this initial transient were not discarded then the model parameter estimates would be heavily biased. Despite discarding a lot of the initial samples, the resulting statistical error associated with each model parameter estimate obtained by the MCMC simulation is less than 1% relative to the true standard deviation of the respective parameter.

Finally, note that the only free parameter to deal with in this particular application of the ensemble sampler was the total number of iterations; no adjustment to the sampler itself during the iteration was required.

5.5.5.2 On the Identifiability of the Model Parameters

The resulting posterior samples, as presented in the corner plots (Figs. 5.19, B.21, B.22, B.23, B.24, and B.25) and as summarized in Table 5.3, demonstrate different constraining ability of the data on the model parameters prior uncertainties depending on the calibration scheme. These calibration results are to be expected according to the sensitivity analysis conducted in Chapter 3.

For instance, the pressure drop output is mainly sensitive to the parameter `iafbIntDr` as shown by Tables B.9–B.12; and the pressure drop data can, in turn, mainly inform the same parameter as demonstrated in Fig. B.22. Meanwhile, although the liquid carryover out-

*Constraining ability
of the data*

²⁵ A rule of thumb argues for discarding at most 20% of the total number of samples in a single particle samplers [219].

²⁶ Note that, despite the lower number of iterations for an ensemble sampler, the computational cost of the sampler in terms of the number of likelihood evaluations is on par with a single particle sampler.

put is sensitive to the model parameters $dffbVIHT$ and $dffbIntDr$ as shown in Tables B.13 and B.20, the importance of the two parameters are eclipsed by the variation in the inlet velocity boundary condition $fillV$ ²⁷. Thus, as indicated in Fig. B.23 the constraining power of the liquid carryover data is fairly limited; the posterior uncertainties associated with the important parameters remain wide.

The results are similar when considering only the clad temperature data for the calibration. The parameters $gridHT$ and $dffbWallHT$ are well constrained by the temperature data. The parameter $gridHT$, in particular, is constrained only at the upper end of the uncertainty bound. From this calibration exercise, it turns out that decreasing the spacer grid heat enhancement below a certain value will not decrease the overall heat transfer any longer. Thus, below that value the enhancement is insensitive with respect to the clad temperature output.

By definition, a model parameter that is not sensitive to a simulation output cannot be informed by the experimental data of that output. That is, the parameter is non-identifiable with respect to that output [250]. Considering other type of output data can potentially solve the problem. And indeed, when considering all types of data, the prior uncertainties of the model parameters was shown to be simultaneously constrained (Fig. 5.19). For instance, the parameters $iafbIntDr$ and $dffbWallHT$ are non-identifiable with respect to the temperature and pressure drop data, respectively (Figs. B.21 and B.22, respectively). But, as shown in Fig. 5.19, the calibration using both types of experimental data solves the non-identifiability problem for both parameters.

There are, however, several parameters that simply could not be constrained by the considered experimental data. The uncertainties, especially the lower bounds of the parameters $iafbWHT$, $dffbWDr$ and $tQuench$ remained close to their initial values, while their upper bounds were only marginally smaller. These parameters were found to be of marginal importance among the selected influential parameters (see Appendices B.2 and B.4). Although it is straightforward to conclude that insensitive parameters simply cannot be constrained by the experimental data, and the most influential ones are strongly constrained, it stays unclear which among the parameters of intermediate importance – as indicated by sensitivity measures of Chapter 3 – can be well constrained by the experimental data.

The calibration results also showed that strong correlation was present among the model parameters. In the case of the calibration against the TC data (Fig. B.21), the parameter $dffbVIHT$ was shown to be correlated with multiple parameters, particularly with the parameter $dffbIntDr$. In the case of the calibration against the DP data (Fig. B.22), the parameters $iafbWHT$ and $tQuench$ were found to be

*Non-identifiability,
multiple types of
data*

*Non-identifiability,
insensitive
parameters*

*Non-identifiability,
correlation*

²⁷ Recall from Section 5.5.1.3 that the variation of the boundary conditions, including that of the inlet velocity $fillV$, is included in the model bias term.

strongly correlated. Considering multiple types of experimental data (Fig. 5.19) did not seem to break these correlations.

Both cases are examples of another form of parameter non-identifiability. Though these parameters were sensitive with respect to the outputs – as the parameters posterior uncertainties were clearly informed and affected by the calibration process –, changes in one of those parameters could be offset by the changes in the other and any of the combinations still reproduce the experimental data. As a result, the univariate posterior marginal uncertainties of these parameters remained large.

This is especially true when comparing the correlation between the parameters *iafbWHT* and *tQuench* and between the parameters *dffbVIHT* and *dffbIntDr*. In the former, due to the correlation over the whole range of both parameters, more precise estimates of either (with respect to the prior) cannot be extracted; while in the latter, the upper bounds of both parameters remained large.

If a more precise estimate of a parameter is of interest then a straightforward solution is to remove an influential parameter that is strongly correlated from the calibration process [251] and to keep it at its prior uncertainty. This approach was investigated in this thesis because of the strong correlation between two important parameters, namely *dffbVIHT* and *dffbIntDr*. The calibration scheme *w/ Bias, no dffbVIHT*, in which the parameter *dffbVIHT* was excluded from the calibration process, further constrained the uncertainty of the parameter *dffbIntDr* (Fig. B.24).

*Calibration,
excluding a
correlated parameter*

5.5.5.3 *On the Calibration with and without the Model Bias Term*

The last calibration scheme investigated was the *w/o Bias* scheme in which no model bias term was incorporated in the calibration process and the only sources of uncertainties were the ones associated with the reported experimental data (see Section 5.5.1.4). The results as presented in Fig. B.25 showed a peculiar behavior. For many model parameters, their nominal values were found to be outside the 95% posterior uncertainty range. Recall that these nominal values were obtained by calibration against experimental data from different SETFs. Therefore, the results imply that the previous calibration results are not able to simulate the reflood experiment of FEBA; and that there are different model parameter values that allow TRACE to simulate the experiment better.

Moreover, the posterior samples of the parameters were concentrated either on one or both sides of the prior uncertainty range. The parameters *iafbWHT* and *tQuench*, for instance, were concentrated on both sides of their prior uncertainty range; while the parameters *dffbWHT* and *dffbVIHT* were concentrated on one side only. The lack of model bias term in the calibration formulation did force the parameters to change dramatically – and at times up to the limit of the

prior uncertainties – in order to compensate any discrepancy between the simulator predictions and the experimental data, beyond the experimental uncertainty. That is, though these distributions might look peculiar, they were the ones found to be consistent with the experimental data under the calibration formulation.

5.5.5.4 On the Propagation of the Posterior Uncertainties

In this section and the next, some aspects of the different calibration schemes and the role of the correlation among model parameters in the posterior samples are discussed in detail. As such, the discussion is focused on the propagation of posterior uncertainties obtained from the calibration schemes *w/ Bias, All, w/o Bias, and w/ Bias, no dffbVIHT* and only for the TC1 (the clad temperature at the top of the assembly) and CO outputs. This choice is motivated by the significant difference observed between the different uncertainty propagation campaigns for these schemes and outputs. The uncertainty propagation of interest are shown in details in Appendix B.7. Some relevant figures to illustrate the discussion are reproduced below.

Fig. 5.24 shows the uncertainty propagation results for the TC1 output. The uncertainty band from the calibration scheme *w/ Bias, All* (and to a lesser extent the one from the scheme *w/ Bias, no dffbVIHT*) shows a consistent shape with the nominal TRACE prediction. The band from the scheme *w/o Bias* is dramatically different (Fig. 5.24b); it bears no resemblance whatsoever with the nominal TRACE prediction. Moreover, the nominal TRACE prediction falls completely outside the uncertainty band.

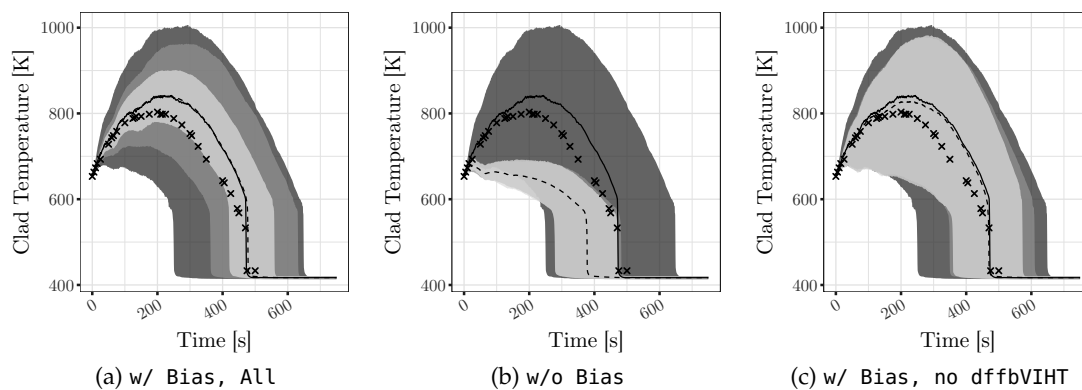


Figure 5.24: Uncertainty propagation results for TC1 output (the clad temperature at the top of the assembly) of FEBA test No. 216 with the posterior of the model parameters from 3 different calibration schemes. The uncertainty bands from darkest to lightest shades correspond to the prior, posterior (independent), and posterior (correlated) model parameters uncertainties, respectively.

Fig. 5.24 also illustrates that the posterior samples from the calibration scheme *w/ Bias, All* are indeed strongly correlated, ignor-

ing the correlation changes significantly the uncertainty bands of the prediction. In essence, the posterior samples are a set of “collectively-fitted” values that are consistent with the calibration data [252]. Within the calibrated (and correlated) region of the parameter space, changes in the model parameter values would not alter the performance of the model against the calibration data. In this illustration, the correlation among parameters tighten the uncertainty band while keeping the experimental data within it. Removing the correlation structure – and using only the posterior range of the univariate marginals – gives realizations that can significantly differ from the calibration data and thus widen the uncertainty band.

Moreover, as it was suspected, the parameter `dffbVIHT` is responsible for the strong correlation; ignoring the correlation structure in the uncertainty propagation of the results from the scheme `w/ Bias`, no `dffbVIHT` shows only a marginal difference for the propagations with and without the correlation structure. And because an important parameter was removed from the calibration – and keeping its uncertainty at the prior range – wider prediction uncertainty bands were produced.

At the same time, allowing the parameter `dffbVIHT` to change dramatically from the initial nominal parameter value also allowed the calibration scheme `w/o Bias` to correct the bias in the prediction of liquid carryover as indicated in Fig. 5.25b. Now the nominal TRACE prediction mostly falls outside the posterior uncertainty band. By construction, the other two schemes allowed the nominal prediction to be within their respective posterior uncertainty bands although the bands did not (or only marginally) cover the experimental data.

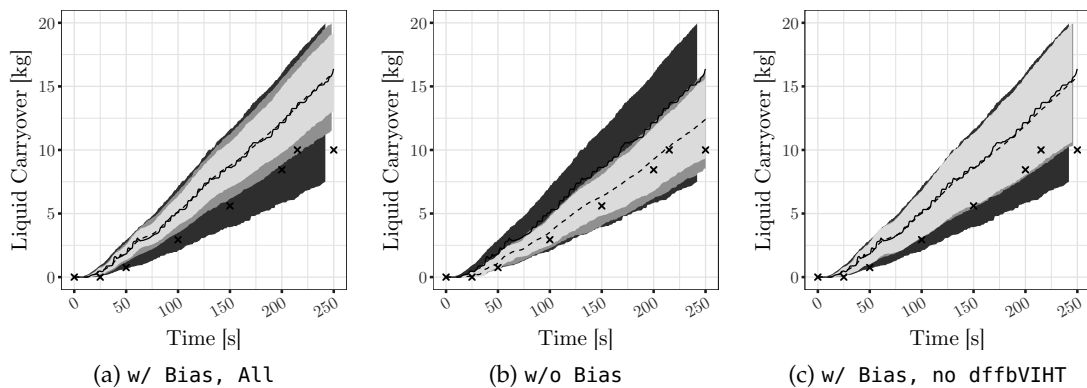


Figure 5.25: Uncertainty propagation results for CO output of FEBA test No. 216 with the posterior of the model parameters from 3 different calibration schemes. The uncertainty bands from darkest to lightest shades correspond to the prior, posterior (independent), and posterior (correlated) model parameters uncertainties, respectively.

This is supported by the sensitivity analysis conducted in Chapter 3 where it was shown that the parameter `dffbVIHT` was important for the CO output and became increasingly important for the TC output

at higher elevations. This can be physically understood as the parameter $dffbVIHT$ is responsible for the heat transfer between the vapor phase and the interface and contributes to the variation in the available entrained liquid and droplets being carried away through the top of the assembly. These droplets, in turn, are an important heat sink for the clad. Thus, an increase in the heat transfer coefficient enhances the heat transfer at the top of the assembly and accelerates the evaporation of the available droplets (thus the lower clad temperature) while simultaneously decreases the amount of liquid being carried away (thus the slower rate of liquid carryover collection).

5.5.5.5 On the Effect of Boundary Conditions

When comparing the uncertainty propagation results of the output TC1 across FEBA tests, another peculiar finding can be observed. Fig. 5.26 shows the uncertainty propagation with respect to the output TC1 for FEBA test Nos. 216, 220, and 222 using the model parameters posterior uncertainties from the calibration scheme w/ Bias, All. It is normal for test No. 216, the test used for the calibration, to be well predicted and covered by the posterior model parameters uncertainties (Fig 5.26a). Surprisingly, the prediction for test No. 220 (Fig 5.26b) looks even better. The experimental data for test No. 222 falls outside the uncertainty band (of the correlated posterior samples), but the shape of the experimental data is very similar to that of the uncertainty band.

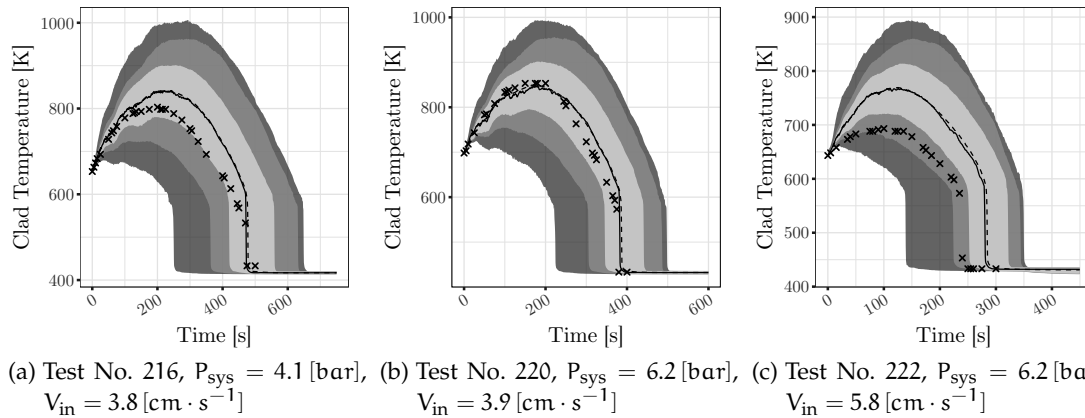


Figure 5.26: Uncertainty propagation results for TC1 output (the clad temperature at the top of the assembly) of FEBA tests No. 216, 220, and 222 with the posterior uncertainties of the model parameters from the calibration scheme w/ Bias, All. The uncertainty bands from darkest to lightest shades correspond to the prior, posterior (independent), and posterior (correlated) model parameters uncertainties, respectively.

On the contrary, the uncertainty propagation with respect to the output TC1 for FEBA test Nos. 216, 220, and 222 using the model parameters posterior uncertainties from the calibration scheme w/o

Bias look significantly off compared with both the experimental data and the nominal TRACE prediction (Fig. 5.27).

In the case of the uncertainty propagation with respect to the output TC1 for FEBA test Nos. 214, 218, and 223, the model parameters posterior uncertainties from the calibration scheme w/ Bias, All produces significantly different results compared with the experimental data (Fig 5.28). The experimental data of test Nos. 218 and 223, in particular, are very much different than the nominal TRACE prediction and the corresponding uncertainty band. Neglecting the correlation structure between the model parameters inflates the uncertainty band and increases the coverage of the experimental data but the experimental data still exhibits a dissimilar transient behavior. This suggests that the observed phenomena might be of a different nature, and that there is a missing physical process in the simulation of re-flood test with lower system pressures (and to some extent higher inlet velocity as indicated by test Nos. 214 and 222). In the case of a very strong bias between the nominal TRACE prediction and the corresponding experimental data, a calibration scheme w/ Bias does not allow to better reproduce the experimental data by making significant adjustments to the model parameters.

And yet, the uncertainty propagation with respect to the output TC1 using the model parameters posterior uncertainties from the calibration scheme w/o Bias do not look terribly off from FEBA test Nos. 214, 218, and 223. While the resulting uncertainty bands still fail to cover most of the experimental data, the upper uncertainty bounds of each test looks conspicuously similar to the experimental data. As mentioned the experimental data of TC1 for these three tests do not exhibit a typical re-flood curve. At the same time, there is a poten-

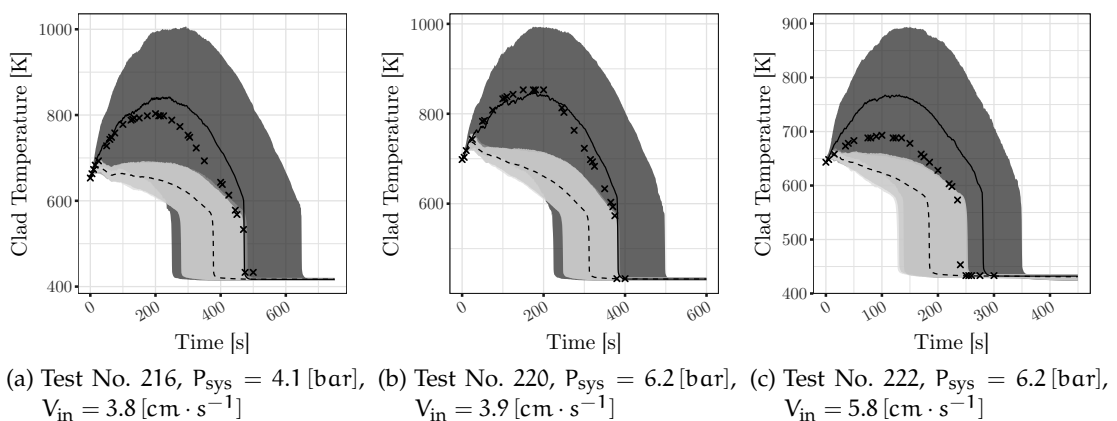


Figure 5.27: Uncertainty propagation results for TC1 output (the clad temperature at the top of the assembly) of FEBA tests No. 216, 220, and 222 with the posterior uncertainties of the model parameters from the calibration scheme w/o Bias. The uncertainty bands from darkest to lightest shades correspond to the prior, posterior (independent), and posterior (correlated) model parameters uncertainties, respectively.

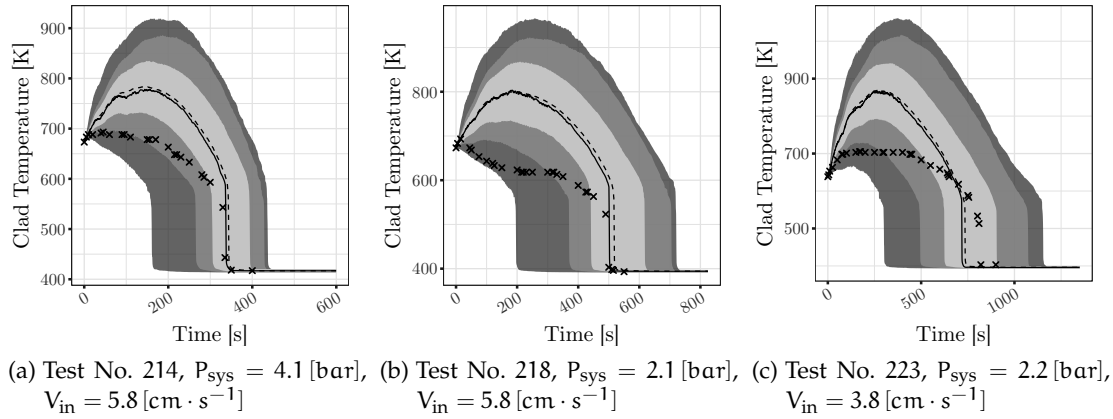


Figure 5.28: Uncertainty propagation results for TC1 output (the clad temperature at the top of the assembly) of FEBA tests No. 214, 218, and 223 with the posterior uncertainties of the model parameters from the calibration scheme w/ Bias, All. The uncertainty bands from darkest to lightest shades correspond to the prior, posterior (independent), and posterior (correlated) model parameters uncertainties, respectively.

tial for TRACE to properly reproduce the data through a significant adjustment of the nominal values of some model parameters (e.g., *df*fbVIHT) as it was allowed by the calibration scheme w/o Bias.

Therefore, it is worth investigating in a further study whether a proper parametrization with respect to system pressure and reflood rate have been implemented in TRACE. This study suggests that a different reflood closure model adjustment might be required to properly simulate reflood in the upper part of a dry assembly, in a low system pressure (here it is 2.1 [bar]) or with higher reflood rate (i.e., test Nos. 214 and 222).

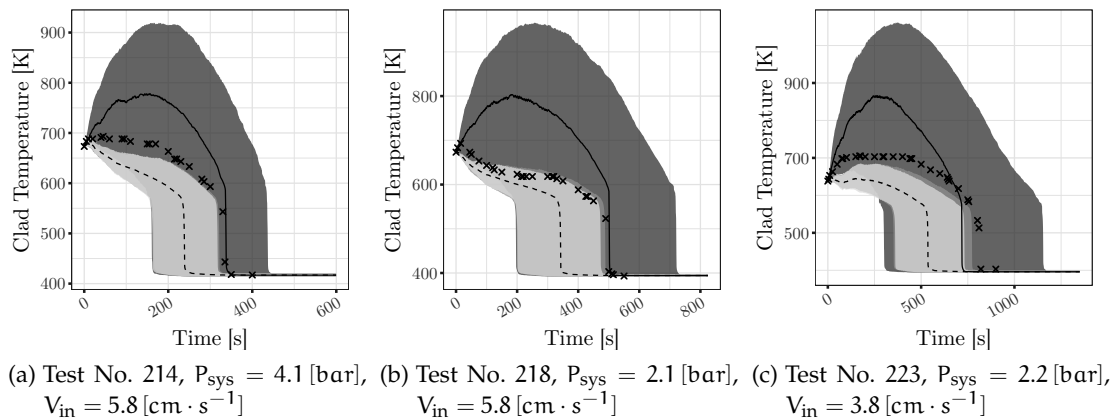


Figure 5.29: Uncertainty propagation results for TC1 output (the clad temperature at the top of the assembly) of FEBA tests No. 214, 218, and 223 with the posterior uncertainties of the model parameters from the calibration scheme w/o Bias.

5.6 CHAPTER SUMMARY

The model calibration part of the proposed statistical framework has been presented in this chapter. The goal of the present chapter was to quantify the a posteriori uncertainty of the model parameters based on experimental data. The quantification followed a Bayesian calibration framework.

The Bayesian calibration framework has been detailed in this chapter and consisted of two parts: the formulation of a probabilistic model for the calibration; and the computation of the formulated model to obtain the posterior uncertainty of the model parameters. A generic calibration formulation was presented along with a description on each of its element. Afterward, the computational aspects of a posterior distribution were presented. MCMC simulation was used in this thesis to directly obtain samples from the posterior, which are useful for the characterization of the posterior uncertainty or for uncertainty propagation.

The calibration framework was applied to the running case study of the simulation of a reflood experiment using TRACE conducted at the FEBA facility (test No. 216). Five calibration schemes that result in five likelihood (thus posterior) formulations were considered. The schemes considered different type of experimental data (TC, DP, CO, or all together), and included (or not) a model bias term in their formulation. The five schemes calibrated the 8 most influential reflood parameters. An additional scheme was introduced to investigate the effect of removing a strongly correlated parameter from the calibration process. The formulated posterior PDFs were then sampled using an implementation of the AIES MCMC ensemble sampler to obtain different sets of posterior samples. Finally, these sets of samples were propagated through all the TRACE models of the FEBA tests with different system backpressure and reflood rate boundary conditions.

The MCMC simulation was shown to converge in exploring the posterior parameter space. The resulting independent samples – readily used for uncertainty propagation – were large enough to yield model parameter estimates with statistical error of less than 1% of the true standard deviation of the respective parameter.

Two types of parameter non-identifiability were encountered during the calibration process: parameter non-identifiability due to insensitivity with respect to a type of data and non-identifiability due to correlation between parameters. The former was solved by simultaneously considering different types of output to which the parameter of interest was sensitive. The latter was more challenging as considering different types of data did not seem to solve the non-identifiability issue (the univariate marginals of the parameter of interest remained large). However, even without precise estimates of each parameter, the correlation structure among model parameters provided a set

of “collectively-fitted” values that was consistent with the calibration data. Specifically, as long as the correlation structure was kept, propagation with parameters with large univariate marginal uncertainties would still produce predictions that were consistent with that data.

This posterior correlation structure *and* the posterior range were specific to the calibration data (here FEBA test No. 216). One of two strongly correlated parameters could be argued to be an extraneous parameter with respect to the calibration data. Its presence would allow many combinations of parameter values to reproduce the data with a tighter prediction uncertainty. However, if the calibration data was deemed not sufficiently large or comprehensive enough then care should be taken to avoid overfitting. Excluding one of the correlated parameters allowed for a more precise estimation of the other parameters that were previously correlated. But at the same time, as the uncertainty of the excluded parameter was kept at its prior, the posterior uncertainty band of the outputs remained relatively wider.

The calibration scheme with model bias term and incorporating all types of outputs was able to constrain the prior uncertainties of the model parameters while keeping the nominal TRACE parameters values within the posterior uncertainty interval. That was in contrast with the results of the calibration scheme without a model bias term, in which the posterior uncertainties were concentrated on either or both sides of the prior range, and at times not including the nominal TRACE parameter values. This results implied that the previous calibration results (i.e., during the model development) were not consistent with the data of FEBA; and that there were different model parameter values that would allow TRACE to better simulate the FEBA experiments. This illustrated the value of incorporating the model bias term in order to avoid overfitting, especially considering the fact that the calibration conducted here was based on one test run only. The calibration scheme without model bias was found slightly more informative (having tighter uncertainty band) but less calibrated (in the sense of the *calibration score*, i.e., having more experimental data points outside the uncertainty band) than the calibration scheme with model bias.

CONCLUSIONS AND FUTURE WORK

The main goal of the present doctoral research was to quantify the uncertainty of physical model parameters implemented in TH system codes and update the uncertainties based on comparison with experimental data. To that end, a methodology has been developed and applied to TRACE models of SETF experiments dedicated to reflood; a relevant phenomenon to consider in the safety analysis of LWRs.

The methodology consisted of three statistical methods for *sensitivity analysis*, *metamodeling*, and *Bayesian calibration*. Starting from the TRACE code modeling of the FEBA facility for reflood experiment and a preliminary selection of uncertain input parameters, a set of sensitivity analysis methods were applied to assess the sensitivity of the code output to each selected input parameter and select the model parameters that were truly important for the reflood simulation. In anticipation of the high computational cost associated with the Bayesian calibration, a Gaussian process (GP) metamodel of the TRACE model of FEBA was then developed and validated. Using the validated metamodel to substitute the TRACE code run, the selected model parameters were calibrated against the experimental data of FEBA which resulted in an a posteriori quantification of the parameters uncertainties. Finally, the quantified uncertainties were verified by means of uncertainty propagation on FEBA tests with boundary conditions different from the conditions of the calibration data.

This final chapter starts with a chapter-wise summary of the thesis, presented in Section 6.1. The main achievements of the thesis are given in Section 6.2, in which corresponding recommendations for future work are proposed.

6.1 CHAPTER-WISE SUMMARY

Chapter 1 introduced the doctoral research through the problem of uncertainty quantification in nuclear engineering TH analysis; both as forward and backward (inverse) problems. The particular problem of inverse uncertainty quantification was then put in the context of the recently concluded OECD/NEA PREMIUM project; a benchmark project comparing different inverse uncertainty quantification methods used in the community. The chapter then presented a set of strategies to eventually quantify the uncertainty, namely sensitivity analysis, statistical metamodeling, and Bayesian calibration. The set

of strategies was consolidated in a statistical framework adapted from the applied statistical literature, on which a review was conducted.

Chapter 2 presented the reflood experiment at the FEBA facility that served as the experimental basis for this work. The FEBA facility was a full-height 5×5 bundle of PWR fuel rod simulator. The particular test series selected corresponded to the case without flow blockage, under three different values of system backpressure boundary condition (2.1, 4.1, and 6.2 [bar]) and two different reflood rates (3.8 and 5.8 [$\text{cm} \cdot \text{s}^{-1}$]). Three types of time series measurement were recorded: clad temperature at eight axial locations, pressure drop at four axial segments, and liquid carryover. The TRACE model of the facility was developed and a set of 27 initial input parameters perceived to be important for the simulation was selected. Thereafter, prior uncertainties of the selected input parameters were assigned and propagated through the TRACE model of FEBA to assess the prior level of prediction uncertainties on all three types of output (data). This model then became the running case study in the three subsequent chapters to which the proposed methods are applied.

Chapter 3 introduced selected global sensitivity analysis (GSA) methods which were applied to the TRACE model of FEBA. First, the importance of the initial set of input parameters was quantitatively assessed through the Morris screening method and the total-effect Sobol' indices. The two provided a basis for parameter screening in which less influential parameters were excluded from further analysis, reducing the size of the problem. After the screening step, only 12 out of the initial 27 input parameters were found to be influential. Focusing on the 12 most influential parameters, the effect of parameter perturbation on the overall time-dependent outputs was investigated. The high-dimensionality of the outputs was reduced by means of techniques derived from FDA. Finally, main- and total-effect Sobol' indices, two global sensitivity measures, were estimated for each parameter with respect to the output in the reduced space. The results regarding parameter sensitivity with respect to different outputs have provided a better understanding of the inputs/outputs relationship in the TRACE model of FEBA.

Chapter 4 detailed the development and validation of a metamodel based on GP to substitute the TRACE model of FEBA. Though a single run of the TRACE model was relatively short ($\approx 6 - 14$ [min]), a large number of runs in the order of hundreds of thousands was expected for the Bayesian calibration. Thus, a computationally efficient metamodel was deemed crucial in the calibration of the model parameters. Built upon the results of the previous chapter, the development was directly focused on the 12 most influential input parameters. The high dimensionality of the output, in time and in space, was dealt with PCA, a linear dimension reduction method. The dimension reduction method was shown to have difficulty in representing the clad

temperature output which exhibited strong discontinuity in the vicinity of the quenching. Yet, the average predictive performance of the metamodel against a test data set of actual TRACE runs was found to be acceptable, especially in comparison to the initial prediction uncertainty due to the prior input parameter uncertainties. The validated metamodel, with a cost of less than 5 [s] per evaluation, was then ready to be used over the prior range of the input parameters in lieu of directly running TRACE.

Chapter 5, the last of the main chapters of the thesis, finally proceeded with the a posteriori quantification of uncertainties of the most influential reflood model parameters on the basis of FEBA test No. 216. Different posteriors PDFs corresponding to different calibration schemes were formulated and directly simulated using an AIES ensemble sampler. Five different calibration schemes having different assumptions were investigated: with or without considering model bias term, incorporating different types of data, and including or excluding a strongly correlated model parameter. Two types of parameter non-identifiability were encountered: parameter non-identifiability due to insensitivity with respect to a type of experimental data and non-identifiability due to correlation between parameters. The former was solved by considering different types of output to which the parameter of interest was sensitive. The latter was more challenging as considering different types of data did not manage to solve the non-identifiability issue (the univariate marginals of the parameters remained large). Excluding one of the correlated parameters did allow for a more precise estimation of the other parameters that were previously correlated. But at the same time, because the excluded parameter kept its large prior uncertainty, the posterior prediction uncertainty band was relatively wider. However, even without precise estimates of each parameter, the correlation structure among model parameters provided a set of “collectively-fitted” values that was consistent with the calibration data. Specifically, as long as the correlation structure was kept, propagation with parameters with large univariate marginal uncertainties would still produce prediction that was consistent with the calibration data.

The results of different calibration schemes corresponded to different level of trade-off between *informativeness* (the width of the prediction uncertainty band) and *calibration score* (consistency with the experimental data and coverage by the uncertainty band). This trade-off was apparent particularly for the schemes with and without model bias term and the scheme with bias but excluding a strongly correlated parameter. The scheme without bias resulted in the largest reduction of the prior uncertainty for most of the important parameters, but in which the nominal TRACE parameter values sometimes laid outside the posterior uncertainty interval. The posterior uncertainties associated with the scheme resulted in predictions with the highest

informativeness and the lowest calibration scores across FEBA tests. The scheme with bias resulted in a more modest reduction of the prior uncertainty of the parameters, keeping the nominal TRACE parameter values within the uncertainty interval, but exhibited strong correlation for some of the parameters. The corresponding prediction uncertainties, in turn, gave a better calibration score while having similar level of informativeness. It could be argued that the relatively worse calibration scores for the scheme with and without bias, in comparison to that of the prior, was due to the too narrow posterior uncertainties for the former and the correlation in the posterior uncertainties for the latter. As the calibration was conducted based only on one FEBA test, this suggested a symptom of overfitting, which was stronger for the former than the latter. Therefore, in the case of limited calibration data, it might be prudent to consider instead the scheme with bias but excluding a strongly correlated parameter, whose calibration scores were consistently high across FEBA tests albeit with relatively lower informativeness compared to the two previous schemes (but, still much higher than that of the prior).

6.2 MAIN ACHIEVEMENTS AND RECOMMENDATIONS FOR FUTURE WORK

The thesis proposed the application of a set of methods adapted from the applied literature with the goal of quantifying the uncertainty of model parameters in a TH system code. The application of each method was illustrated and demonstrated on the basis of a reflood experiment simulation model in the TRACE code. According to Section 1.3.2 the listed objectives of the proposed methods were to:

- analyze and better understand the inputs/outputs relationship in a computer simulation with uncertain input;
- approximate the inputs/outputs relationship of a complex computer simulation for a faster evaluation; and,
- calibrate the physical model parameters against various relevant experimental data.

During the course of this doctoral research, each of these methods was investigated and applied to the running example of the FEBA reflood facility simulation model in the TRACE code. Each of these applications was aimed to illustrate the particularities – and difficulties – of applying the method to the TRACE model as well as to demonstrate the values of the method. Chapters 3–5 provided a detailed account on the methods and their applications, of which the main achievements are highlighted below. Given the limited scope and duration of the project, many difficulties found along the way remained unaddressed, and are the basis for the proposed recommendations.

The thesis project was initiated by the participation of PSI-LRS to the OECD/NEA PREMIUM benchmark. The work related to that participation also constitutes a portion – and achievement – of the thesis project.

Four papers were presented in international conferences [23, 112, 142, 153], a journal article was published [253], and two contributions were submitted [17, 126] to the PREMIUM project and included in the NEA reports [20, 21].

6.2.1 Contributions to OECD/NEA PREMIUM Project

The TRACE model of FEBA was successfully developed within that context and became the basis for several follow-up studies. The model is stable and is relatively quick to run allowing even a relatively brute force sensitivity analysis method to be applied. It is now part of the in-house TRACE code validation database at LRS.

*TRACE model of
FEBA*

The prior uncertainties of the input parameters were quantified under the supervision of thermal-hydraulics experts at LRS [126]. The quantified uncertainties were then propagated both in the TRACE models of FEBA and PERICLES (another reflood facility not presented in this thesis). The results of the propagation submitted to the PREMIUM project were deemed satisfactory as it served the purpose of the prior quantification. That is, the prediction uncertainties of both facilities were wide but covered the experimental data well, confirming that the prior range was not underestimated.

*Contribution to
PREMIUM, prior
uncertainty
quantification*

Still within the context of PREMIUM, a python scripting tool was developed to assist conducting computer experiment on the TRACE model of FEBA. The tool `trace-simexp` has reached a stable version, is well documented, and has been applied in several follow-up studies within and outside the scope of the present doctoral research.

trace-simexp

RECOMMENDATIONS FOR FUTURE WORK

Although stable, the current version of `trace-simexp` has been only tested so far for the TRACE model of FEBA. Extension to other TRACE models are feasible. However, depending on the complexity of those models and the computing infrastructure, further development of the tool might become unrealistic. In the long run, it would be better to opt for the use of an integrated uncertainty framework (e.g., UQLab [254], Dakota [255], OpenTurns [256], Uranie [257]). Typically, such framework supports an application programming interface (API) to make a connection with an external simulation model or to a third-party program. It does require an initial effort of getting acquainted with the terminologies the framework but in the long run for a generic complex model these are the preferred solution.

6.2.2 *Implementation and application of GSA methods (to analyze and better understand the inputs/outputs relationship in a computer simulation with uncertain input)*

Implementation and application of screening methods

The size of the initial selection of input parameters, as exemplified in PREMIUM, can be large. Lacking prior knowledge, the selection should also include all the parameters that are vaguely perceived as important. The implementation and the application of screening methods (Morris screening method and Sobol' total-effect indices), as demonstrated in this thesis for the TRACE model of FEBA, allows for a quick, systematic, quantitative screening of the initial set of input parameters in a global manner (i.e., simultaneous perturbation over the whole range of parameter uncertainties), and with less assumptions regarding the linearity or monotonicity of the model or it being additive. The last point motivates the departure from more conventional GSA methods based on correlation coefficients (such as Person and Spearman's)¹. In the case studied here, more than half of the initial parameters were found to be non-influential to the relevant outputs of the reflood simulation.

Application of GSA coupled with FDA

In accordance with the aim of increasing the understanding of inputs/outputs relationship in a simulation, a novel set of QoIs was derived using FDA techniques to characterize the overall time-dependent output variation. It was able to capture the most essential features of the model behavior through its time-dependent output, thus significantly departing from the more conventional QoIs (e.g., minimum, maximum, or time-average scalar value) that have been used so far in similar SA studies of TH simulation model. The resulting QoIs were then coupled with the GSA methods (Sobol' main- and total-effect indices) to investigate, quantitatively, the effect of the input parameters on the overall time-dependent outputs. When considering FDA-based QoIs, which better represents the whole transient of an output, it was found that the important parameters and the nature of their interactions were changing during the transient. The nature of these interactions, however, remains to be investigated and is outside the scope of this thesis.

gsa-module

Finally, the implementations of the employed GSA methods were developed in-house as a python module to allow full internal control on the implementation. The module `gsa-module` has been documented and tested against a suite of test functions. It was applied to obtain all the results presented in Chapter 3.

¹ Although these methods, unlike Sobol' indices, still allow for dealing with correlated inputs in straightforward manner.

RECOMMENDATIONS FOR FUTURE WORK

The landmark registration procedure to separate the phase and amplitude variations in a functional data set is one of the most straightforward procedure available. However, landmarks might not always be visible and miss-specification might affect the downstream analysis. As seen in this thesis, slight residual variation during quenching persisted after registration which caused an inflated (artifact) sensitivity indices around the vicinity of quenching. Therefore, a more automatic registration technique is worth investigating and applied to different types of functional data of interest in TH simulation.

In this thesis, Monte Carlo (MC) simulation was used to estimate the Sobol' sensitivity indices (main- and total-effect). Though it was considered affordable for the analysis of the TRACE model of FEBA, this will become a bottleneck for an application of the method to wide range of computationally expensive transient simulations. In such cases, an alternative approach to compute the indices is required.

The `gsa`-module was developed during the course of the thesis with the idea of implementing the available methods from the literature in a quick manner; without having to deal with the learning curve of adopting existing framework. Additionally, such an approach allows for a full control on the implementation. The structure of the module makes it easy to be extended for other GSA methods. However, state of the art uncertainty quantification frameworks such as the ones mentioned above are much more powerful and some are actively developed with substantial user base. Thus, for an advanced GSA methods that are already implemented in any such frameworks, it is worthwhile to simply adopt the frameworks in the future.

6.2.3 *Development and validation of a TRACE metamodel (to approximate the inputs/outputs relationship of a complex computer simulation for a faster evaluation)*

Gaussian process (GP) metamodeling has been demonstrated for the TRACE model of FEBA, which has high-dimensional outputs. In this thesis, the high-dimensionality of the outputs was treated by PCA resulting in a GP PC metamodel. The validation and testing steps then showed that the error of the metamodel across the prior range of input parameters were within a reasonable range. In other words, it managed to approximate the important features of the inputs/outputs relationship of the reflood simulation model in TRACE. Using the Gaussian process (GP) principal component (PC) as a surrogate for TRACE, the prediction for arbitrary input parameter values could be made much faster (i.e., < 5 [s] per metamodel evaluation vs. $6 - 15$ [min] per TRACE). The thesis has also demonstrated the applicability of PCA to reduce the high dimension of the output. The tech-

nique performed best for relatively smooth outputs (in this particular application, the pressure drop and liquid carryover transients), while it performed worse for reconstructing an output exhibiting strong discontinuity (e.g., the clad temperature output exhibited a discontinuity around quenching). Finally, though many practical aspects were involved in the construction of the metamodel, the work in the thesis concluded that the size of the training sample (i.e., the actual number of code runs) was the most important factor; if they can be afforded, more runs should be conducted.

RECOMMENDATIONS FOR FUTURE WORK

*Alternative
dimension reduction
technique*

The worse performance of the PCA on reconstructing the clad temperature output was, in turn, due to the use of PCA as the linear dimension reduction. As such, a first step of improvement in this regard can be aimed toward replacing PCA with another more advanced, dimension reduction tool. Simulations with high-dimensional outputs, either in time or space, are typical in TH analysis. It is thus worth investigating the application of different dimension reduction techniques, linear (extension of PCA, e.g., [258]) or nonlinear (e.g., isomap [259], locally linear embedding (LLE) [260], or wavelet [261]). Many of such developments are made in the area of image processing. Indeed as shown in Chapter 4, a 1-dimensional time-dependent TRACE simulation output can be represented as an image.

*Alternative
metamodeling
techniques*

Furthermore, GP metamodel is not the only available metamodeling technique. The response surface method was traditionally employed for TH system analysis but more advanced techniques are currently available such as the ones mentioned in Section 1.4.2. The investigation on their applicability – the predictive performance and the computational cost of construction – for a variety of TH models is of interest in its own right.

*Alternative
workflow*

Finally, the step proposed in this thesis is to conduct sensitivity analysis before constructing the metamodel. In that case, metamodeling error can be excluded from the sensitivity analysis. However, it is also possible to construct the metamodel before moving on to the sensitivity analysis step. Some metamodeling techniques allow the metamodeling and sensitivity analysis to be combined while providing an estimate of the associated error. In particular polynomial chaos expansion (PCE) allows the computation of Sobol' sensitivity indices by post-processing the resulting coefficients of the expansion [71].

6.2.4 *Bayesian calibration of the TRACE reflood model parameters against various relevant experimental data*

Bayesian calibration was successfully applied to quantify the uncertainty of the selected TRACE reflood model parameters on the basis of the

FEBA experiments. Different posterior distributions of the model parameters, corresponding to different calibration assumptions, were formulated and directly sampled from using an MCMC ensemble sampler. The uncertainty propagation of each resulting posterior samples was conducted on all FEBA tests and the results were compared in terms of *informativeness* (the width of the prediction uncertainty band) and *calibration score* (consistency with the experimental data and coverage by the uncertainty band).

The value of incorporating model bias term in the calibration process has also been demonstrated. Without the model bias term, the calibration results exhibited stronger symptom of overfitting, i.e., although the prediction uncertainty band was narrower, more experimental data points fell outside the band. At the same time, the posterior uncertainties from the calibration scheme with model bias term resulted in a particular correlation structure that might be overly specific to the calibration data. Indeed, though better in terms of calibration score with respect to the scheme without model bias term, the posterior with bias term had consistently lower calibration scores across all FEBA tests compared to that of the prior. By removing a strongly correlated parameter from the calibration – and keeping it at its prior uncertainty – the resulting posterior prediction uncertainty was found to have a much improved informativeness with similar level of calibration scores across all FEBA tests compared to that of the prior. Therefore, it can be argued that the calibration resulted in a posterior range and a posterior correlation structure which were, by construction, specific to the calibration data. However, if the calibration data was deemed not sufficiently large or comprehensive enough (here it was based on one FEBA test run) then care should be taken to avoid overfitting. In this particular case, the calibration by excluding a strongly correlated parameter was proved to be a compromise and a pragmatic solution.

Another type of parameter non-identifiability was also encountered during the calibration process. The non-identifiability due to the parameter insensitivity with respect to a type of experimental data was solved by employing a calibration scheme that incorporated multiple types of experimental data simultaneously.

RECOMMENDATIONS FOR FUTURE WORK

It is worth noting that the calibration conducted in the present doctoral research was based only on the data from one FEBA test. The formulation of the model bias term only considered the bias from one experimental boundary conditions. The applicability of the resulting posterior uncertainty is only applicable insofar that the assumed bias from the calibration data is valid for the test data (i.e., all the other FEBA tests). For FEBA, different experimental conditions leads to dif-

ferent bias structure of the model, which was apparent in the case of test Nos. 223 and 218 (i.e., tests with lower system backpressure compared to test for the calibration). Indeed the propagation of the posterior uncertainties performed poorly in such situation. Furthermore, a more consistent uncertainty propagation should incorporate the bias term if the term is used in the calibration.

A more comprehensive calibration procedure should therefore take into account the difference in the bias structure from different experimental conditions. One possibility is to concatenate all the experimental data of FEBA into a single calibration process resulting in a posterior uncertainty of model parameters that takes into account all available experimental boundary conditions of FEBA. However, to understand better the difference between the conditions, an alternative approach is the hierarchical modeling [110], which allows the model parameters to take different posterior uncertainties depending on the experimental conditions, while at the same time allowing sharing information from the data across different experimental conditions. This approach might give a better insight on the model validity and reveal its discrepancy for a particular experimental condition in a more precise manner.

It should be noted that the model parameters are not of primary interest themselves. Their calibration against experimental data are aimed at increasing the confidence in their application for the actual plant analysis (or, to a lesser degree, integral effect test facilities). In its own right, the presence of correlation in the model parameters presents a challenge in the application of the posterior uncertainties. Model parameters uncertainty are typically assumed to be independent a priori. After calibration such assumption might not hold anymore. A consistent propagation of uncertainty should consider the correlation structure that is informed by experimental data. As was observed in this thesis, the correlation structure of the posterior might not be readily represented as a familiar multivariate Gaussian. Once more, in this thesis, the issue was sidestepped by using directly the posterior samples for the uncertainty propagation thus implicitly capturing the correlation structure. On how to summarize this correlation structure for the purpose of uncertainty propagation remains an open question

Regarding computational aspects of the Bayesian calibration, MCMC sampler is the backbone of the method. In this thesis only one kind of sampler was used and no direct comparison on its performance was made against different kind of sampler. For robustness, it is necessary to extend the verification study using different MCMC samplers.

TRACE CODE GOVERNING EQUATIONS

The hydraulic module of TRACE is based on a two-fluid six-equation model, solving the conservation equations of mass, momentum, and energy for the liquid and vapor phases in the coolant [28]. Furthermore, the formulations are given in volumetric term (i.e., per unit volume basis) with a reference to a select control volume (or *node*). A symbol is defined the first time it appears in an equation.

In the subsequent section the angle brackets and the overbar will be dropped from the void fraction notation Eq. (2.2) and any mention of void fraction will refer to the above time- and volume (area)-averaged formulation.

A.1 MASS BALANCE EQUATIONS

The mass balance equations given for liquid and gas phases are,

$$\frac{\partial[(1-\alpha)\rho_l]}{\partial t} + \nabla \cdot [(1-\alpha)\rho_l \mathbf{v}_l] = -\Gamma \quad (\text{A.1})$$

$$\frac{\partial[\alpha\rho_g]}{\partial t} + \nabla \cdot [\alpha\rho_g \mathbf{v}_g] = \Gamma \quad (\text{A.2})$$

where the subscripts indicate the phase, l for the liquid phase and g for the gas phase (vapor); α is the void fraction; ρ_l (ρ_g) is the mass density of the liquid (gas) phase; and \mathbf{v}_l (\mathbf{v}_g) is the flow velocity of the liquid (gas) phase. The terms in either sides of the two mass balance equations are explained in Table A.1.

Table A.1: The terms in TRACE two-fluid model mass balance equations (all are given in volumetric term)

TERMS	LIQUID PHASE	GAS PHASE
mass rate of change	$\frac{\partial[(1-\alpha)\rho_l]}{\partial t}$	$\frac{\partial[\alpha\rho_g]}{\partial t}$
mass convection rate	$\nabla \cdot [(1-\alpha)\rho_l \mathbf{v}_l]$	$\nabla \cdot [\alpha\rho_g \mathbf{v}_g]$
interfacial mass-transfer rate	$-\Gamma$	Γ

Note that the term Γ , the volumetric interfacial mass-transfer rate, is given with a convention that it is positive for the transfer from liquid phase to gas phase. This term is defined in Eq. (A.15) below.

A.2 MOMENTUM BALANCE EQUATIONS

The momentum balance equations are given for liquid and gas phases as follows,

$$\begin{aligned} \frac{\partial[(1-\alpha)\rho_l\mathbf{v}_l]}{\partial t} + \nabla \cdot [(1-\alpha)\rho_l\mathbf{v}_l \otimes \mathbf{v}_l] + (1-\alpha)\nabla p \\ = \mathbf{f}_i + \mathbf{f}_{wl} + (1-\alpha)\rho_l\mathbf{g} - \Gamma\mathbf{v}_i \end{aligned} \quad (\text{A.3})$$

$$\begin{aligned} \frac{\partial[\alpha\rho_g\mathbf{v}_g]}{\partial t} + \nabla \cdot [\alpha\rho_g\mathbf{v}_g \otimes \mathbf{v}_g] + \alpha\nabla p \\ = -\mathbf{f}_i + \mathbf{f}_{wg} + \alpha\rho_g\mathbf{g} + \Gamma\mathbf{v}_i \end{aligned} \quad (\text{A.4})$$

where ∇p is the pressure gradient; \mathbf{f}_i is the volumetric force due to shear at the phase interface; \mathbf{f}_{wl} is the volumetric force acting on the liquid phase due to shear at the wall (i.e., fluid-structure contact); \mathbf{f}_{wg} is the volumetric force acting on the gas phase due to shear at the wall; \mathbf{g} is the gravitational acceleration; and \mathbf{v}_i is the flow velocity at the phase interface. Table A.2 lists the terms in either sides of the two momentum balance equations.

Table A.2: The terms in TRACE two-fluid model momentum balance equations (all are given in volumetric term)

TERMS	LIQUID PHASE	GAS PHASE
momentum rate of change	$\frac{\partial[(1-\alpha)\rho_l\mathbf{v}_l]}{\partial t}$	$\frac{\partial[\alpha\rho_g\mathbf{v}_g]}{\partial t}$
momentum convection rate	$\nabla \cdot [(1-\alpha)\rho_l\mathbf{v}_l \otimes \mathbf{v}_l]$	$\nabla \cdot [\alpha\rho_g\mathbf{v}_g \otimes \mathbf{v}_g]$
pressure gradient	$(1-\alpha)\nabla p$	$\alpha\nabla p$
momentum change due to:		
interfacial friction	\mathbf{f}_i	$-\mathbf{f}_i$
wall friction	\mathbf{f}_{wl}	\mathbf{f}_{wg}
body force	$(1-\alpha)\rho_l\mathbf{g}$	$\alpha\rho_g\mathbf{g}$
interfacial mass-transfer	$-\Gamma\mathbf{v}_i$	$\Gamma\mathbf{v}_i$

Note that the formulation in TRACE uses the simplifying assumption of $p_i = p_g = p_l$. That is, the pressure in a given control volume is the same in either phases as well as at the interface [28].

For the friction (shear) terms in right hand side, TRACE uses the following formulations,

$$\mathbf{f}_i = C_i(\mathbf{v}_g - \mathbf{v}_l)|\mathbf{v}_g - \mathbf{v}_l| \quad (\text{A.5})$$

$$\mathbf{f}_{wl} = -C_{wl}\mathbf{v}_l|\mathbf{v}_l| \quad (\text{A.6})$$

$$\mathbf{f}_{wg} = -C_{wg}\mathbf{v}_g|\mathbf{v}_g| \quad (\text{A.7})$$

where the friction coefficients C_i , C_{wl} , C_{wg} for interfacial shear, wall-liquid shear, and wall-gas shear, respectively are obtained from flow regime-dependent empirical correlations.

A.3 ENERGY BALANCE EQUATIONS

The energy balance equations are defined for liquid and gas phases as,

$$\begin{aligned} & \frac{\partial[(1-\alpha)\rho_l(e_l + |\mathbf{v}_l|^2/2)]}{\partial t} + \nabla \cdot \left[(1-\alpha)\rho_l \left(e_l + \frac{P}{\rho_l} + \frac{|\mathbf{v}_l|^2}{2} \right) \mathbf{v}_l \right] \\ & = q_{il} + q_{wl} + q_{wsat} + q_{dl} + (1-\alpha)\rho_l \mathbf{g} \cdot \mathbf{v}_l \\ & \quad - \Gamma h'_l + (\mathbf{f}_i + \mathbf{f}_{wl}) \cdot \mathbf{v}_l \end{aligned} \quad (\text{A.8})$$

$$\begin{aligned} & \frac{\partial[\alpha\rho_g(e_g + |\mathbf{v}_g|^2/2)]}{\partial t} + \nabla \cdot \left[\alpha\rho_g \left(e_g + \frac{P}{\rho_g} + \frac{|\mathbf{v}_g|^2}{2} \right) \mathbf{v}_g \right] \\ & = q_{ig} + q_{wg} + q_{dg} + \alpha\rho_g \mathbf{g} \cdot \mathbf{v}_g + \Gamma h'_g + (-\mathbf{f}_i + \mathbf{f}_{wg}) \cdot \mathbf{v}_g \end{aligned} \quad (\text{A.9})$$

where e_l (e_g) is the liquid (gas) phase internal energy; q_{il} (q_{ig}) is the volumetric interfacial heat transfer on the liquid (gas) phase; q_{wl} (q_{wg}) is the volumetric wall (sensible) heat transfer on the liquid (gas) phase; q_{wsat} is the volumetric wall (latent) heat transfer on the liquid phase; q_{dl} (q_{dg}) is the volumetric direct power deposition on the liquid (gas) phase; h'_l is the bulk liquid enthalpy; and h'_g is the gas phase saturation enthalpy. Table A.3 lists all of the terms in either sides of the two energy balance equations.

The heat transfer terms between the wall and the phases follow Newton's law of cooling,

$$q_{wl} = h_{wl} a_{wl} (T_w - T_l) \quad (\text{A.10})$$

$$q_{wg} = h_{wg} a_{wg} (T_w - T_g) \quad (\text{A.11})$$

$$q_{wsat} = h_{wsat} a_{wl} (T_w - T_{sat}) \quad (\text{A.12})$$

where T_w , T_l , T_g , T_{sat} are the wall, liquid phase, liquid phase, and liquid saturation temperatures, respectively; a_{wl} (a_{wg}) is the volumetric surface contact area between the wall and liquid (gas) phase (or *the interfacial area concentration*); and h_{wl} , h_{wg} , and h_{wsat} are the HTC's

Table A.3: The terms in TRACE two-fluid model momentum energy equations (all are given in volumetric term)

TERMS	LIQUID PHASE	GAS PHASE
energy rate of change	$\frac{\partial[(1-\alpha)\rho_l(e_l+ \mathbf{v}_l ^2/2)]}{\partial t}$	$\frac{\partial[\alpha\rho_g(e_g+ \mathbf{v}_g ^2/2)]}{\partial t}$
energy convection rate	$\nabla \cdot \left[(1-\alpha)\rho_l \left(e_l + \frac{p}{\rho_l} + \frac{ \mathbf{v}_l ^2}{2} \right) \mathbf{v}_l \right]$	$\nabla \cdot \left[\alpha\rho_g \left(e_g + \frac{p}{\rho_g} + \frac{ \mathbf{v}_g ^2}{2} \right) \mathbf{v}_g \right]$
(sensible) interfacial heat transfer	q_{il}	$-\mathbf{f}_i$
(sensible) wall heat transfer	q_{wl}	\mathbf{f}_{wg}
(latent) wall heat transfer	q_{wsat}	$\alpha\rho_g\mathbf{g}$
direct heat deposition	q_{dl}	$\Gamma\mathbf{v}_i$
energy loss (gain) due to:		
gravity	$(1-\alpha)\rho_l\mathbf{g} \cdot \mathbf{v}_l$	$\alpha\rho_l\mathbf{g} \cdot \mathbf{v}_l$
phase change	$-\Gamma h'_l$	$\Gamma h'_g$
wall and interfacial friction	$(\mathbf{f}_i + \mathbf{f}_{wl}) \cdot \mathbf{v}_l$	$(-\mathbf{f}_i + \mathbf{f}_{wg}) \cdot \mathbf{v}_g$

between wall and liquid, wall and gas, and wall-saturated liquid, respectively. The volumetric surface contact area as well as the heat transfer coefficients are obtained from a set of flow regime-dependent empirical correlations.

Additionally, the heat transfer terms at the interface between the two phases are also modeled using the same law,

$$q_{il} = h_{il} a_i (T_{sg} - T_l) \quad (\text{A.13})$$

$$q_{ig} = \frac{p_g}{p} h_{ig} a_i (T_{sg} - T_g) \quad (\text{A.14})$$

where h_{il} (h_{ig}) is the HTC for liquid (gas) phase at the interface; a_i is the volumetric interfacial surface area; p_g is the partial pressure of the gas phase; and T_{sg} is the saturation temperature corresponding to partial pressure of the gas phase.

Finally, the mass-transfer rate at the interface is defined using a thermal-energy jump condition that results in

$$\Gamma = \frac{-(q_{ig} + q_{il}) + q_{wsat}}{(h'_g - h'_l)} \quad (\text{A.15})$$

In other words, the net heat transfer rate given to the saturated liquid phase, is used entirely for phase change.

A.4 HEAT CONDUCTION EQUATIONS

Besides the set balance equations that govern the two-phase fluid flow, TRACE also includes a heat conduction module (known as *heat structure* component) to model correctly the heat transfer process in solid structures (e.g., active fuel, internal passive structures, etc.) and between the surface of such structures and the contacting fluid.

HEAT CONDUCTION EQUATION, SOLID STRUCTURES:

$$\rho_s C_{ps} \frac{\partial T}{\partial t} - \nabla \cdot (k_s \nabla T) = q_s \quad (\text{A.16})$$

where ρ_s is the solid structure mass density; C_{ps} is the solid structure thermal capacity; k_s is the solid structure thermal conductivity; and q_s is the volumetric heat source term in the solid.

At the contact between fluid and solid material, the total heat flux is given as,

$$q'' = h_{wl} (T_w - T_l) + h_{wsat} (T_w - T_{sat}) + h_{wg} (T_w - T_g) \quad (\text{A.17})$$

where the heat flux at the surface of the structure, q'' is partitioned to different phases of the fluid, either as sensible or latent heat. As can be seen, Eq. (A.17) couples the heat conduction equation with energy balance equations of the fluid through the terms defined in Eqs. (A.10), (A.11), and (A.12).

A.5 CLOSURE AND FLOW REGIMES

In each of the balance equations given above, the right hand side represents the source and sink terms mainly due to fluid interaction with solid structure (*wall*) and the interaction between the phases, among others. The set of balance equations characterizes exactly the two-phase flow inside a control volume in a time- and volume-averaged manner provided that the terms in the right hand side of the equation (such as, Eqs. (A.5)-(A.7) and Eqs. (A.10)-(A.14)) are correct.

ADDITIONAL RESULTS

B.1 PRIOR UNCERTAINTY PROPAGATION OF THE FEBA TESTS

B.1.1 Clad Temperature Output (TC)

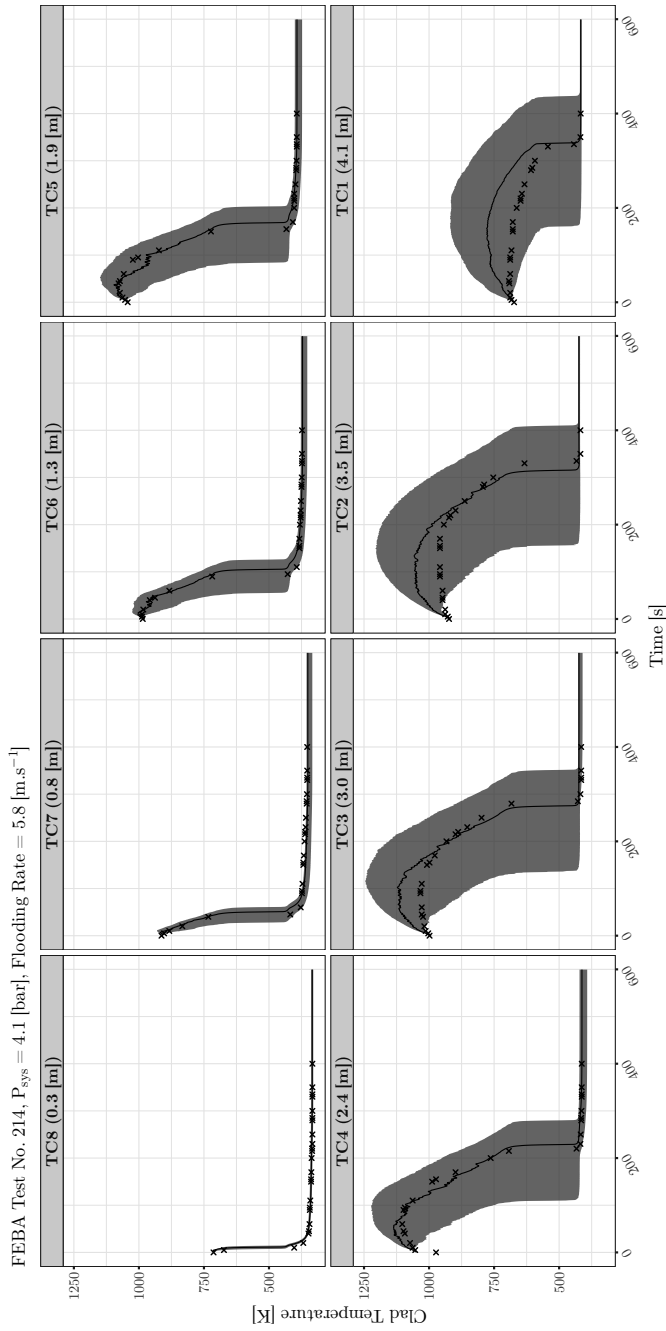


Figure B.1: Propagation of the 27 input parameters prior uncertainties on FEBA test no. 214 for the clad temperature output (TC) at different axial locations using TRACE. The uncertainty bounds correspond to the symmetric (95%) probability; solid lines and crosses indicate the simulation with the nominal parameters values and the experimental data, respectively.

FEBA Test No. 216, $P_{\text{sys}} = 4.1 \text{ [bar]}$, Flooding Rate = $3.8 \text{ [m.s}^{-1}\text{]}$

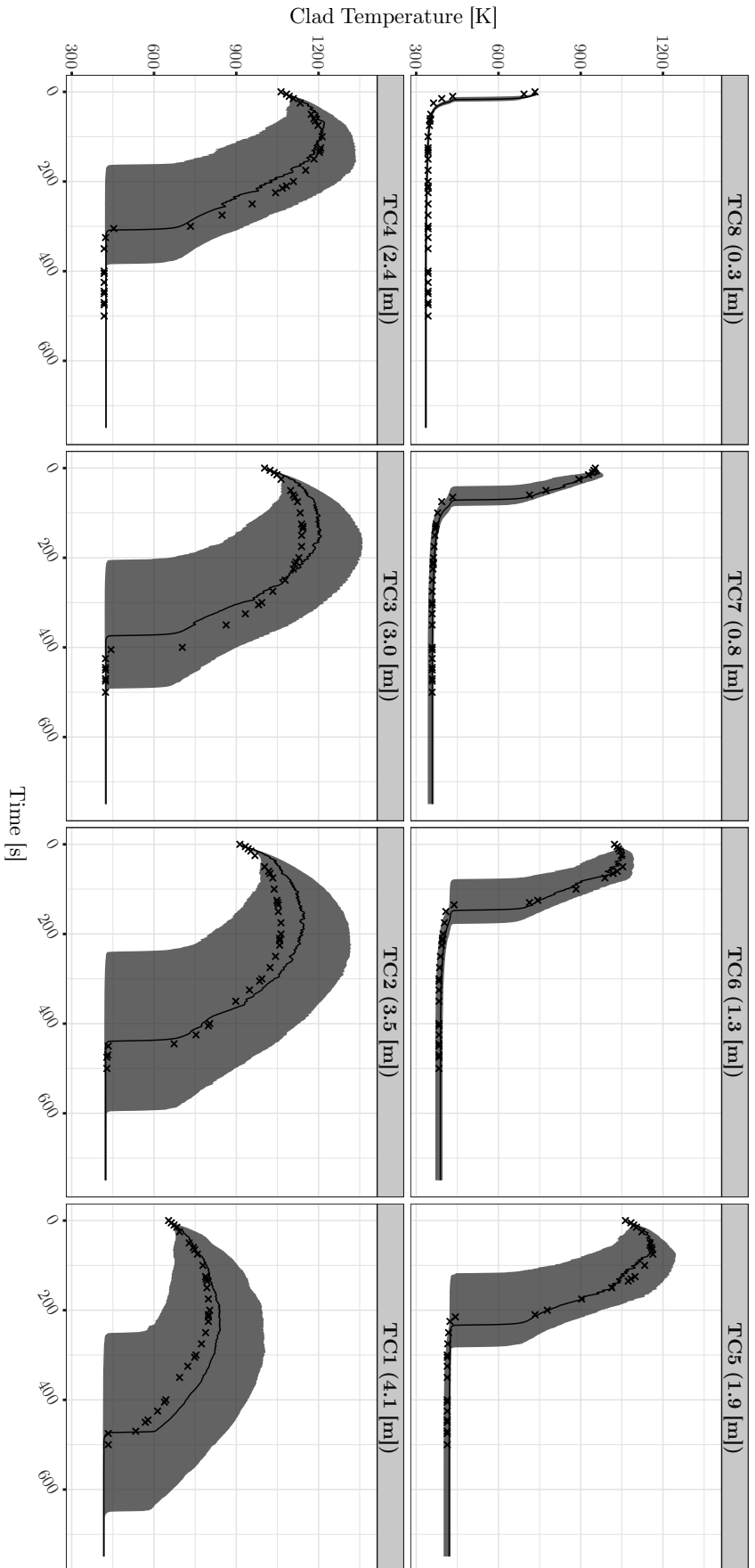


Figure B.2: Propagation of the 27 input parameters prior uncertainties on FEBA test no. 216 for the clad temperature output (TC) at different axial locations using TRACE. The uncertainty bounds correspond to the symmetric (95%) probability; solid lines and crosses indicate the simulation with the nominal parameters values and the experimental data, respectively.

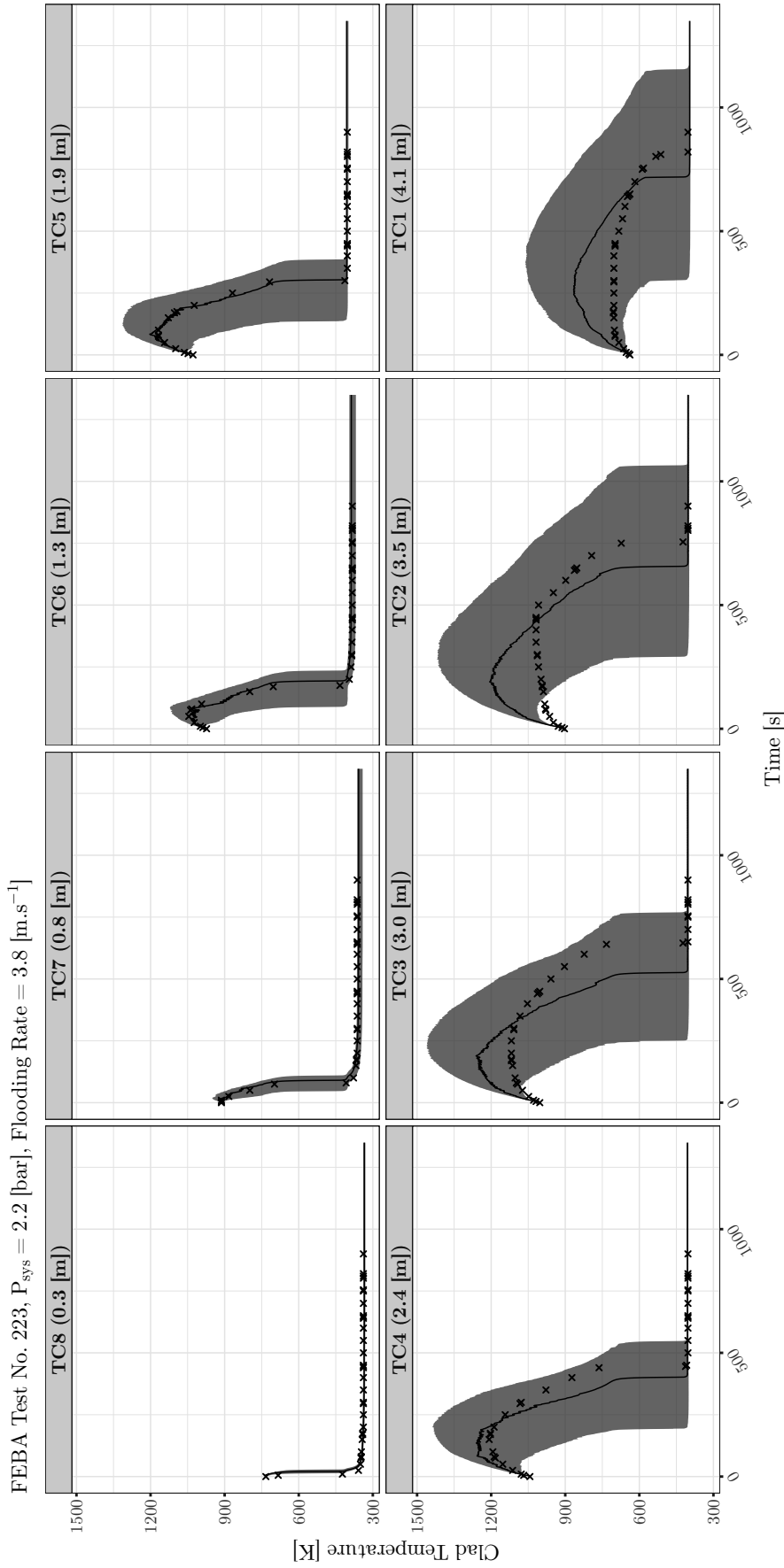


Figure B.3: Propagation of the 27 input parameters prior uncertainties on FEBA test no. 223 for the clad temperature output (TC) at different axial locations using TRACE. The uncertainty bounds correspond to the symmetric (95%) probability; solid lines and crosses indicate the simulation with the nominal parameters values and the experimental data, respectively.

FEBA Test No. 218, $P_{\text{sys}} = 2.1$ [bar], Flooding Rate = 5.8 [m.s⁻¹]

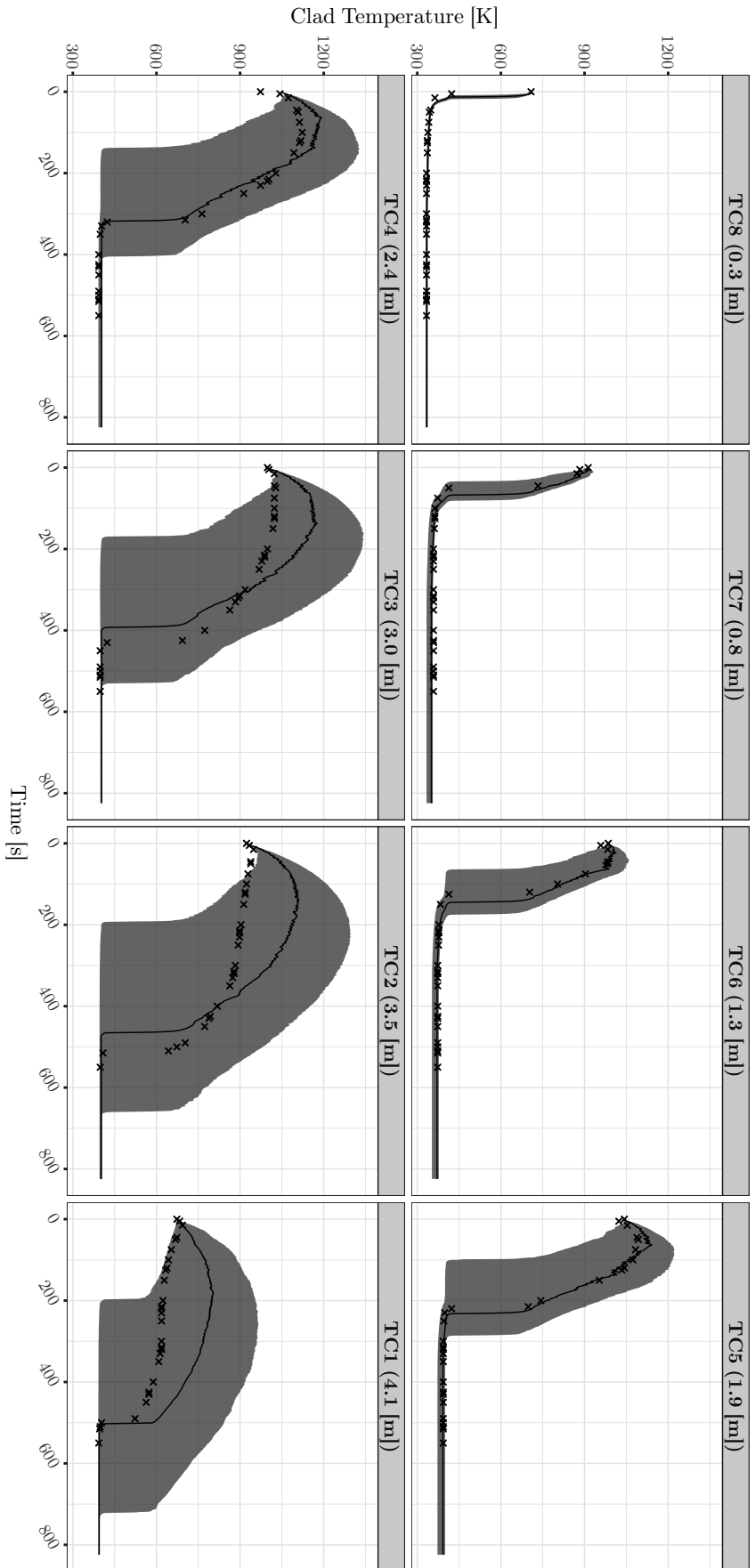


Figure B.4: Propagation of the 27 input parameters prior uncertainties on FEBA test no. 218 for the clad temperature output (TC) at different axial locations using TRACE. The uncertainty bounds correspond to the symmetric (95%) probability; solid lines and crosses indicate the simulation with the nominal parameters values and the experimental data, respectively.

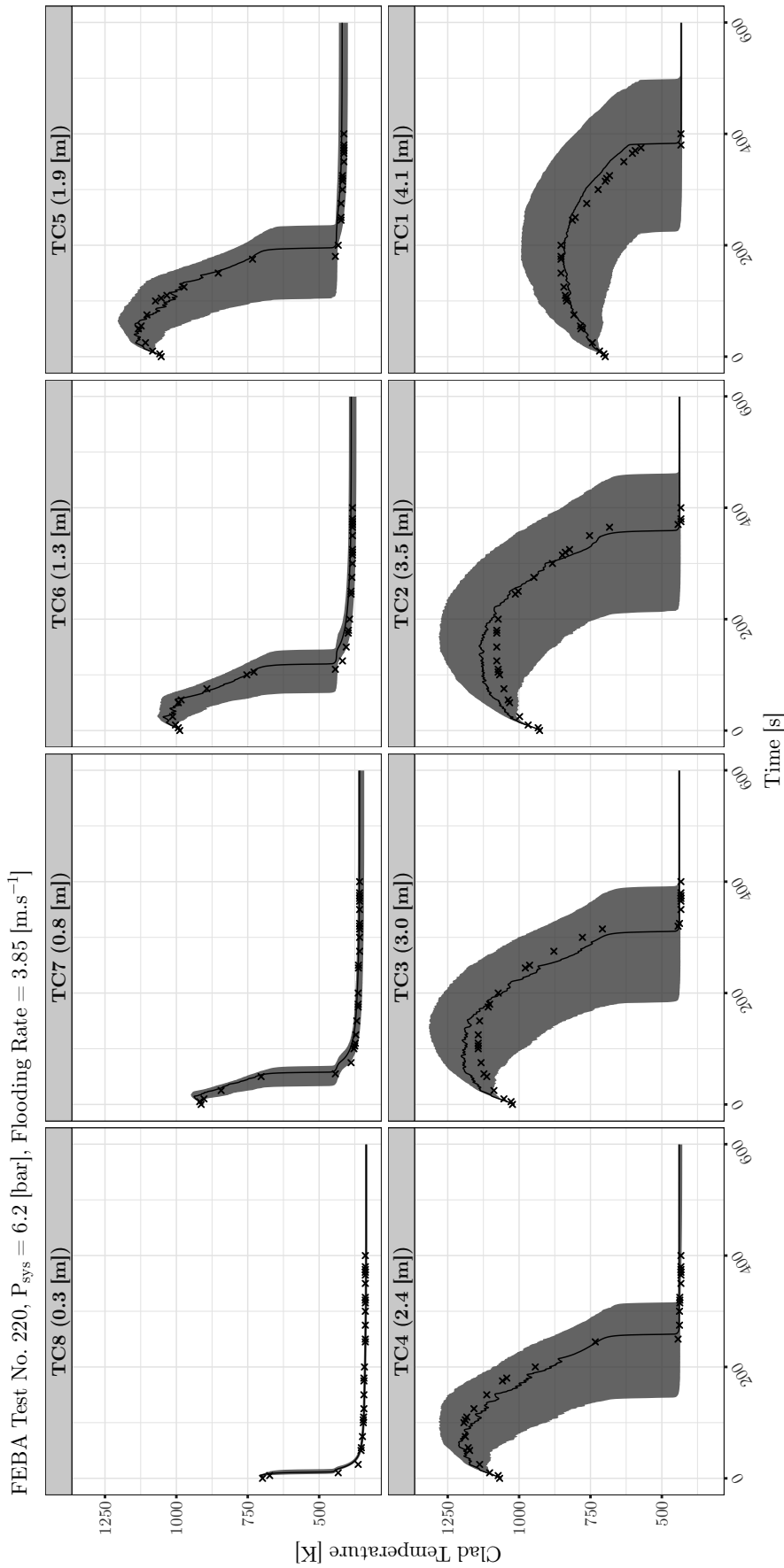


Figure B.5: Propagation of the 27 input parameters prior uncertainties on FEBA test no. 220 for the clad temperature output (TC) at different axial locations using TRACE. The uncertainty bounds correspond to the symmetric (95%) probability; solid lines and crosses indicate the simulation with the nominal parameters values and the experimental data, respectively.

FEBA Test No. 222, $P_{\text{sys}} = 6.2$ [bar], Flooding Rate = 5.8 [m.s⁻¹]

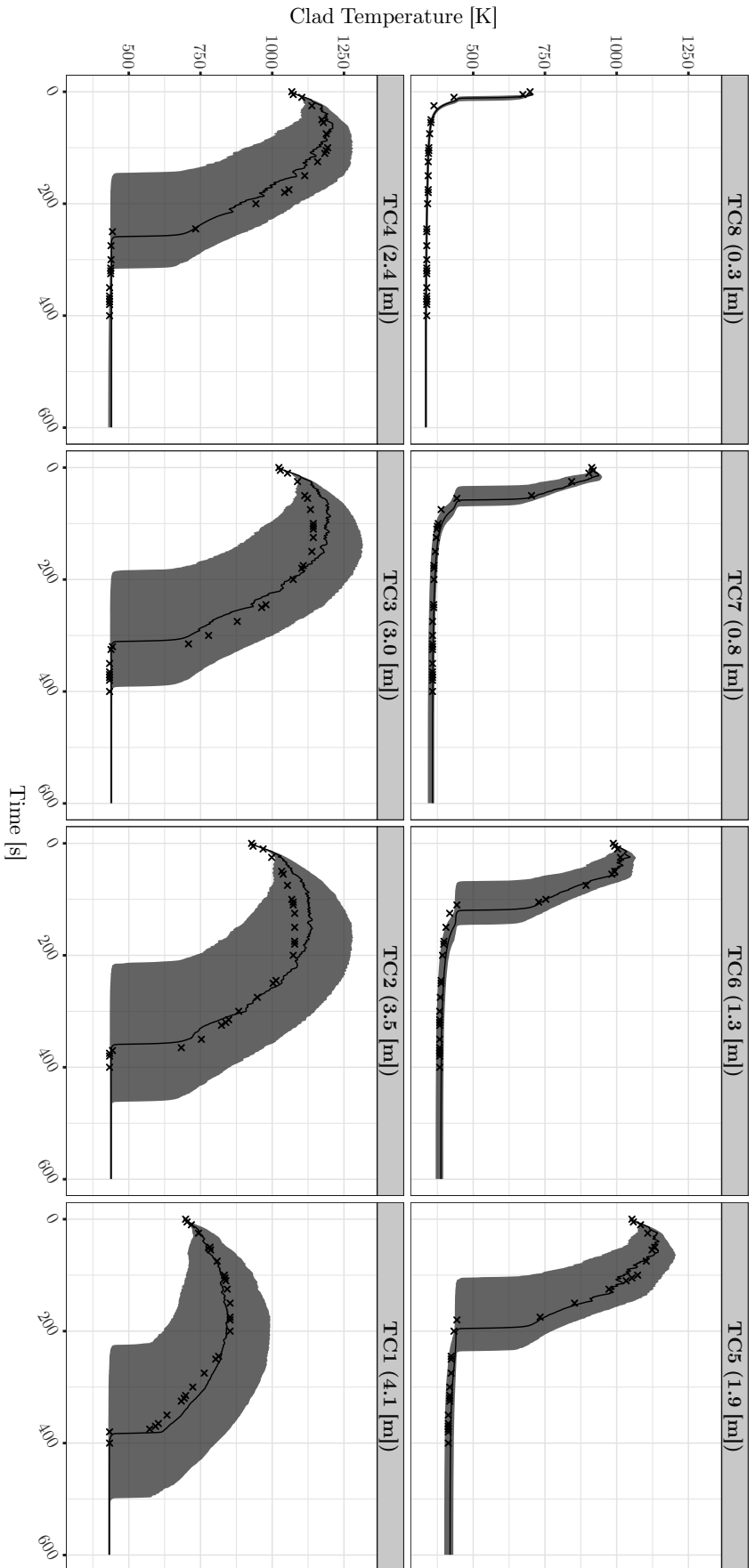


Figure B6: Propagation of the 27 input parameters prior uncertainties on FEBA test no. 222 for the clad temperature output (TC) at different axial locations using TRACE. The uncertainty bounds correspond to the symmetric (95%) probability; solid lines and crosses indicate the simulation with the nominal parameters values and the experimental data, respectively.

B.1.2 Pressure Drop Output (DP)

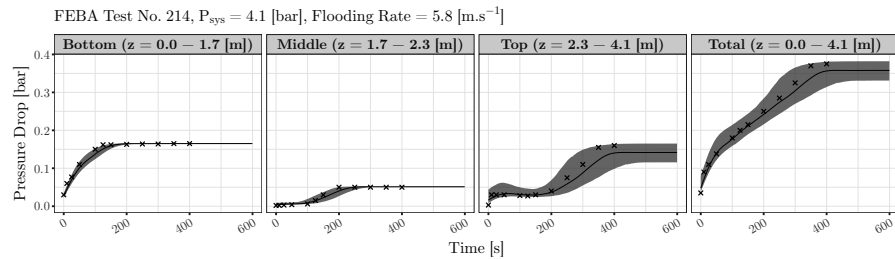


Figure B.7: Propagation of the 27 input parameters prior uncertainties on FEBA test no. 214 for the pressure drop output (DP). The uncertainty bounds correspond to the symmetric (95%) probability; solid lines and crosses indicate the simulation with the nominal parameters values and the experimental data, respectively.

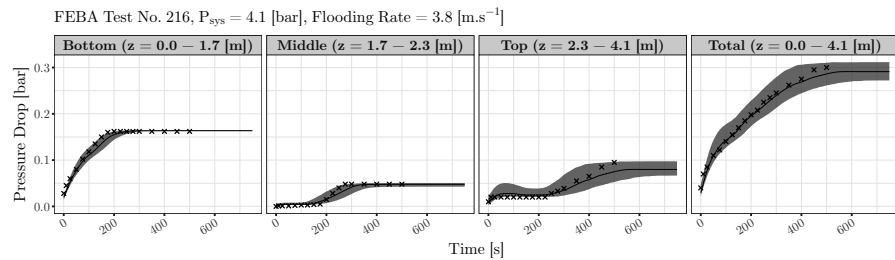


Figure B.8: Propagation of the 27 input parameters prior uncertainties on FEBA test no. 216 for the pressure drop output (DP). The uncertainty bounds correspond to the symmetric (95%) probability; solid lines and crosses indicate the simulation with the nominal parameters values and the experimental data, respectively.

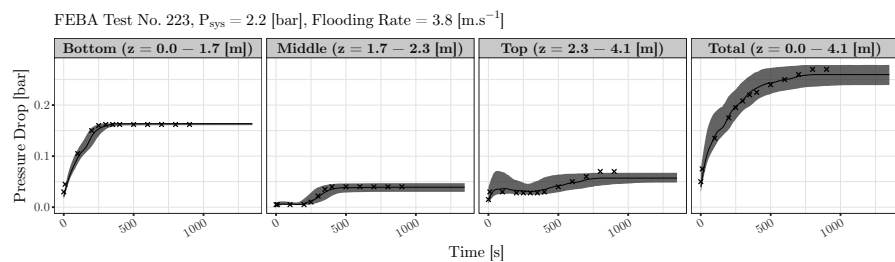


Figure B.9: Propagation of the 27 input parameters prior uncertainties on FEBA test no. 223 for the pressure drop output (DP). The uncertainty bounds correspond to the symmetric (95%) probability; solid lines and crosses indicate the simulation with the nominal parameters values and the experimental data, respectively.

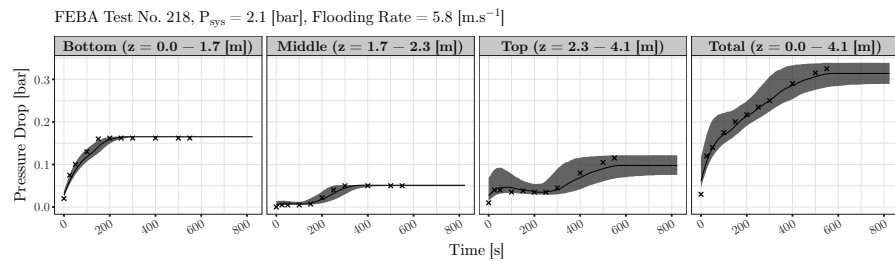


Figure B.10: Propagation of the 27 input parameters prior uncertainties on FEBA test no. 218 for the pressure drop output (DP). The uncertainty bounds correspond to the symmetric (95%) probability; solid lines and crosses indicate the simulation with the nominal parameters values and the experimental data, respectively.

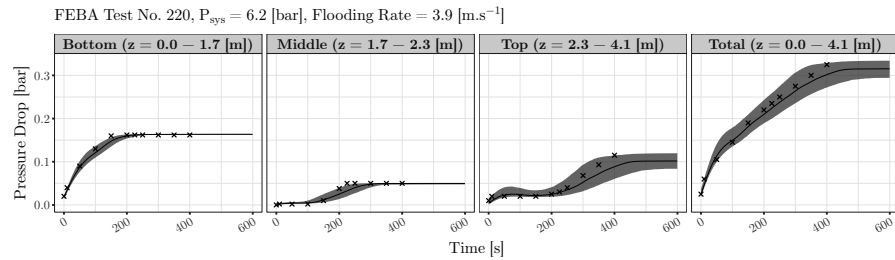


Figure B.11: Propagation of the 27 input parameters prior uncertainties on FEBA test no. 220 for the pressure drop output (DP). The uncertainty bounds correspond to the symmetric (95%) probability; solid lines and crosses indicate the simulation with the nominal parameters values and the experimental data, respectively.

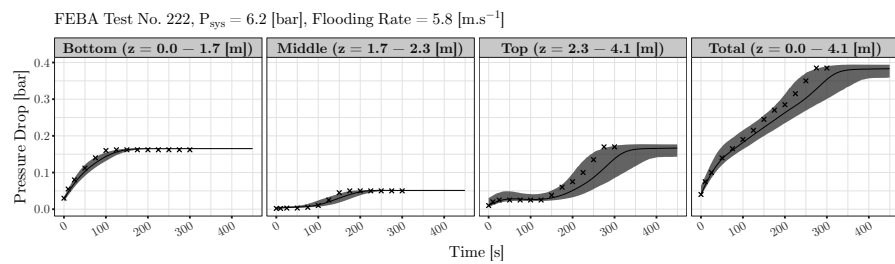
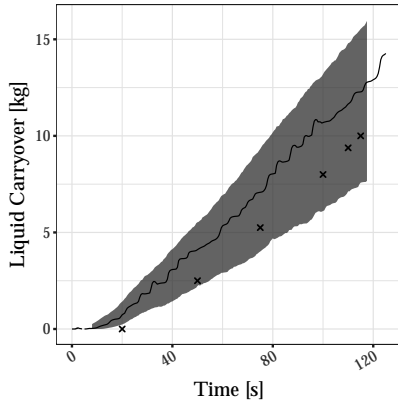
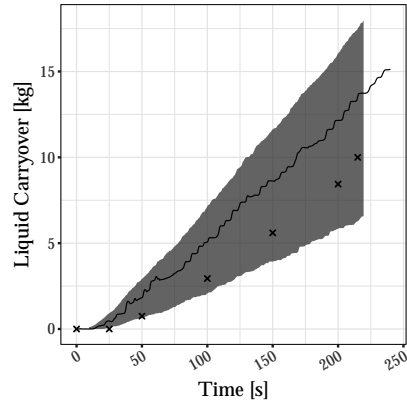


Figure B.12: Propagation of the 27 input parameters prior uncertainties on FEBA test no. 222 for the pressure drop output (DP). The uncertainty bounds correspond to the symmetric (95%) probability; solid lines and crosses indicate the simulation with the nominal parameters values and the experimental data, respectively.

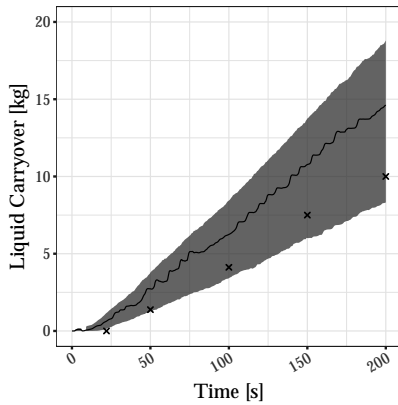
B.1.3 *Liquid carryover Output (CO)*

(a) FEBA test no. 214

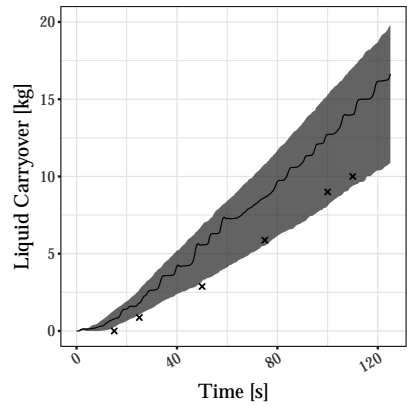


(b) FEBA test no. 216

Figure B.13: Propagation of the 27 input parameters prior uncertainties on FEBA test nos. 214 & 218 for the liquid carryover output (CO). The uncertainty bounds correspond to the symmetric (95%) probability; solid lines and crosses indicate the simulation with the nominal parameters values and the experimental data, respectively.



(a) FEBA test no. 223



(b) FEBA test no. 218

Figure B.14: Propagation of the 27 input parameters prior uncertainties on FEBA test nos. 223 & 218 for the liquid carryover output (CO). The uncertainty bounds correspond to the symmetric (95%) probability; solid lines and crosses indicate the simulation with the nominal parameters values and the experimental data, respectively.

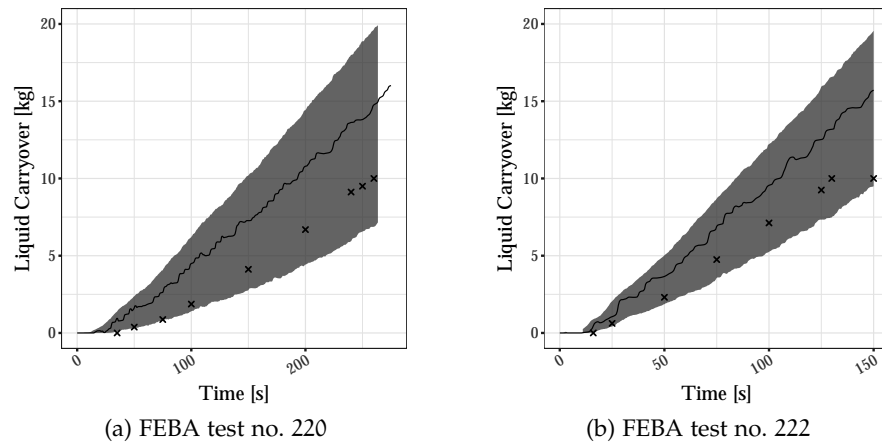


Figure B.15: Propagation of the 27 input parameters prior uncertainties on FEBA test nos. 220 & 222 for the liquid carryover output (CO). The uncertainty bounds correspond to the symmetric (95%) probability; solid lines and crosses indicate the simulation with the nominal parameters values and the experimental data, respectively.

B.2 SCREENING ANALYSIS (27-PARAMETER MODEL)

Table B.1: Parameters importance ranking with respect to average clad temperature output at $z \approx 4.1$ [m] (TC1)

No.	Parameter	Morris Radial			Morris Trajectory			Sobol'-Saltelli		
		Rank	μ_d^*	σ_d	Rank	μ_d^*	σ_d	Rank	$\hat{S}T_d$	95%CI _{pct}
1	breakP	8	0.080	0.058	8	0.077	0.046	8	0.008	(0.007;0.009)
2	fillT	14	0.029	0.045	13	0.021	0.012	13	0.001	(0.001;0.001)
3	fillV	4	0.170	0.056	4	0.167	0.051	4	0.032	(0.029;0.036)
4	pwr	7	0.099	0.064	6	0.098	0.038	6	0.010	(0.009;0.012)
5	nicK	25	0.011	0.037	26	0.002	0.003	25	0.000	(0.000;0.000)
6	nicCP	17	0.020	0.037	14	0.016	0.013	15	0.000	(0.000;0.000)
7	nicEM	27	0.005	0.009	24	0.003	0.005	23	0.000	(0.000;0.000)
8	mgoK	18	0.019	0.096	21	0.005	0.005	21	0.000	(0.000;0.000)
9	mgoCP	10	0.068	0.078	9	0.060	0.039	9	0.004	(0.004;0.005)
10	vesEps	26	0.009	0.026	25	0.002	0.004	27	0.000	(0.000;0.000)
11	ssK	23	0.013	0.038	22	0.004	0.006	22	0.000	(0.000;0.000)
12	ssCp	13	0.031	0.107	16	0.014	0.013	16	0.000	(0.000;0.000)
13	ssEm	24	0.011	0.024	20	0.005	0.007	19	0.000	(0.000;0.000)
14	gridK	11	0.061	0.080	11	0.055	0.021	11	0.003	(0.003;0.004)
15	gridHT	3	0.206	0.210	3	0.220	0.207	3	0.079	(0.069;0.091)
16	iafbWHT	12	0.040	0.063	12	0.028	0.033	12	0.001	(0.001;0.001)
17	dffbWHT	5	0.131	0.146	5	0.124	0.127	5	0.020	(0.017;0.024)
18	iafbVIHT	22	0.014	0.035	19	0.006	0.009	20	0.000	(0.000;0.000)
19	iafbLIHT	15	0.028	0.157	18	0.006	0.012	18	0.000	(0.000;0.000)
20	dffbVIHT	1	0.987	0.489	1	0.939	0.368	1	0.605	(0.547;0.667)
21	dffbLIHT	21	0.015	0.047	23	0.004	0.006	24	0.000	(0.000;0.000)
22	iafbIntDr	6	0.112	0.383	10	0.060	0.091	10	0.003	(0.003;0.004)
23	dffbIntDr	2	0.772	0.541	2	0.765	0.411	2	0.315	(0.284;0.350)
24	iafbWDr	19	0.018	0.069	27	0.002	0.004	26	0.000	(0.000;0.000)
25	dffbWDr	20	0.016	0.025	17	0.010	0.009	17	0.000	(0.000;0.000)
26	transWHT	16	0.022	0.039	15	0.015	0.019	14	0.000	(0.000;0.001)
27	tQuench	9	0.080	0.079	7	0.078	0.062	7	0.008	(0.007;0.009)

Table B.2: Parameters importance ranking with respect to the average clad temperature output at $z \approx 3.5$ [m] (TC2)

No.	Parameter	Morris Radial			Morris Trajectory			Sobol'-Saltelli		
		Rank	μ_d^*	σ_d	Rank	μ_d^*	σ_d	Rank	$\hat{S}T_d$	95%CI _{pct}
1	breakP	9	0.111	0.070	8	0.103	0.051	8	0.013	(0.012;0.015)
2	fillT	16	0.030	0.051	13	0.020	0.016	13	0.001	(0.001;0.001)
3	fillV	5	0.193	0.061	5	0.189	0.051	5	0.040	(0.036;0.045)
4	pwr	8	0.116	0.079	6	0.113	0.045	6	0.014	(0.013;0.016)
5	nicK	25	0.016	0.069	27	0.003	0.005	25	0.000	(0.000;0.000)
6	nicCP	15	0.031	0.080	14	0.019	0.014	15	0.001	(0.000;0.001)
7	nicEM	27	0.008	0.016	23	0.005	0.007	23	0.000	(0.000;0.000)
8	mgoK	20	0.022	0.112	21	0.006	0.007	22	0.000	(0.000;0.000)
9	mgoCP	10	0.082	0.109	10	0.070	0.032	10	0.006	(0.005;0.006)
10	vesEps	26	0.011	0.028	25	0.003	0.006	26	0.000	(0.000;0.000)
11	ssK	18	0.026	0.124	22	0.006	0.009	21	0.000	(0.000;0.000)
12	ssCp	14	0.041	0.168	15	0.017	0.016	14	0.001	(0.000;0.001)
13	ssEm	24	0.016	0.032	20	0.007	0.010	18	0.000	(0.000;0.000)
14	gridK	12	0.045	0.058	12	0.039	0.013	12	0.002	(0.002;0.002)
15	gridHT	4	0.425	0.315	3	0.489	0.211	1	0.304	(0.269;0.342)
16	iafbWHT	11	0.079	0.075	11	0.069	0.045	11	0.005	(0.005;0.006)
17	dffbWHT	3	0.430	0.265	4	0.409	0.218	4	0.162	(0.144;0.183)
18	iafbVIHT	17	0.027	0.113	18	0.009	0.013	19	0.000	(0.000;0.000)
19	iafbLIHT	13	0.045	0.234	19	0.008	0.014	20	0.000	(0.000;0.000)
20	dffbVIHT	2	0.573	0.275	2	0.551	0.235	3	0.211	(0.189;0.236)
21	dffbLIHT	21	0.022	0.090	24	0.004	0.007	24	0.000	(0.000;0.000)
22	iafbIntDr	6	0.153	0.404	9	0.102	0.146	9	0.008	(0.007;0.010)
23	dffbIntDr	1	0.759	0.529	1	0.747	0.401	2	0.300	(0.269;0.334)
24	iafbWDr	23	0.021	0.072	26	0.003	0.006	27	0.000	(0.000;0.000)
25	dffbWDr	22	0.022	0.059	17	0.010	0.011	17	0.000	(0.000;0.000)
26	transWHT	19	0.024	0.049	16	0.014	0.020	16	0.000	(0.000;0.000)
27	tQuench	7	0.116	0.065	7	0.108	0.050	7	0.014	(0.012;0.016)

Table B.3: Parameters importance ranking with respect to average clad temperature output at $z \approx 3.0$ [m] (TC3)

No.	Parameter	Morris Radial			Morris Trajectory			Sobol'-Saltelli		
		Rank	μ_d^*	σ_d	Rank	μ_d^*	σ_d	Rank	$\hat{S}T_d$	95%CI _{pct}
1	breakP	10	0.115	0.077	10	0.104	0.059	9	0.014	(0.013;0.016)
2	fillT	16	0.033	0.061	14	0.021	0.019	15	0.001	(0.001;0.001)
3	fillV	6	0.192	0.071	5	0.188	0.056	5	0.040	(0.036;0.045)
4	pwr	9	0.115	0.078	8	0.112	0.055	8	0.015	(0.013;0.017)
5	nicK	24	0.021	0.112	27	0.004	0.006	25	0.000	(0.000;0.000)
6	nicCP	17	0.032	0.050	13	0.023	0.015	13	0.001	(0.001;0.001)
7	nicEM	27	0.010	0.021	24	0.006	0.008	23	0.000	(0.000;0.000)
8	mgoK	19	0.027	0.129	21	0.008	0.009	19	0.000	(0.000;0.000)
9	mgoCP	11	0.093	0.114	11	0.079	0.029	11	0.007	(0.007;0.008)
10	vesEps	26	0.012	0.045	26	0.004	0.007	26	0.000	(0.000;0.000)
11	ssK	21	0.026	0.104	22	0.007	0.009	22	0.000	(0.000;0.000)
12	ssCp	12	0.054	0.247	15	0.019	0.020	14	0.001	(0.001;0.001)
13	ssEm	25	0.019	0.035	20	0.009	0.012	18	0.000	(0.000;0.000)
14	gridK	14	0.036	0.037	12	0.030	0.011	12	0.001	(0.001;0.001)
15	gridHT	2	0.484	0.364	2	0.558	0.235	1	0.384	(0.340;0.433)
16	iafbWHT	8	0.124	0.122	9	0.108	0.066	10	0.014	(0.012;0.016)
17	dffbWHT	3	0.479	0.295	3	0.455	0.243	3	0.207	(0.184;0.234)
18	iafbVIHT	15	0.034	0.120	17	0.012	0.016	17	0.000	(0.000;0.000)
19	iafbLIHT	13	0.046	0.201	18	0.010	0.019	20	0.000	(0.000;0.000)
20	dffbVIHT	4	0.422	0.196	4	0.410	0.177	4	0.123	(0.109;0.138)
21	dffbLIHT	18	0.028	0.096	23	0.006	0.009	24	0.000	(0.000;0.000)
22	iafbIntDr	5	0.214	0.427	6	0.150	0.207	7	0.021	(0.018;0.024)
23	dffbIntDr	1	0.675	0.490	1	0.649	0.344	2	0.239	(0.214;0.267)
24	iafbWDr	22	0.026	0.096	25	0.004	0.008	27	0.000	(0.000;0.000)
25	dffbWDr	23	0.022	0.049	19	0.010	0.011	21	0.000	(0.000;0.000)
26	transWHT	20	0.026	0.056	16	0.015	0.020	16	0.000	(0.000;0.001)
27	tQuench	7	0.146	0.085	7	0.131	0.058	6	0.022	(0.020;0.025)

Table B.4: Parameters importance ranking with respect to the average clad temperature output at $z \approx 2.4$ [m] (TC4)

No.	Parameter	Morris Radial			Morris Trajectory			Sobol'-Saltelli		
		Rank	μ_d^*	σ_d	Rank	μ_d^*	σ_d	Rank	$\hat{S}T_d$	95%CI _{pct}
1	breakP	11	0.108	0.083	10	0.098	0.065	10	0.014	(0.012;0.015)
2	fillT	17	0.038	0.077	13	0.023	0.024	14	0.001	(0.001;0.001)
3	fillV	6	0.193	0.076	5	0.189	0.064	5	0.039	(0.035;0.044)
4	pwr	9	0.116	0.109	9	0.109	0.056	9	0.015	(0.013;0.017)
5	nicK	25	0.028	0.145	26	0.005	0.007	25	0.000	(0.000;0.000)
6	nicCP	15	0.040	0.089	12	0.027	0.021	12	0.001	(0.001;0.001)
7	nicEM	27	0.013	0.028	24	0.007	0.010	21	0.000	(0.000;0.000)
8	mgoK	23	0.030	0.122	20	0.011	0.011	20	0.000	(0.000;0.000)
9	mgoCP	10	0.109	0.134	11	0.086	0.035	11	0.009	(0.008;0.010)
10	vesEps	26	0.017	0.057	27	0.005	0.008	26	0.000	(0.000;0.000)
11	ssK	16	0.039	0.155	22	0.008	0.011	23	0.000	(0.000;0.000)
12	ssCp	12	0.057	0.177	14	0.022	0.026	13	0.001	(0.001;0.001)
13	ssEm	22	0.030	0.074	19	0.012	0.016	17	0.000	(0.000;0.000)
14	gridK	24	0.028	0.041	15	0.019	0.012	16	0.001	(0.000;0.001)
15	gridHT	2	0.543	0.412	1	0.618	0.246	1	0.459	(0.410;0.515)
16	iafbWHT	7	0.172	0.158	8	0.152	0.088	8	0.027	(0.024;0.031)
17	dffbWHT	3	0.473	0.275	3	0.454	0.224	2	0.207	(0.184;0.233)
18	iafbVIHT	14	0.049	0.181	18	0.014	0.019	19	0.000	(0.000;0.000)
19	iafbLIHT	13	0.055	0.192	17	0.015	0.027	18	0.000	(0.000;0.000)
20	dffbVIHT	4	0.293	0.174	4	0.287	0.124	4	0.062	(0.055;0.070)
21	dffbLIHT	19	0.037	0.130	23	0.008	0.013	24	0.000	(0.000;0.000)
22	iafbIntDr	5	0.255	0.538	6	0.176	0.245	7	0.028	(0.024;0.033)
23	dffbIntDr	1	0.618	0.600	2	0.571	0.288	3	0.185	(0.166;0.207)
24	iafbWDr	18	0.037	0.143	25	0.005	0.011	27	0.000	(0.000;0.000)
25	dffbWDr	21	0.031	0.074	21	0.010	0.014	22	0.000	(0.000;0.000)
26	transWHT	20	0.035	0.087	16	0.018	0.023	15	0.001	(0.001;0.001)
27	tQuench	8	0.166	0.109	7	0.154	0.079	6	0.031	(0.027;0.035)

Table B.5: Parameters importance ranking with respect to the average clad temperature output at $z \approx 1.9$ [m] (TC5)

No.	Parameter	Morris Radial			Morris Trajectory			Sobol'-Saltelli		
		Rank	μ_d^*	σ_d	Rank	μ_d^*	σ_d	Rank	$\hat{S}T_d$	95%CI _{pet}
1	breakP	11	0.109	0.097	10	0.097	0.057	10	0.012	(0.011;0.014)
2	fillT	14	0.053	0.083	12	0.039	0.028	12	0.002	(0.002;0.002)
3	fillV	5	0.221	0.086	4	0.218	0.061	4	0.050	(0.045;0.056)
4	pwr	9	0.128	0.108	9	0.119	0.050	9	0.017	(0.015;0.019)
5	nicK	21	0.037	0.246	26	0.006	0.009	25	0.000	(0.000;0.000)
6	nicCP	16	0.050	0.152	13	0.031	0.025	14	0.001	(0.001;0.002)
7	nicEM	27	0.014	0.038	24	0.009	0.012	22	0.000	(0.000;0.000)
8	mgoK	22	0.037	0.131	19	0.013	0.014	20	0.000	(0.000;0.000)
9	mgoCP	10	0.122	0.184	11	0.095	0.036	11	0.010	(0.009;0.011)
10	vesEps	26	0.016	0.049	27	0.006	0.010	26	0.000	(0.000;0.000)
11	ssK	19	0.041	0.149	21	0.010	0.014	21	0.000	(0.000;0.000)
12	ssCp	13	0.056	0.142	14	0.025	0.031	13	0.001	(0.001;0.002)
13	ssEm	25	0.029	0.069	20	0.012	0.017	19	0.000	(0.000;0.000)
14	gridK	24	0.029	0.088	18	0.015	0.015	17	0.000	(0.000;0.001)
15	gridHT	1	0.593	0.452	1	0.673	0.254	1	0.539	(0.483;0.599)
16	iafbWHT	6	0.195	0.182	6	0.186	0.096	5	0.035	(0.031;0.040)
17	dffbWHT	3	0.440	0.265	3	0.419	0.184	2	0.176	(0.157;0.199)
18	iafbVIHT	15	0.051	0.221	17	0.015	0.019	18	0.000	(0.000;0.000)
19	iafbLIHT	12	0.082	0.320	16	0.020	0.033	16	0.000	(0.000;0.001)
20	dffbVIHT	7	0.180	0.137	7	0.172	0.092	8	0.024	(0.021;0.027)
21	dffbLIHT	18	0.045	0.186	23	0.010	0.015	24	0.000	(0.000;0.000)
22	iafbIntDr	4	0.290	0.541	5	0.197	0.258	6	0.034	(0.029;0.039)
23	dffbIntDr	2	0.512	0.453	2	0.487	0.251	3	0.125	(0.112;0.140)
24	iafbWDr	17	0.046	0.203	25	0.006	0.011	27	0.000	(0.000;0.000)
25	dffbWDr	23	0.034	0.087	22	0.010	0.014	23	0.000	(0.000;0.000)
26	transWHT	20	0.040	0.093	15	0.022	0.028	15	0.001	(0.001;0.001)
27	tQuench	8	0.171	0.125	8	0.158	0.085	7	0.032	(0.028;0.037)

Table B.6: Parameters importance ranking with respect to the average clad temperature output at $z \approx 1.3$ [m] (TC6)

No.	Parameter	Morris Radial			Morris Trajectory			Sobol'-Saltelli		
		Rank	μ_d^*	σ_d	Rank	μ_d^*	σ_d	Rank	$\hat{S}T_d$	95%CI _{pct}
1	breakP	10	0.111	0.101	10	0.094	0.052	10	0.011	(0.010;0.013)
2	fillT	13	0.091	0.102	12	0.082	0.027	11	0.007	(0.006;0.008)
3	fillV	5	0.267	0.084	4	0.264	0.061	3	0.072	(0.065;0.080)
4	pwr	9	0.135	0.068	8	0.133	0.047	8	0.019	(0.017;0.021)
5	nicK	23	0.033	0.108	25	0.007	0.011	24	0.000	(0.000;0.000)
6	nicCP	15	0.062	0.183	13	0.034	0.026	13	0.002	(0.001;0.002)
7	nicEM	27	0.017	0.043	24	0.010	0.012	25	0.000	(0.000;0.000)
8	mgoK	20	0.041	0.142	19	0.016	0.016	18	0.000	(0.000;0.000)
9	mgoCP	8	0.137	0.194	9	0.108	0.045	9	0.013	(0.012;0.015)
10	vesEps	26	0.021	0.061	26	0.007	0.011	26	0.000	(0.000;0.000)
11	ssK	19	0.044	0.150	20	0.013	0.017	20	0.000	(0.000;0.000)
12	ssCp	14	0.071	0.200	14	0.028	0.033	14	0.001	(0.001;0.002)
13	ssEm	25	0.028	0.064	21	0.013	0.017	21	0.000	(0.000;0.000)
14	gridK	21	0.038	0.135	17	0.019	0.017	17	0.001	(0.000;0.001)
15	gridHT	1	0.636	0.489	1	0.726	0.267	1	0.611	(0.552;0.678)
16	iafbWHT	6	0.222	0.167	5	0.219	0.108	5	0.045	(0.040;0.051)
17	dffbWHT	3	0.356	0.193	3	0.352	0.153	2	0.129	(0.115;0.146)
18	iafbVIHT	17	0.054	0.224	18	0.016	0.019	19	0.000	(0.000;0.001)
19	iafbLIHT	12	0.104	0.651	16	0.025	0.042	16	0.001	(0.001;0.001)
20	dffbVIHT	11	0.110	0.208	11	0.084	0.067	12	0.006	(0.005;0.007)
21	dffbLIHT	16	0.059	0.239	22	0.012	0.019	22	0.000	(0.000;0.000)
22	iafbIntDr	4	0.279	0.401	6	0.194	0.250	6	0.034	(0.029;0.040)
23	dffbIntDr	2	0.377	0.346	2	0.362	0.195	4	0.068	(0.061;0.076)
24	iafbWDr	24	0.031	0.100	27	0.006	0.013	27	0.000	(0.000;0.000)
25	dffbWDr	22	0.036	0.107	23	0.010	0.016	23	0.000	(0.000;0.000)
26	transWHT	18	0.048	0.095	15	0.025	0.033	15	0.001	(0.001;0.001)
27	tQuench	7	0.178	0.168	7	0.158	0.089	7	0.031	(0.027;0.035)

Table B.7: Parameters importance ranking with respect to the average clad temperature output at $z \approx 0.8$ [m] (TC7)

No.	Parameter	Morris Radial			Morris Trajectory			Sobol'-Saltelli		
		Rank	μ_d^*	σ_d	Rank	μ_d^*	σ_d	Rank	$\hat{S}T_d$	95%CI _{pet}
1	breakP	11	0.089	0.109	11	0.072	0.042	11	0.006	(0.006;0.007)
2	fillT	8	0.169	0.110	7	0.170	0.023	6	0.027	(0.024;0.030)
3	fillV	2	0.254	0.085	2	0.257	0.043	2	0.068	(0.061;0.075)
4	pwr	9	0.126	0.068	9	0.122	0.034	9	0.015	(0.014;0.017)
5	nicK	17	0.060	0.476	24	0.008	0.013	23	0.000	(0.000;0.000)
6	nicCP	16	0.072	0.339	13	0.032	0.019	13	0.001	(0.001;0.002)
7	nicEM	27	0.015	0.038	23	0.009	0.011	24	0.000	(0.000;0.000)
8	mgoK	20	0.057	0.183	17	0.018	0.018	17	0.001	(0.001;0.001)
9	mgoCP	10	0.125	0.120	10	0.108	0.041	10	0.013	(0.011;0.014)
10	vesEps	26	0.019	0.041	25	0.008	0.013	25	0.000	(0.000;0.000)
11	ssK	19	0.058	0.204	21	0.012	0.016	19	0.001	(0.000;0.001)
12	ssCp	18	0.059	0.175	16	0.024	0.024	15	0.001	(0.001;0.001)
13	ssEm	25	0.023	0.065	26	0.007	0.013	26	0.000	(0.000;0.000)
14	gridK	23	0.041	0.109	18	0.016	0.018	18	0.001	(0.000;0.001)
15	gridHT	1	0.674	0.518	1	0.762	0.301	1	0.705	(0.642;0.774)
16	iafbWHT	3	0.238	0.237	3	0.227	0.115	4	0.050	(0.044;0.056)
17	dffbWHT	4	0.236	0.190	4	0.226	0.130	3	0.062	(0.055;0.071)
18	iafbVIHT	12	0.085	0.922	20	0.014	0.022	20	0.000	(0.000;0.001)
19	iafbLIHT	15	0.076	0.241	15	0.027	0.040	16	0.001	(0.001;0.001)
20	dffbVIHT	14	0.083	0.350	12	0.034	0.045	14	0.001	(0.001;0.002)
21	dffbLIHT	13	0.085	0.393	19	0.015	0.023	21	0.000	(0.000;0.000)
22	iafbIntDr	5	0.236	0.472	8	0.142	0.178	8	0.021	(0.018;0.024)
23	dffbIntDr	6	0.215	0.288	6	0.185	0.116	7	0.021	(0.018;0.023)
24	iafbWDr	24	0.031	0.104	27	0.006	0.011	27	0.000	(0.000;0.000)
25	dffbWDr	22	0.050	0.174	22	0.011	0.018	22	0.000	(0.000;0.000)
26	transWHT	21	0.056	0.110	14	0.030	0.035	12	0.002	(0.001;0.002)
27	tQuench	7	0.213	0.226	5	0.191	0.095	5	0.044	(0.039;0.050)

Table B.8: Parameters importance ranking with respect to the average clad temperature output at $z \approx 0.3$ [m] (TC8)

No.	Parameter	Morris Radial			Morris Trajectory			Sobol'-Saltelli		
		Rank	μ_d^*	σ_d	Rank	μ_d^*	σ_d	Rank	$\hat{S}T_d$	95%CI _{pct}
1	breakP	9	0.091	0.173	10	0.045	0.042	10	0.003	(0.003;0.004)
2	fillT	1	0.749	0.256	1	0.774	0.031	1	0.549	(0.497;0.606)
3	fillV	3	0.270	0.134	3	0.272	0.045	3	0.072	(0.065;0.081)
4	pwr	6	0.219	0.037	4	0.230	0.012	4	0.047	(0.043;0.053)
5	nicK	22	0.040	0.121	22	0.013	0.019	21	0.001	(0.001;0.001)
6	nicCP	13	0.077	0.321	12	0.036	0.024	11	0.002	(0.002;0.002)
7	nicEM	27	0.009	0.018	25	0.006	0.008	24	0.000	(0.000;0.000)
8	mgoK	18	0.054	0.093	15	0.031	0.021	13	0.002	(0.001;0.002)
9	mgoCP	7	0.131	0.255	7	0.095	0.040	7	0.010	(0.009;0.011)
10	vesEps	23	0.032	0.238	23	0.010	0.019	23	0.000	(0.000;0.000)
11	ssK	20	0.045	0.149	21	0.014	0.023	22	0.001	(0.001;0.001)
12	ssCp	14	0.069	0.252	16	0.026	0.022	17	0.001	(0.001;0.001)
13	ssEm	26	0.011	0.031	26	0.004	0.006	26	0.000	(0.000;0.000)
14	gridK	21	0.045	0.104	20	0.015	0.027	20	0.001	(0.001;0.001)
15	gridHT	19	0.050	0.116	18	0.025	0.035	16	0.001	(0.001;0.002)
16	iafbWHT	4	0.225	0.714	5	0.180	0.095	5	0.033	(0.030;0.038)
17	dffbWHT	8	0.107	0.156	8	0.082	0.068	8	0.009	(0.007;0.010)
18	iafbVIHT	24	0.024	0.097	24	0.008	0.014	25	0.000	(0.000;0.000)
19	iafbLIHT	11	0.085	0.192	11	0.038	0.054	14	0.002	(0.001;0.002)
20	dffbVIHT	17	0.057	0.146	14	0.032	0.042	12	0.002	(0.001;0.002)
21	dffbLIHT	10	0.087	0.255	17	0.025	0.038	19	0.001	(0.001;0.001)
22	iafbIntDr	5	0.224	0.694	6	0.113	0.132	6	0.012	(0.010;0.013)
23	dffbIntDr	16	0.062	0.174	13	0.033	0.054	15	0.001	(0.001;0.002)
24	iafbWDr	25	0.012	0.037	27	0.003	0.006	27	0.000	(0.000;0.000)
25	dffbWDr	15	0.065	0.194	19	0.016	0.029	18	0.001	(0.001;0.001)
26	transWHT	12	0.084	0.169	9	0.058	0.047	9	0.005	(0.004;0.005)
27	tQuench	2	0.533	0.338	2	0.512	0.134	2	0.272	(0.244;0.303)

Table B.9: Parameters importance ranking with respect to the average bottom pressure drop output (DP Bot., the segment between $z = 0.0$ [m] and $z = 1.7$ [m])

No.	Parameter	Morris Radial			Morris Trajectory			Sobol'-Saltelli		
		Rank	μ_d^*	σ_d	Rank	μ_d^*	σ_d	Rank	$\hat{S}T_d$	95%CI _{pet}
1	breakP	6	0.181	0.095	7	0.176	0.049	6	0.032	(0.028;0.036)
2	fillT	7	0.173	0.096	6	0.178	0.030	7	0.029	(0.026;0.033)
3	fillV	2	0.460	0.094	2	0.465	0.067	2	0.213	(0.191;0.238)
4	pwr	11	0.125	0.071	11	0.129	0.039	11	0.016	(0.014;0.018)
5	nicK	21	0.045	0.298	26	0.007	0.011	25	0.000	(0.000;0.000)
6	nicCP	16	0.051	0.098	15	0.038	0.021	15	0.002	(0.001;0.002)
7	nicEM	27	0.015	0.041	25	0.008	0.011	26	0.000	(0.000;0.000)
8	mgoK	23	0.039	0.151	23	0.010	0.013	21	0.000	(0.000;0.000)
9	mgoCP	8	0.151	0.155	9	0.136	0.048	9	0.019	(0.017;0.022)
10	vesEps	26	0.019	0.055	27	0.006	0.010	27	0.000	(0.000;0.000)
11	ssK	22	0.043	0.135	21	0.011	0.014	20	0.000	(0.000;0.000)
12	ssCp	12	0.099	0.142	12	0.083	0.027	12	0.007	(0.006;0.008)
13	ssEm	25	0.025	0.046	20	0.011	0.014	22	0.000	(0.000;0.000)
14	gridK	20	0.045	0.081	18	0.029	0.016	17	0.001	(0.001;0.001)
15	gridHT	4	0.209	0.303	4	0.224	0.263	3	0.076	(0.065;0.087)
16	iafbWHT	9	0.139	0.200	8	0.141	0.135	10	0.019	(0.017;0.023)
17	dffbWHT	10	0.137	0.158	10	0.135	0.124	8	0.026	(0.022;0.030)
18	iafbVIHT	19	0.047	0.217	19	0.014	0.018	19	0.000	(0.000;0.000)
19	iafbLIHT	14	0.082	0.316	16	0.030	0.042	18	0.001	(0.001;0.001)
20	dffbVIHT	13	0.084	0.170	13	0.062	0.064	13	0.004	(0.003;0.005)
21	dffbLIHT	18	0.048	0.156	22	0.011	0.016	23	0.000	(0.000;0.000)
22	iafbIntDr	1	0.976	0.794	1	0.876	0.636	1	0.531	(0.475;0.593)
23	dffbIntDr	3	0.382	0.374	3	0.364	0.203	4	0.066	(0.059;0.075)
24	iafbWDr	24	0.038	0.164	24	0.009	0.011	24	0.000	(0.000;0.000)
25	dffbWDr	15	0.065	0.094	14	0.050	0.020	14	0.003	(0.002;0.003)
26	transWHT	17	0.049	0.098	17	0.029	0.035	16	0.001	(0.001;0.002)
27	tQuench	5	0.203	0.206	5	0.190	0.165	5	0.050	(0.042;0.058)

Table B.10: Parameters importance ranking with respect to the average middle pressure drop output (DP Mid., the segment between $z = 1.7$ [m] and $z = 2.3$ [m])

No.	Parameter	Morris Radial			Morris Trajectory			Sobol'-Saltelli		
		Rank	μ_d^*	σ_d	Rank	μ_d^*	σ_d	Rank	$\hat{S}T_d$	95%CI _{pct}
1	breakP	6	0.223	0.091	7	0.220	0.059	7	0.051	(0.045;0.057)
2	fillT	9	0.186	0.084	10	0.181	0.052	9	0.035	(0.031;0.040)
3	fillV	2	0.548	0.146	2	0.541	0.093	1	0.298	(0.268;0.333)
4	pwr	5	0.230	0.076	5	0.238	0.070	6	0.056	(0.050;0.063)
5	nicK	20	0.037	0.320	26	0.005	0.008	25	0.000	(0.000;0.000)
6	nicCP	18	0.041	0.088	16	0.028	0.020	16	0.001	(0.001;0.001)
7	nicEM	27	0.012	0.028	24	0.007	0.010	23	0.000	(0.000;0.000)
8	mgoK	25	0.023	0.080	21	0.008	0.011	22	0.000	(0.000;0.000)
9	mgoCP	12	0.112	0.148	12	0.090	0.040	12	0.010	(0.009;0.011)
10	vesEps	26	0.015	0.043	27	0.004	0.007	27	0.000	(0.000;0.000)
11	ssK	23	0.035	0.119	22	0.008	0.011	21	0.000	(0.000;0.000)
12	ssCp	13	0.096	0.165	14	0.074	0.027	14	0.006	(0.006;0.007)
13	ssEm	24	0.025	0.044	20	0.014	0.014	18	0.000	(0.000;0.000)
14	gridK	16	0.048	0.079	15	0.037	0.012	15	0.001	(0.001;0.002)
15	gridHT	7	0.219	0.289	4	0.240	0.238	4	0.086	(0.073;0.099)
16	iafbWHT	10	0.179	0.154	9	0.184	0.108	10	0.033	(0.029;0.038)
17	dffbWHT	8	0.203	0.175	8	0.202	0.132	8	0.050	(0.043;0.057)
18	iafbVIHT	17	0.046	0.207	19	0.016	0.019	19	0.000	(0.000;0.000)
19	iafbLIHT	15	0.061	0.237	18	0.017	0.027	20	0.000	(0.000;0.000)
20	dffbVIHT	11	0.175	0.209	11	0.159	0.097	11	0.022	(0.019;0.025)
21	dffbLIHT	21	0.036	0.141	23	0.008	0.012	24	0.000	(0.000;0.000)
22	iafbIntDr	1	0.668	0.692	1	0.549	0.523	2	0.250	(0.219;0.284)
23	dffbIntDr	3	0.510	0.448	3	0.477	0.259	3	0.120	(0.107;0.134)
24	iafbWDr	19	0.041	0.194	25	0.005	0.009	26	0.000	(0.000;0.000)
25	dffbWDr	14	0.085	0.072	13	0.080	0.019	13	0.007	(0.006;0.008)
26	transWHT	22	0.036	0.067	17	0.024	0.029	17	0.001	(0.001;0.001)
27	tQuench	4	0.234	0.155	6	0.231	0.133	5	0.068	(0.059;0.078)

Table B.11: Parameters importance ranking with respect to the average top pressure drop output (DP Top, the segment between $z = 2.3$ [m] and $z = 4.1$ [m])

No.	Parameter	Morris Radial			Morris Trajectory			Sobol'-Saltelli		
		Rank	μ_d^*	σ_d	Rank	μ_d^*	σ_d	Rank	$\hat{S}T_d$	95%CI _{pct}
1	breakP	10	0.147	0.067	10	0.140	0.052	10	0.023	(0.020;0.026)
2	fillT	12	0.116	0.058	12	0.112	0.045	12	0.015	(0.013;0.016)
3	fillV	2	0.427	0.099	2	0.418	0.091	2	0.186	(0.167;0.209)
4	pwr	9	0.184	0.096	9	0.179	0.060	9	0.034	(0.030;0.038)
5	nicK	25	0.016	0.045	27	0.004	0.006	25	0.000	(0.000;0.000)
6	nicCP	21	0.028	0.041	17	0.018	0.013	17	0.001	(0.000;0.001)
7	nicEM	27	0.010	0.020	24	0.005	0.007	22	0.000	(0.000;0.000)
8	mgoK	23	0.022	0.096	23	0.005	0.008	23	0.000	(0.000;0.000)
9	mgoCP	14	0.081	0.098	14	0.063	0.032	14	0.005	(0.005;0.006)
10	vesEps	26	0.012	0.031	26	0.004	0.006	26	0.000	(0.000;0.000)
11	ssK	20	0.030	0.119	21	0.006	0.008	21	0.000	(0.000;0.000)
12	ssCp	15	0.063	0.136	15	0.041	0.019	15	0.002	(0.002;0.002)
13	ssEm	24	0.018	0.033	20	0.009	0.011	18	0.000	(0.000;0.000)
14	gridK	13	0.109	0.030	13	0.110	0.019	13	0.012	(0.011;0.014)
15	gridHT	7	0.280	0.257	6	0.313	0.203	3	0.143	(0.125;0.164)
16	iafbWHT	11	0.134	0.108	11	0.125	0.065	11	0.016	(0.014;0.019)
17	dffbWHT	6	0.292	0.194	7	0.283	0.152	6	0.086	(0.075;0.099)
18	iafbVIHT	17	0.036	0.132	18	0.012	0.014	19	0.000	(0.000;0.000)
19	iafbLIHT	16	0.048	0.249	19	0.009	0.015	20	0.000	(0.000;0.000)
20	dffbVIHT	4	0.333	0.175	4	0.324	0.152	7	0.086	(0.075;0.098)
21	dffbLIHT	22	0.024	0.066	22	0.005	0.010	24	0.000	(0.000;0.000)
22	iafbIntDr	3	0.416	0.520	3	0.327	0.355	5	0.096	(0.082;0.112)
23	dffbIntDr	1	0.670	0.545	1	0.624	0.342	1	0.204	(0.182;0.230)
24	iafbWDr	19	0.034	0.155	25	0.004	0.008	27	0.000	(0.000;0.000)
25	dffbWDr	5	0.322	0.069	5	0.323	0.040	4	0.110	(0.098;0.123)
26	transWHT	18	0.035	0.065	16	0.024	0.029	16	0.001	(0.001;0.001)
27	tQuench	8	0.188	0.112	8	0.181	0.101	8	0.042	(0.037;0.048)

Table B.12: Parameters importance ranking with respect to the average total pressure drop output (DP Tot., the segment between $z = 0.0$ [m] and $z = 4.1$ [m])

No.	Parameter	Morris Radial			Morris Trajectory			Sobol'-Saltelli		
		Rank	μ_d^*	σ_d	Rank	μ_d^*	σ_d	Rank	$\hat{S}T_d$	95%CI _{pct}
1	breakP	10	0.178	0.064	10	0.173	0.034	10	0.031	(0.028;0.035)
2	fillT	11	0.150	0.056	11	0.148	0.022	11	0.022	(0.020;0.025)
3	fillV	3	0.484	0.076	3	0.477	0.043	1	0.231	(0.208;0.258)
4	pwr	9	0.189	0.063	9	0.190	0.032	9	0.035	(0.031;0.039)
5	nicK	23	0.025	0.157	26	0.004	0.006	25	0.000	(0.000;0.000)
6	nicCP	19	0.034	0.046	16	0.025	0.015	17	0.001	(0.001;0.001)
7	nicEM	27	0.010	0.024	24	0.006	0.007	23	0.000	(0.000;0.000)
8	mgoK	24	0.023	0.077	22	0.006	0.009	22	0.000	(0.000;0.000)
9	mgoCP	13	0.104	0.108	13	0.088	0.034	13	0.009	(0.008;0.010)
10	vesEps	26	0.013	0.037	27	0.004	0.006	27	0.000	(0.000;0.000)
11	ssK	20	0.031	0.117	21	0.006	0.008	21	0.000	(0.000;0.000)
12	ssCp	15	0.076	0.112	15	0.060	0.020	15	0.004	(0.004;0.005)
13	ssEm	25	0.020	0.036	20	0.010	0.011	20	0.000	(0.000;0.000)
14	gridK	14	0.083	0.036	14	0.081	0.016	14	0.006	(0.006;0.007)
15	gridHT	6	0.245	0.272	4	0.276	0.224	4	0.114	(0.098;0.132)
16	iafbWHT	12	0.144	0.119	12	0.143	0.083	12	0.021	(0.018;0.023)
17	dffbWHT	5	0.245	0.174	5	0.244	0.139	5	0.066	(0.057;0.076)
18	iafbVIHT	17	0.038	0.161	19	0.013	0.015	19	0.000	(0.000;0.000)
19	iafbLIHT	16	0.050	0.182	18	0.014	0.021	18	0.000	(0.000;0.000)
20	dffbVIHT	4	0.245	0.139	6	0.242	0.122	8	0.048	(0.042;0.055)
21	dffbLIHT	22	0.027	0.078	23	0.006	0.010	24	0.000	(0.000;0.000)
22	iafbIntDr	1	0.610	0.618	2	0.514	0.465	2	0.209	(0.183;0.238)
23	dffbIntDr	2	0.602	0.500	1	0.564	0.303	3	0.164	(0.146;0.184)
24	iafbWDr	21	0.030	0.133	25	0.005	0.008	26	0.000	(0.000;0.000)
25	dffbWDr	7	0.219	0.058	7	0.224	0.031	7	0.051	(0.046;0.058)
26	transWHT	18	0.036	0.067	17	0.025	0.029	16	0.001	(0.001;0.001)
27	tQuench	8	0.205	0.133	8	0.201	0.118	6	0.051	(0.045;0.059)

Table B.13: Parameters importance ranking with respect to the average liquid carryover output (CO)

No.	Parameter	Morris Radial			Morris Trajectory			Sobol'-Saltelli		
		Rank	μ_d^*	σ_d	Rank	μ_d^*	σ_d	Rank	$\hat{S}T_d$	95%CI _{pct}
1	breakP	8	0.091	0.046	8	0.089	0.027	8	0.008	(0.008;0.010)
2	fillT	17	0.021	0.045	14	0.012	0.011	15	0.000	(0.000;0.000)
3	fillV	3	0.628	0.070	2	0.644	0.025	1	0.396	(0.358;0.439)
4	pwr	14	0.025	0.052	11	0.020	0.025	12	0.000	(0.000;0.001)
5	nicK	22	0.016	0.084	27	0.003	0.004	25	0.000	(0.000;0.000)
6	nicCP	23	0.015	0.028	19	0.008	0.010	16	0.000	(0.000;0.000)
7	nicEM	27	0.006	0.013	24	0.004	0.005	23	0.000	(0.000;0.000)
8	mgoK	18	0.018	0.101	23	0.004	0.006	22	0.000	(0.000;0.000)
9	mgoCP	11	0.037	0.078	12	0.020	0.025	11	0.001	(0.001;0.001)
10	vesEps	26	0.009	0.021	26	0.003	0.004	27	0.000	(0.000;0.000)
11	ssK	20	0.017	0.066	21	0.005	0.006	21	0.000	(0.000;0.000)
12	ssCp	13	0.027	0.103	15	0.012	0.014	14	0.000	(0.000;0.000)
13	ssEm	25	0.012	0.022	20	0.006	0.008	20	0.000	(0.000;0.000)
14	gridK	10	0.045	0.066	10	0.040	0.011	10	0.002	(0.001;0.002)
15	gridHT	5	0.178	0.158	4	0.196	0.131	4	0.055	(0.049;0.063)
16	iafbWHT	9	0.087	0.070	9	0.083	0.038	9	0.007	(0.006;0.007)
17	dffbWHT	6	0.153	0.117	6	0.150	0.075	5	0.023	(0.020;0.026)
18	iafbVIHT	15	0.024	0.111	18	0.009	0.010	19	0.000	(0.000;0.000)
19	iafbLIHT	12	0.035	0.207	17	0.009	0.014	18	0.000	(0.000;0.000)
20	dffbVIHT	2	0.645	0.252	3	0.637	0.196	2	0.282	(0.254;0.313)
21	dffbLIHT	24	0.015	0.051	22	0.004	0.006	24	0.000	(0.000;0.000)
22	iafbIntDr	4	0.203	0.364	5	0.150	0.169	6	0.019	(0.016;0.021)
23	dffbIntDr	1	0.678	0.468	1	0.668	0.341	3	0.229	(0.206;0.255)
24	iafbWDr	21	0.017	0.060	25	0.003	0.005	26	0.000	(0.000;0.000)
25	dffbWDr	19	0.018	0.036	16	0.010	0.010	17	0.000	(0.000;0.000)
26	transWHT	16	0.022	0.046	13	0.013	0.017	13	0.000	(0.000;0.000)
27	tQuench	7	0.123	0.064	7	0.121	0.047	7	0.016	(0.014;0.018)

B.3 CONVERGENCE OF THE SOBOL' INDICES

The convergence of the Sobol' indices' estimator can be investigated from their evolutions as functions of the number of samples. Shown in Fig. B.16 is the evolution (trace plot) of the estimated main-effect indices with the maximum clad temperature as the QoI. The Saltelli et al. estimator performs poorly compared to the Janon et al. estimator for this particular output. This means that a larger number of samples are required to obtain a stable ranking.

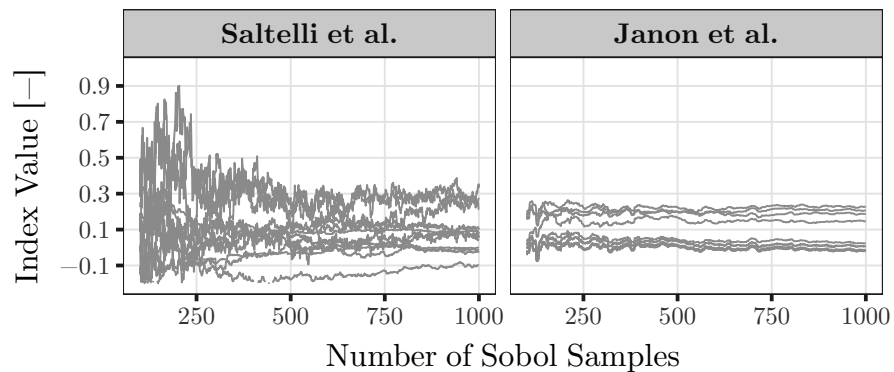


Figure B.16: Evolution of the main-effect sensitivity indices for all input parameters using two different estimators as a function of the number of Sobol' sequence samples. The QoI is the maximum clad temperature

If the main purpose of the SA is simply to rank the parameter importance with respect to a particular QoI, then the Janon et al. estimator visibly requires fewer samples and the ranking can be reliably constructed. However, the apparent stabilization of the indices' estimator is not sufficient to establish a robust estimate of the indices since MC estimation entails uncertainty due to finite number of samples. Such uncertainty needs to be addressed for all the estimates.

In this work, an empirical convergence study was established using three different sample sizes (250; 500; 1000), and for each size, the 95% CI length (the difference between the upper and lower bounds) is determined using the bootstrap technique [155] using 10'000 replications. The results are shown in Fig. B.17 for the maximum clad temperature as the QoI. Note that the comparisons between CI lengths of different estimates can be made directly as the Sobol' index itself is dimensionless. As can be seen, with respect to this QoI, the Janon et al. estimator is further confirmed as the more efficient estimator. The uncertainty of indices estimated by the Saltelli et al. estimator is still high for numbers of samples in the range of thousands. The efficiency of the Saltelli et al. estimator is also found to be more sensitive to the choice of estimand (i.e., Sobol' index of a given input parameter).

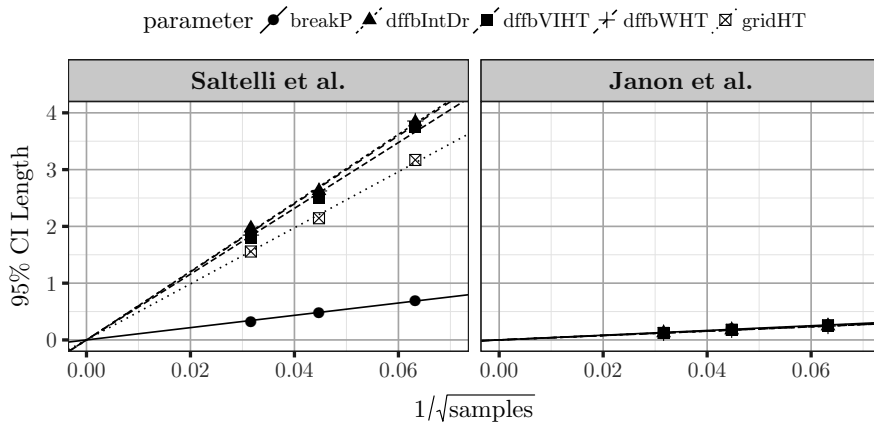


Figure B.17: The 95% percentile bootstrapped CI length as a function of the number of samples for five selected estimated Sobol' main-effect indices, with respect to the maximum clad temperature using two different estimators. The lines shown are the regression through the origin lines.

However, further investigation also revealed that the efficiency of an estimator depended on the QoI in a more complex manner than initially considered. As can be seen in Fig. B.18, where the first principal component scores were taken as the QoI, both estimators were found to be comparable, with the Saltelli et al. estimator being even slightly more efficient. And as before, the Janon et al. estimator shows less sensitivity to the choice of estimand in its convergence as compared to the Saltelli et al. estimator.

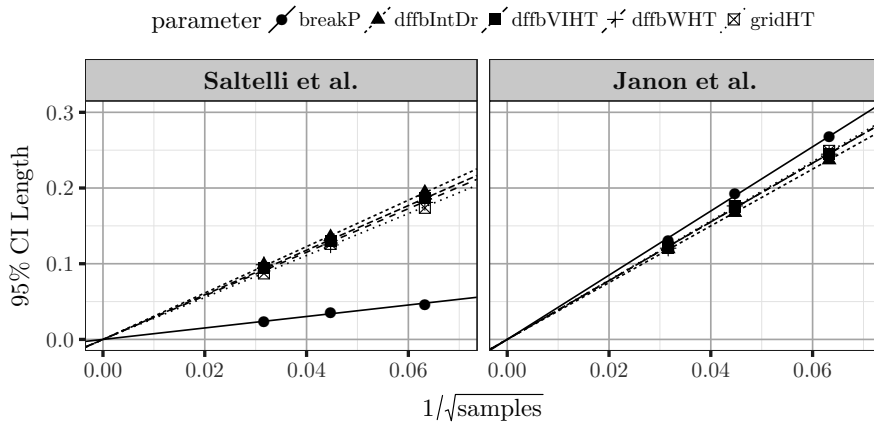


Figure B.18: The 95% percentile bootstrapped CI length as a function of the number of samples for five selected estimated Sobol' main-effect indices, with respect to the first principal component using two different estimators. The lines shown are the regression through the origin lines.

The convergence analysis plots shown in Figs. B.17 and B.18 can be useful in the planning of the simulation experiments. As can be inferred from both figures, the CI length of a given estimator depends on the QoI, the estimand, the estimator used, and the number of samples. The regression lines also indicate the projection of the reduction in the CI length with increasing number of samples.

As for the total-effect indices, the results obtained using the Jansen's estimator confirmed the good efficiency of the estimator, reaching below 10% CI length for 1'000 MC samples across all QoI.

B.4 SOBOL INDICES (12-PARAMETER MODEL)

Table B.14: Main-effect and total-effect sensitivity indices for 12-parameter FEBA model with respect to the maximum clad temperature at the mid-height of the assembly as the QoI.

No.	Parameter	S_d		ST_d	
		\hat{S}_d	95%CI _{pct}	$\hat{S}T_d$	95%CI _{pct}
1	breakP	-0.011	(-0.057;0.034)	0.008	(0.008;0.009)
2	fillT	-0.015	(-0.061;0.030)	0.001	(0.001;0.001)
3	fillV	0.028	(-0.016;0.071)	0.044	(0.040;0.047)
4	pwr	0.009	(-0.038;0.054)	0.027	(0.025;0.030)
5	gridHT	0.143	(0.099;0.186)	0.197	(0.179;0.215)
6	iafbWHT	-0.016	(-0.062;0.029)	0.008	(0.007;0.009)
7	dffbWHT	0.206	(0.163;0.248)	0.262	(0.241;0.286)
8	dffbVIHT	0.196	(0.153;0.239)	0.244	(0.224;0.265)
9	iafbIntDr	0.013	(-0.033;0.057)	0.031	(0.028;0.034)
10	dffbIntDr	0.223	(0.181;0.265)	0.287	(0.264;0.312)
11	dffbWDr	-0.013	(-0.059;0.032)	0.001	(0.001;0.001)
12	tQuench	-0.016	(-0.062;0.029)	0.007	(0.007;0.008)

Table B.15: Main-effect and total-effect sensitivity indices for 12-parameter FEBA model with respect to the time of quenching at the mid-height of the assembly as the QoI.

No.	Parameter	S_d		ST_d	
		\hat{S}_d	95%CI _{pct}	$\hat{S}T_d$	95%CI _{pct}
1	breakP	0.003	(-0.044;0.048)	0.015	(0.014;0.017)
2	fillT	-0.011	(-0.057;0.036)	0.001	(0.001;0.001)
3	fillV	0.018	(-0.027;0.063)	0.028	(0.026;0.031)
4	pwr	-0.001	(-0.048;0.045)	0.011	(0.010;0.012)
5	gridHT	0.501	(0.468;0.532)	0.530	(0.491;0.572)
6	iafbWHT	0.043	(-0.004;0.089)	0.063	(0.058;0.069)
7	dffbWHT	0.101	(0.057;0.145)	0.142	(0.129;0.155)
8	dffbVIHT	0.007	(-0.039;0.052)	0.023	(0.021;0.025)
9	iafbIntDr	0.032	(-0.015;0.079)	0.066	(0.060;0.074)
10	dffbIntDr	0.076	(0.033;0.120)	0.094	(0.086;0.101)
11	dffbWDr	-0.010	(-0.057;0.036)	0.000	(0.000;0.000)
12	tQuench	0.074	(0.028;0.120)	0.100	(0.091;0.109)

Table B.16: Main-effect and total-effect sensitivity indices for 12-parameter FEBA model with respect to the 1st fPC scores of the registered clad temperature transient at the mid-height of the assembly as the QoI.

No.	Parameter	S_d		ST_d	
		\hat{S}_d	95%CI _{pct}	$\hat{S}T_d$	95%CI _{pct}
1	breakP	0.000	(-0.008;0.008)	0.019	(0.017;0.022)
2	fillT	0.001	(-0.005;0.007)	0.011	(0.008;0.013)
3	fillV	0.031	(0.017;0.046)	0.051	(0.047;0.055)
4	pwr	0.023	(0.012;0.035)	0.035	(0.031;0.039)
5	gridHT	0.164	(0.134;0.195)	0.217	(0.199;0.236)
6	iafbWHT	0.001	(-0.010;0.012)	0.031	(0.028;0.035)
7	dffbWHT	0.225	(0.191;0.260)	0.268	(0.247;0.291)
8	dffbVIHT	0.219	(0.187;0.252)	0.278	(0.256;0.302)
9	iafbIntDr	0.036	(0.020;0.052)	0.062	(0.057;0.068)
10	dffbIntDr	0.212	(0.179;0.247)	0.269	(0.248;0.292)
11	dffbWDr	0.002	(-0.002;0.007)	0.006	(0.004;0.008)
12	tQuench	-0.002	(-0.013;0.009)	0.035	(0.031;0.039)

Table B.17: Main-effect and total-effect sensitivity indices for 12-parameter FEBA model with respect to the 2nd fPC scores of the registered clad temperature transient at the mid-height of the assembly as the QoI.

No.	Parameter	S_d		ST_d	
		\hat{S}_d	95%CI _{pct}	$\hat{S}T_d$	95%CI _{pct}
1	breakP	0.008	(−0.015;0.031)	0.163	(0.139;0.189)
2	fillT	0.000	(−0.019;0.019)	0.105	(0.085;0.126)
3	fillV	−0.014	(−0.042;0.014)	0.200	(0.174;0.228)
4	pwr	−0.006	(−0.031;0.020)	0.180	(0.153;0.210)
5	gridHT	0.079	(0.033;0.126)	0.590	(0.541;0.643)
6	iafbWHT	0.019	(−0.017;0.057)	0.374	(0.340;0.410)
7	dffbWHT	0.007	(−0.034;0.047)	0.494	(0.455;0.537)
8	dffbVIHT	−0.031	(−0.071;0.009)	0.434	(0.396;0.476)
9	iafbIntDr	0.033	(−0.001;0.069)	0.349	(0.313;0.389)
10	dffbIntDr	−0.010	(−0.059;0.037)	0.636	(0.591;0.687)
11	dffbWDr	0.007	(−0.005;0.020)	0.056	(0.042;0.071)
12	tQuench	0.112	(0.068;0.157)	0.525	(0.485;0.568)

Table B.18: Main-effect and total-effect sensitivity indices for 12-parameter FEBA model with respect to the 1st fPC scores of the warping function for the clad temperature transient at the mid-height of the assembly as the QoI.

No.	Parameter	S_d		ST_d	
		\hat{S}_d	95%CI _{pct}	$\hat{S}T_d$	95%CI _{pct}
1	breakP	0.012	(0.004;0.020)	0.016	(0.015;0.018)
2	fillT	−0.001	(−0.004;0.001)	0.001	(0.001;0.002)
3	fillV	0.028	(0.018;0.039)	0.031	(0.028;0.033)
4	pwr	0.008	(0.002;0.014)	0.012	(0.011;0.013)
5	gridHT	0.496	(0.447;0.547)	0.524	(0.486;0.564)
6	iafbWHT	0.042	(0.029;0.056)	0.050	(0.046;0.055)
7	dffbWHT	0.122	(0.098;0.147)	0.152	(0.140;0.166)
8	dffbVIHT	0.022	(0.011;0.032)	0.031	(0.028;0.034)
9	iafbIntDr	0.036	(0.021;0.050)	0.056	(0.050;0.062)
10	dffbIntDr	0.106	(0.085;0.128)	0.120	(0.111;0.130)
11	dffbWDr	0.000	(−0.001;0.001)	0.000	(0.000;0.000)
12	tQuench	0.065	(0.048;0.083)	0.081	(0.074;0.088)

Table B.19: Main-effect and total-effect sensitivity indices for 12-parameter FEBA model with respect to the 1st fPC scores of the pressure drop transient at the middle of the assembly as the QoI.

No.	Parameter	S_d		ST_d	
		\hat{S}_d	95%CI _{pct}	$\hat{S}T_d$	95%CI _{pct}
1	breakP	0.057	(0.042;0.072)	0.060	(0.056;0.066)
2	fillT	0.038	(0.026;0.050)	0.039	(0.036;0.042)
3	fillV	0.302	(0.267;0.341)	0.304	(0.282;0.329)
4	pwr	0.056	(0.041;0.071)	0.062	(0.057;0.067)
5	gridHT	0.019	(0.002;0.037)	0.078	(0.071;0.087)
6	iafbWHT	0.029	(0.017;0.041)	0.038	(0.035;0.042)
7	dffbWHT	0.019	(0.006;0.033)	0.048	(0.042;0.053)
8	dffbVIHT	0.021	(0.011;0.030)	0.022	(0.020;0.024)
9	iafbIntDr	0.195	(0.162;0.229)	0.278	(0.252;0.305)
10	dffbIntDr	0.097	(0.077;0.118)	0.106	(0.097;0.115)
11	dffbWDr	0.000	(-0.001;0.001)	0.000	(0.000;0.000)
12	tQuench	0.048	(0.031;0.065)	0.075	(0.068;0.083)

Table B.20: Main-effect and total-effect sensitivity indices for 12-parameter FEBA model with respect to the 1st fPC scores of the liquid carryover transient as the QoI.

No.	Parameter	S_d		ST_d	
		\hat{S}_d	95%CI _{pct}	$\hat{S}T_d$	95%CI _{pct}
1	breakP	0.000	(-0.002;0.001)	0.001	(0.001;0.001)
2	fillT	0.003	(-0.001;0.006)	0.003	(0.003;0.004)
3	fillV	0.901	(0.833;0.972)	0.907	(0.856;0.960)
4	pwr	0.005	(-0.001;0.011)	0.010	(0.009;0.011)
5	gridHT	-0.001	(-0.005;0.003)	0.005	(0.004;0.005)
6	iafbWHT	0.000	(-0.001;0.001)	0.000	(0.000;0.000)
7	dffbWHT	0.000	(-0.002;0.002)	0.001	(0.001;0.001)
8	dffbVIHT	0.048	(0.034;0.062)	0.055	(0.051;0.059)
9	iafbIntDr	0.000	(-0.002;0.002)	0.001	(0.001;0.002)
10	dffbIntDr	0.025	(0.015;0.036)	0.028	(0.026;0.031)
11	dffbWDr	0.000	(-0.001;0.000)	0.000	(0.000;0.000)
12	tQuench	-0.001	(-0.003;0.001)	0.001	(0.001;0.001)

B.5 GAUSSIAN PROCESS METAMODEL CONSTRUCTION

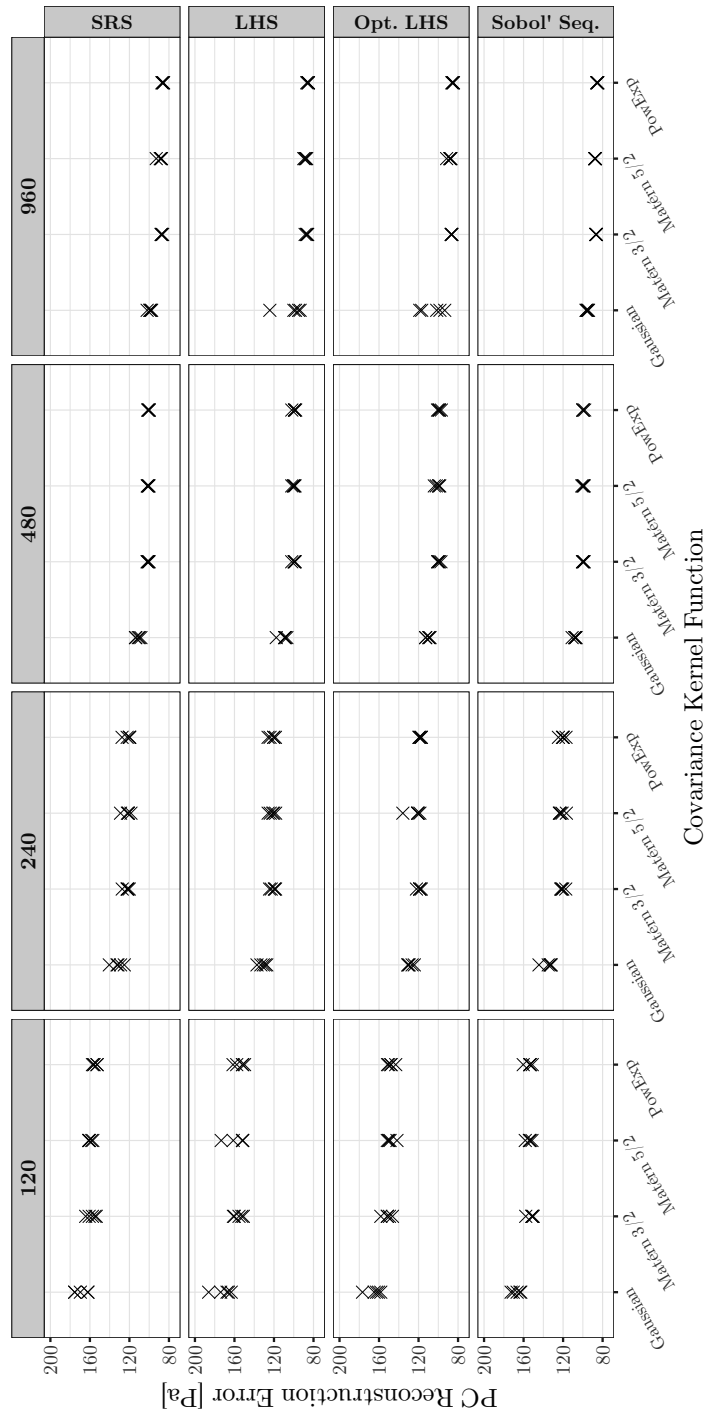
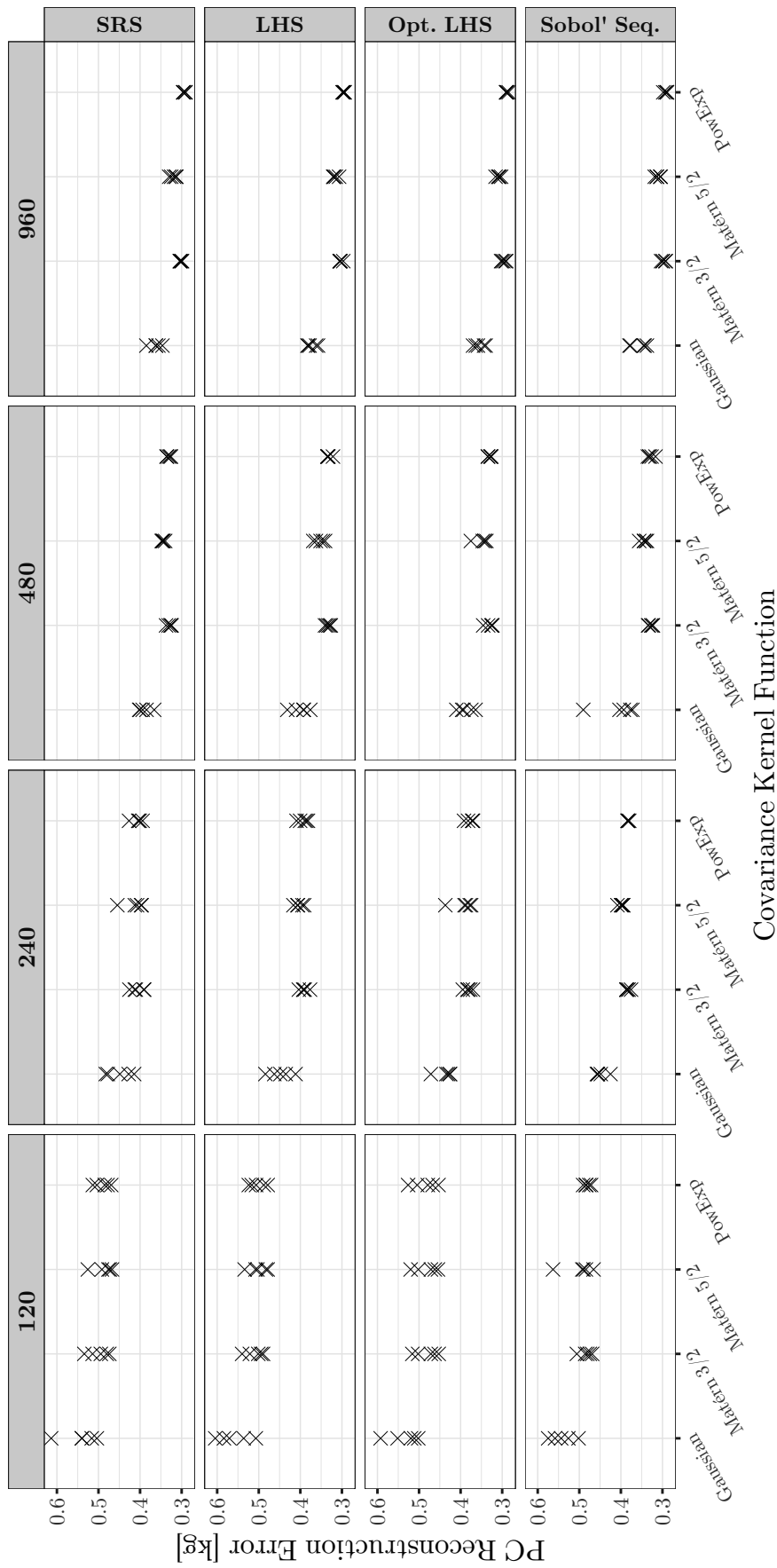


Figure B.19: The effect of training sample size, experimental design, and covariance function on the predictive performance (in terms of RMSE) of GP PC metamodel with respect to the pressure drop output DP. 10 PCs were used for the reconstruction.



Covariance Kernel Function

Figure B.20: The effect of training sample size, experimental design, and covariance function on the predictive performance (in terms of RMSE) of GP PC metamodel with respect to the liquid carryover output CO. 5 PCs were used for the reconstruction.

B.6 MCMC SAMPLES FROM DIFFERENT CALIBRATION SCHEMES

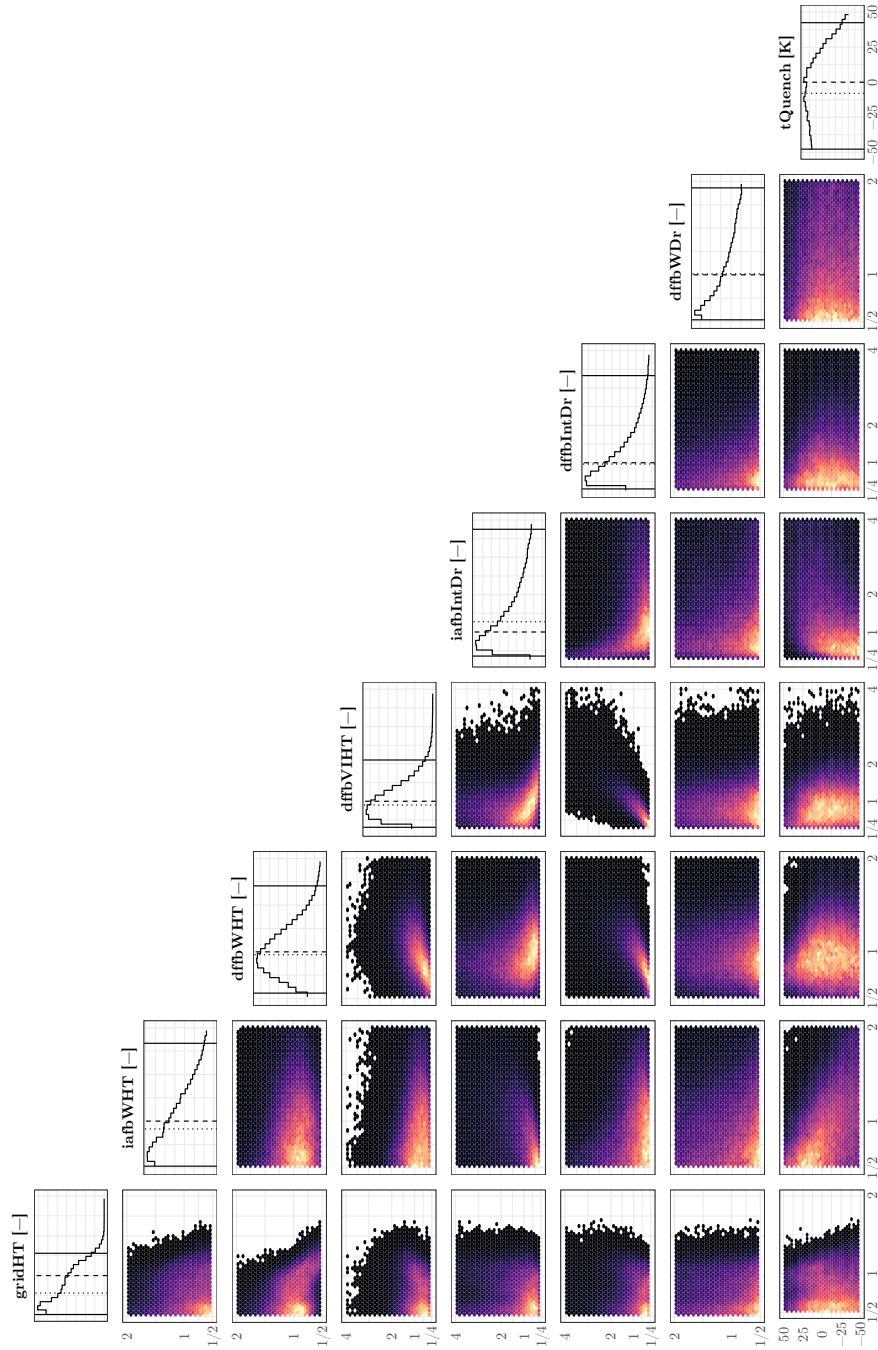


Figure B.21: Univariate and bivariate marginals of the posterior samples for each of the 8 model parameters. Solid lines, dashed, and dotted lines indicate the 95% highest posterior density intervals, the nominal parameter values, and the posterior median parameter values. Calibration with respect to the clad temperature output (TC) and with model bias term.

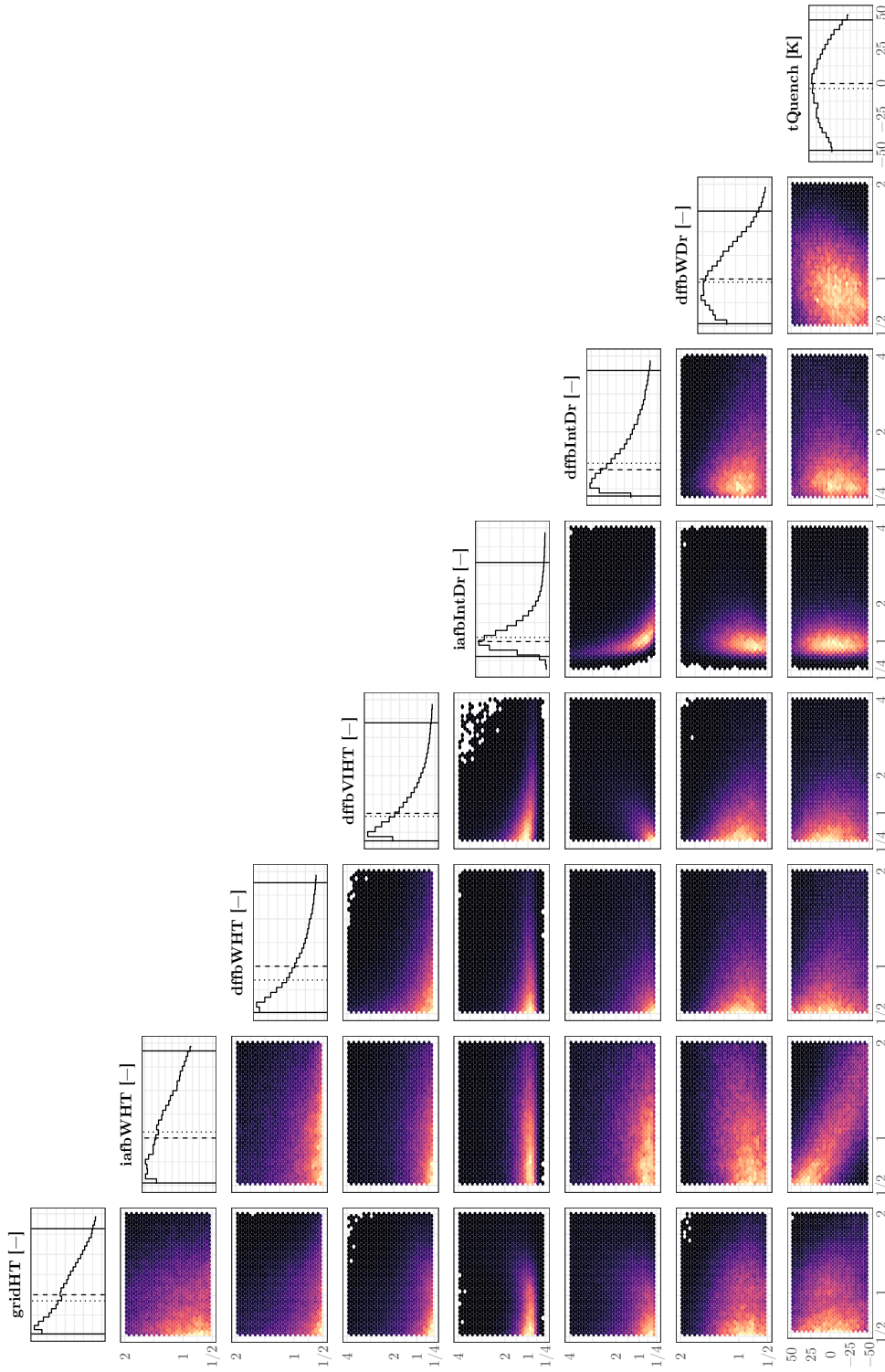


Figure B.22: Univariate and bivariate marginals of the posterior samples for each of the 8 model parameters. Solid lines, dashed, and dotted lines indicate the 95% HPDIs, the nominal parameter values, and the posterior median parameter values. Calibration with respect to the pressure drop output (DP) and with model bias term.

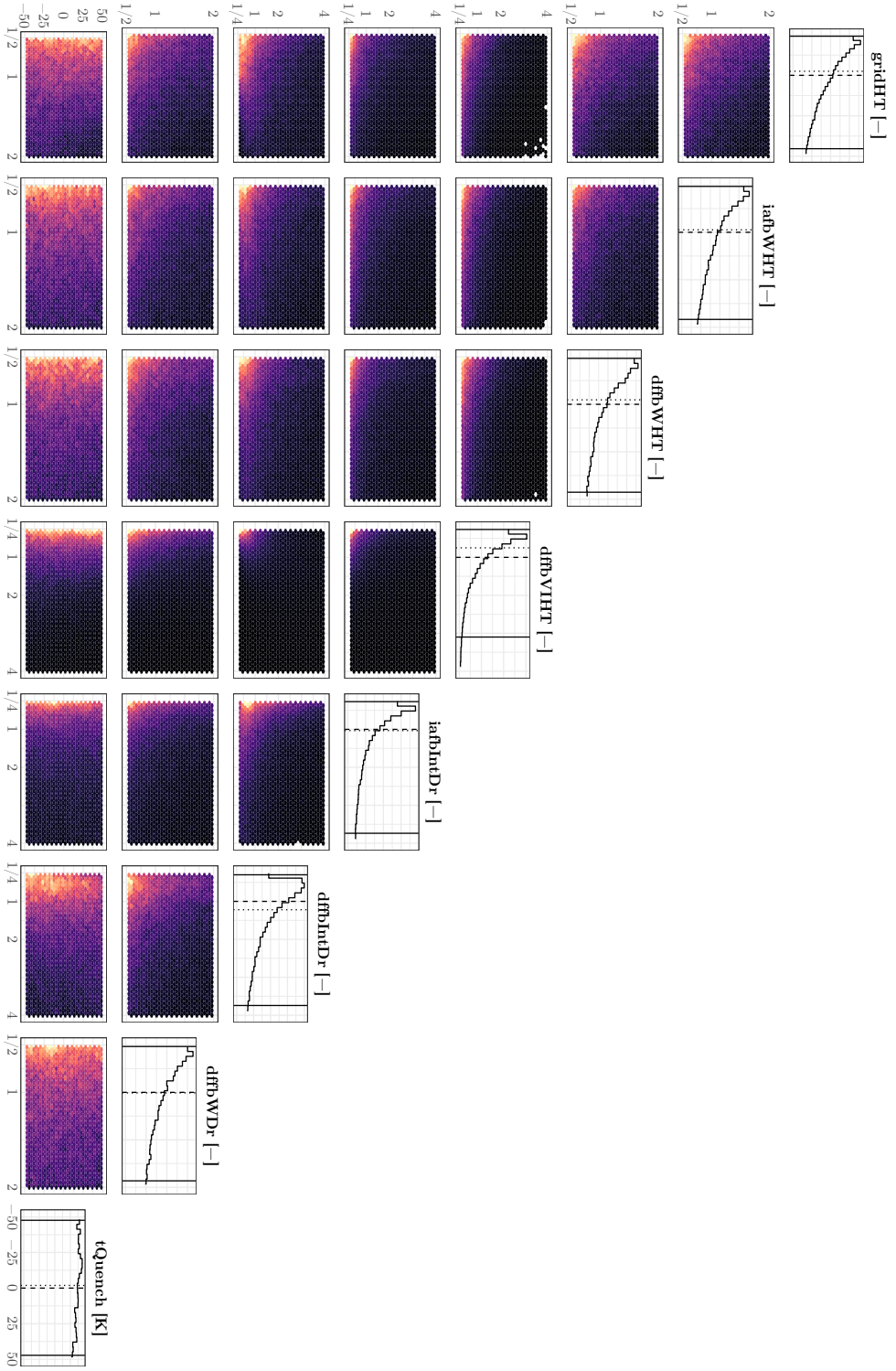


Figure B.23: Univariate and bivariate marginals of the posterior samples for each of the 8 model parameters. Solid lines, dashed, and dotted lines indicate the 95% HPDs, the nominal parameter values, and the posterior median parameter values. Calibration with respect to the liquid carryover output (CO) and with model bias term.

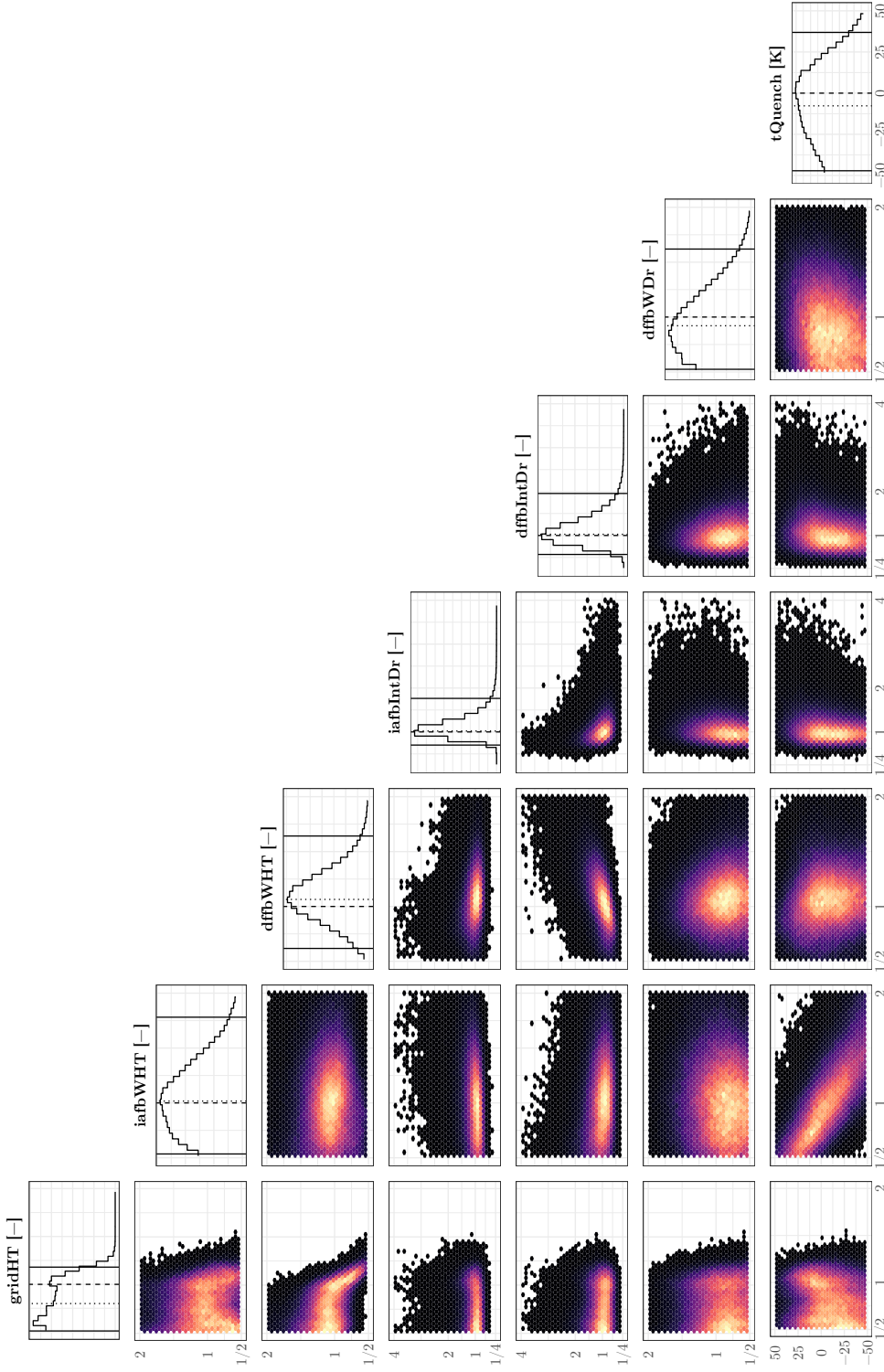


Figure B.24: Univariate and bivariate marginals of the posterior samples for each of the 8 model parameters, excluding $dffbVIHTC$ parameter. Solid lines, dashed, and dotted lines indicate the 95% HPDIs, the nominal parameter values, and the posterior median parameter values. Calibration with respect to all types of output (TC, DP, and CO) and without model bias term.

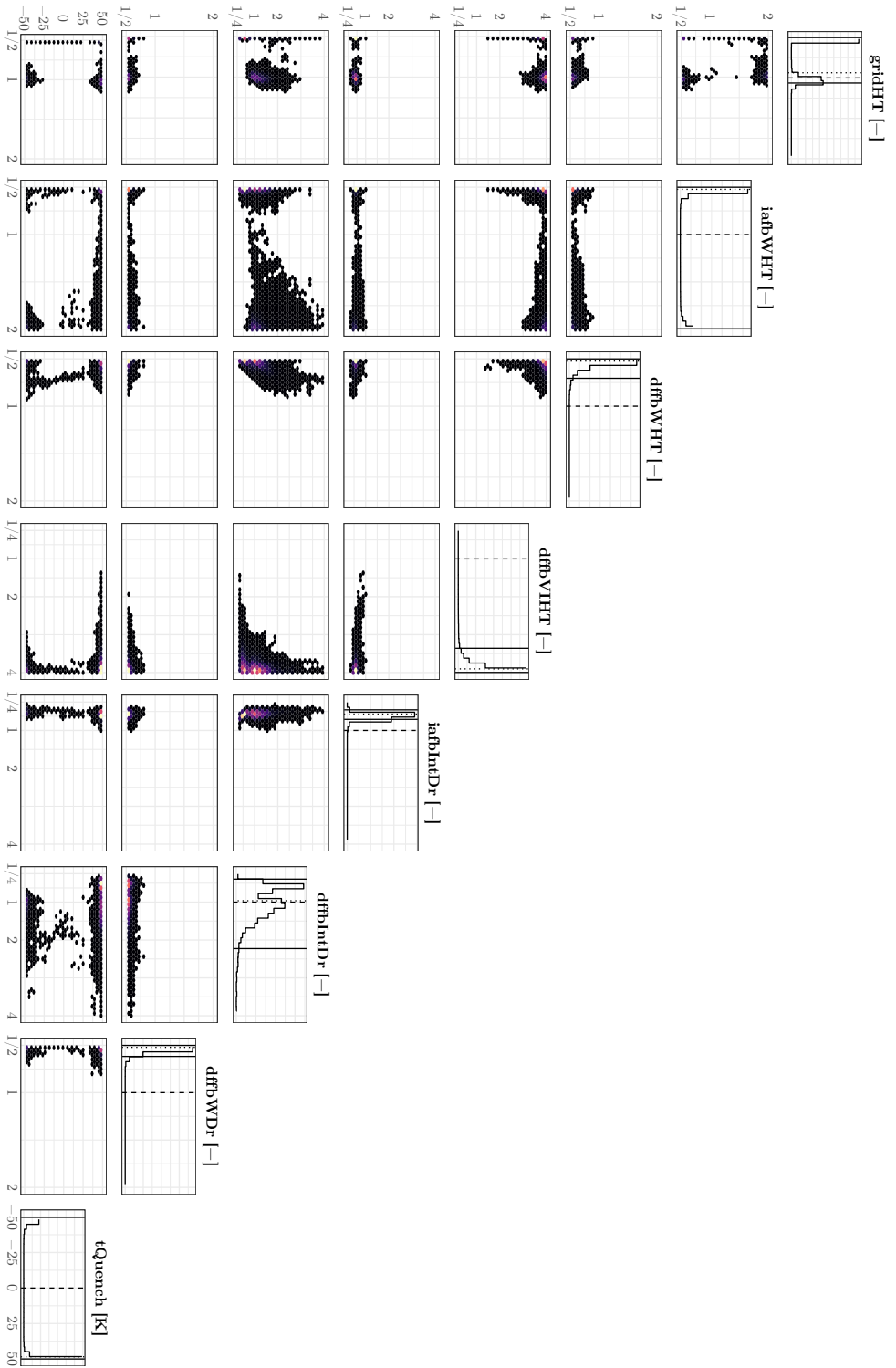


Figure B.25: Univariate and bivariate marginals of the posterior samples for each of the 8 model parameters. Solid lines, dashed, and dotted lines indicate the 95% credible intervals, the nominal parameter values, and the posterior median parameter values. Calibration with respect to all types of output (TC, DP, and CO) and without model bias term.

B.7 FORWARD UNCERTAINTY PROPAGATION OF MCMC SAMPLES

B.7.1 FEBA Test No. 216, clad Temperature Output (TC)

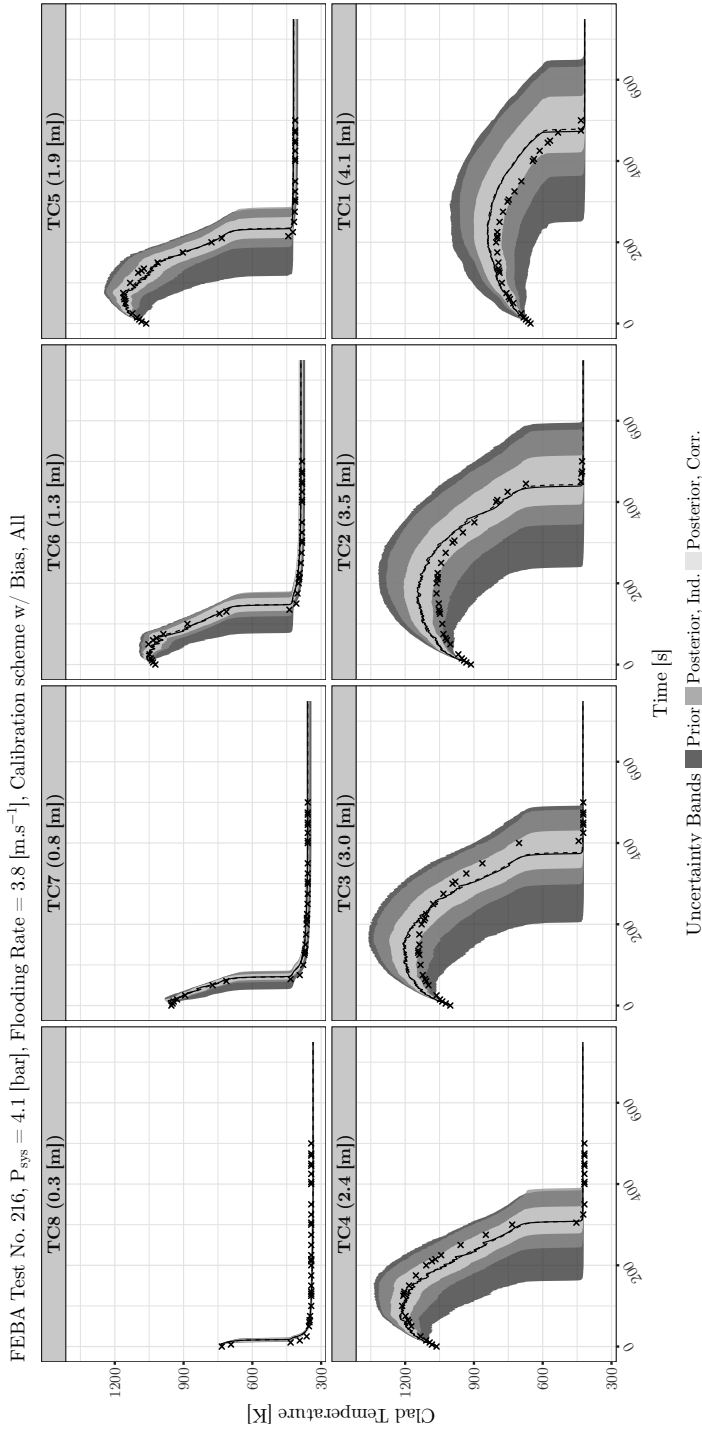


Figure B.26: Propagation of the model parameters uncertainty on FEBA test No. 216 for the clad temperature output (TC) at different axial locations. The uncertainty bands refer to the symmetric 95% probabilities. Solid lines, dashed lines, and crosses indicate the simulation with the nominal parameters values, the median of the posterior, and the experimental data, respectively. The posterior samples are from the calibration with model bias term and considering all types of output (w/ Bias, All).

FEBA Test No. 216, $P_{sys} = 4.1$ [bar], Flooding Rate = 3.8 [m.s⁻¹], Calibration scheme w/ Bias, no dfbVHTT

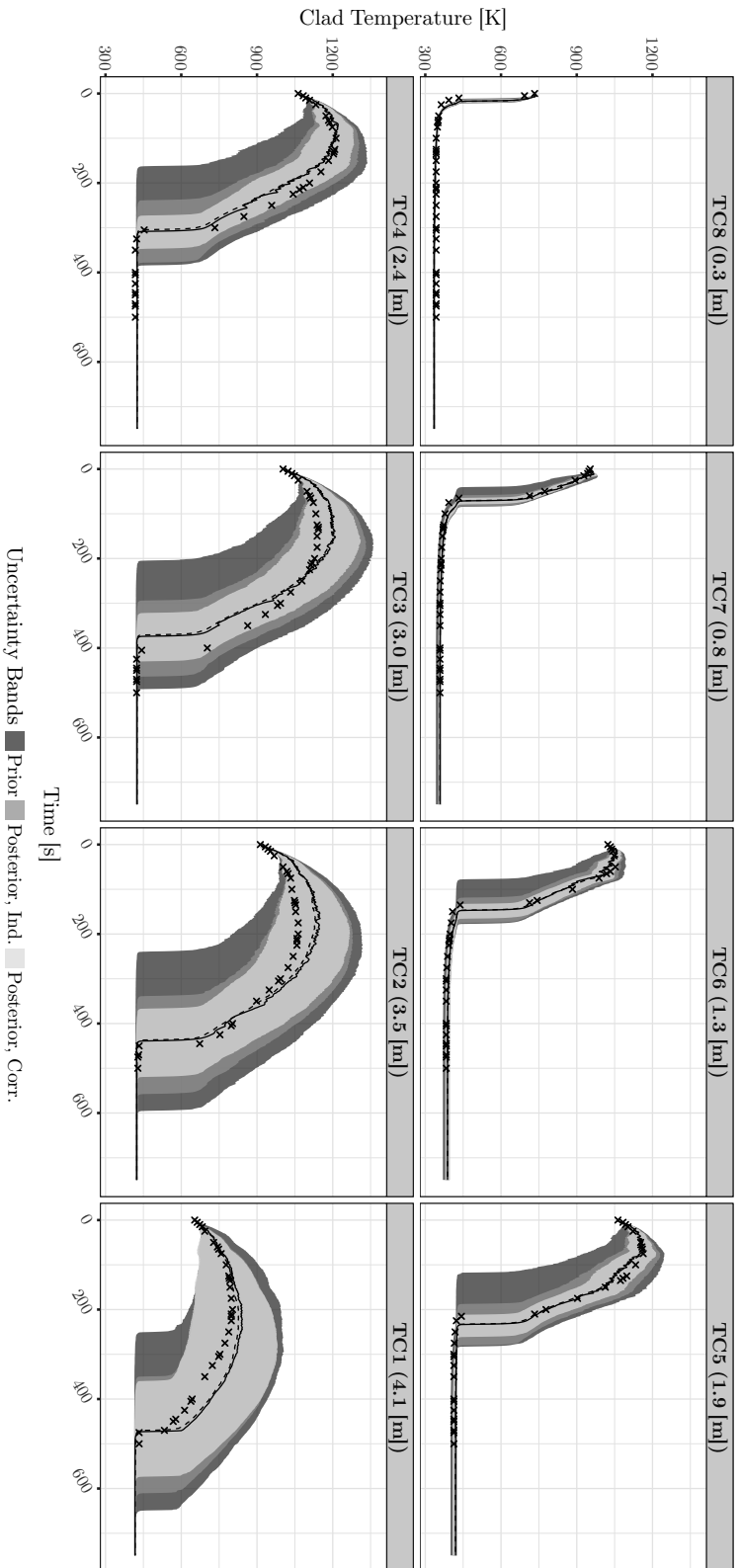


Figure B.27: Propagation of the model parameters uncertainty on FEBA test No. 216 for the clad temperature output (TC) at different axial locations. The uncertainty bands refer to the symmetric uncertainty 95% probabilities. Solid lines, dashed lines, and crosses indicate the simulation with the nominal parameters values, the median of the posterior, and the experimental data, respectively. The posterior samples are from the calibration with model bias term, considering all types of output, but excluding the parameter dfbVHTT (w/ Bias, no dfbVHTT).

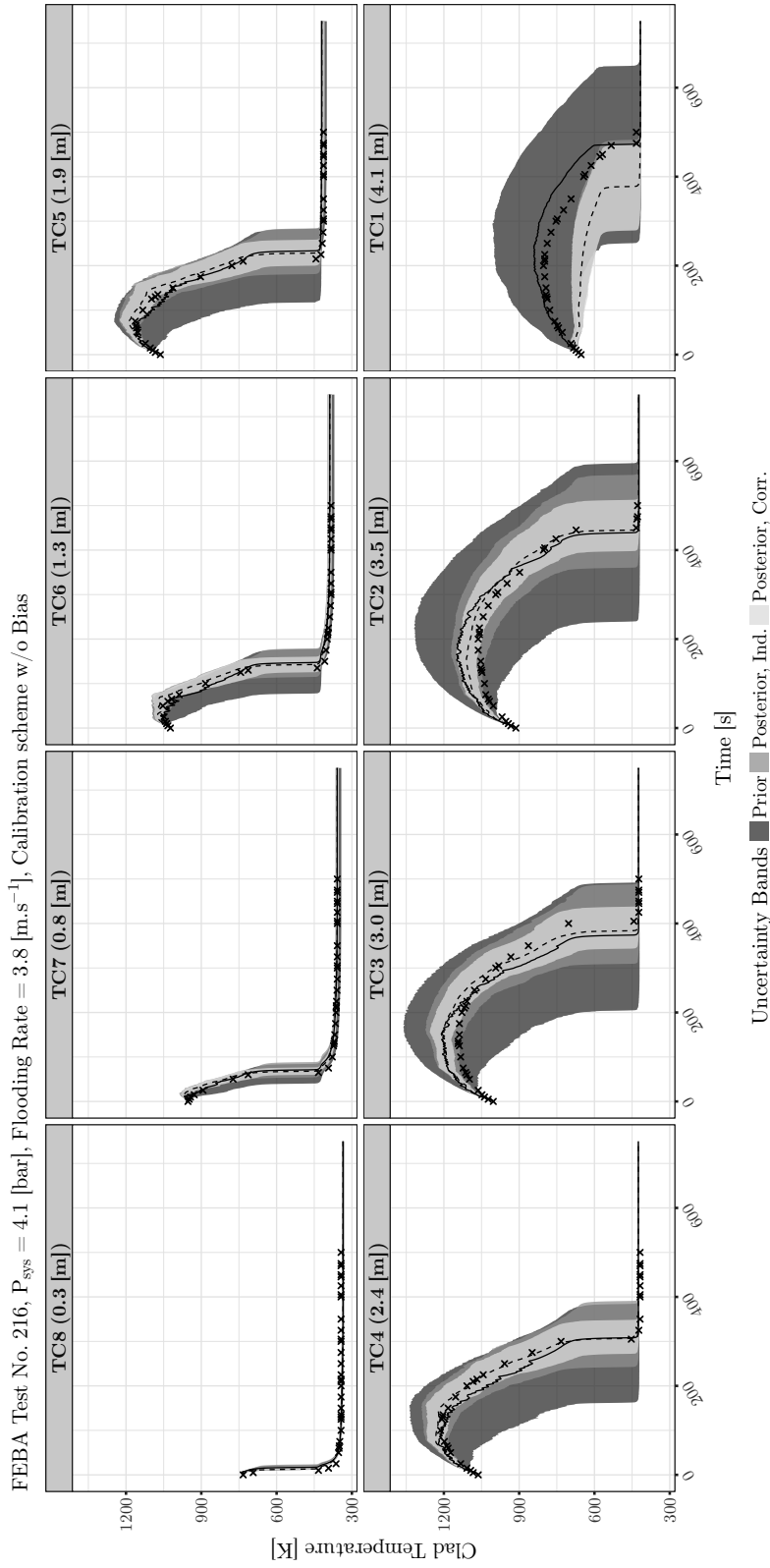


Figure B.28: Propagation of the model parameters uncertainty on FEBA test No. 216 for the clad temperature output (TC) at different axial locations. The uncertainty bands refer to the symmetric 95% probabilities. Solid lines, dashed lines, and crosses indicate the simulation with the nominal parameters values, the median of the posterior, and the experimental data, respectively. The posterior samples are from the calibration without model bias term and considering all types of output (w/o Bias).

B.7.2 FEBA Test No. 214, clad Temperature Output (TC)

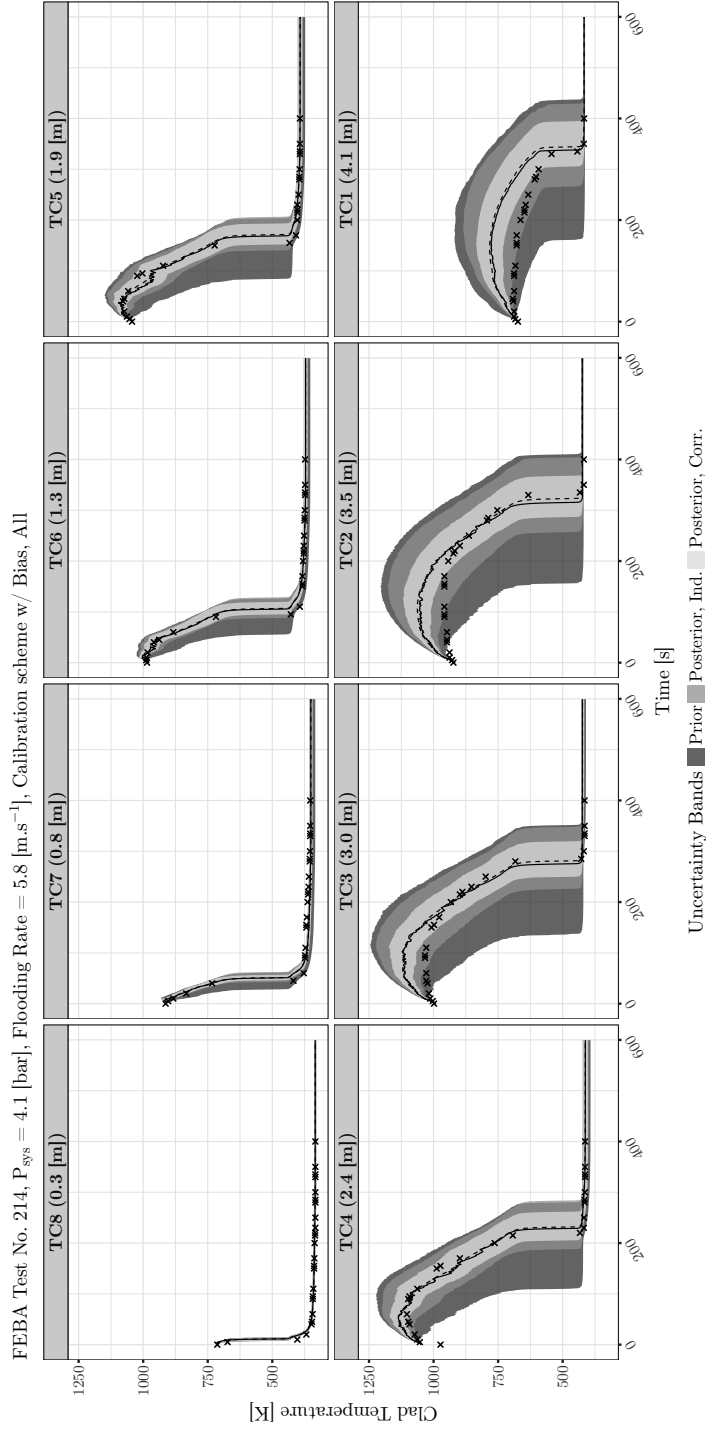


Figure B.29: Propagation of the model parameters uncertainty on FEBA test No. 214 for the clad temperature output (TC) at different axial locations. The uncertainty bands refer to the symmetric 95% probabilities. Solid lines, dashed lines, and crosses indicate the simulation with the nominal parameters values, the median of the posterior, and the experimental data, respectively. The posterior samples are from the calibration with model bias term and considering all types of output (w/ Bias, All).

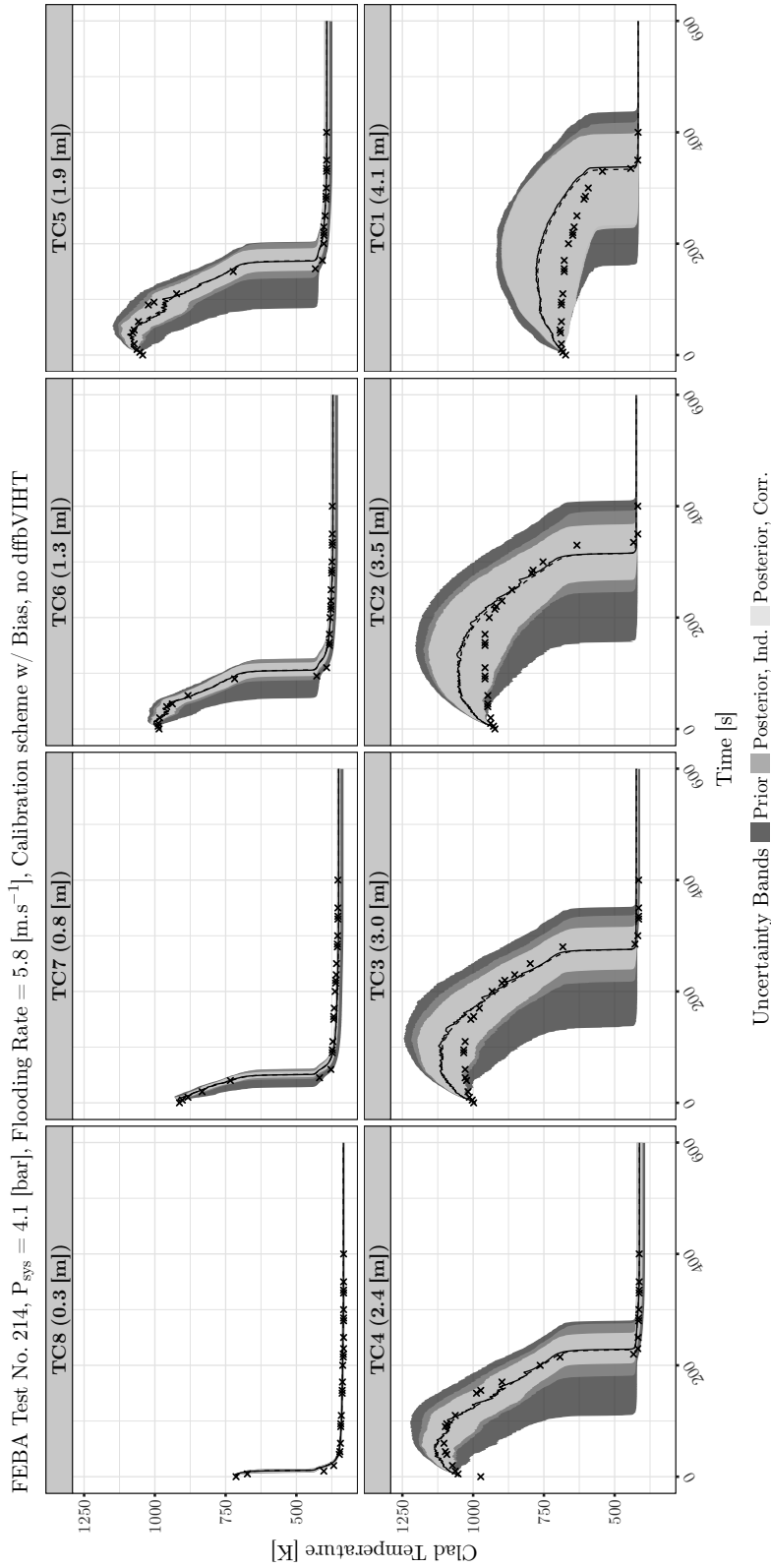


Figure B.30: Propagation of the model parameters uncertainty on FEBA test No. 214 for the clad temperature output (TC) at different axial locations. The uncertainty bands refer to the symmetric 95% probabilities. Solid lines, dashed lines, and crosses indicate the simulation with the nominal parameters values, the median of the posterior, and the experimental data, respectively. The posterior samples are from the calibration with model bias term, considering all types of output, but excluding the parameter dffbVIHT (w/ Bias, no dffbVIHT).

FEBA Test No. 214, $P_{sys} = 4.1$ [bar], Flooding Rate = 5.8 [$\text{m}\cdot\text{s}^{-1}$], Calibration scheme w/o Bias

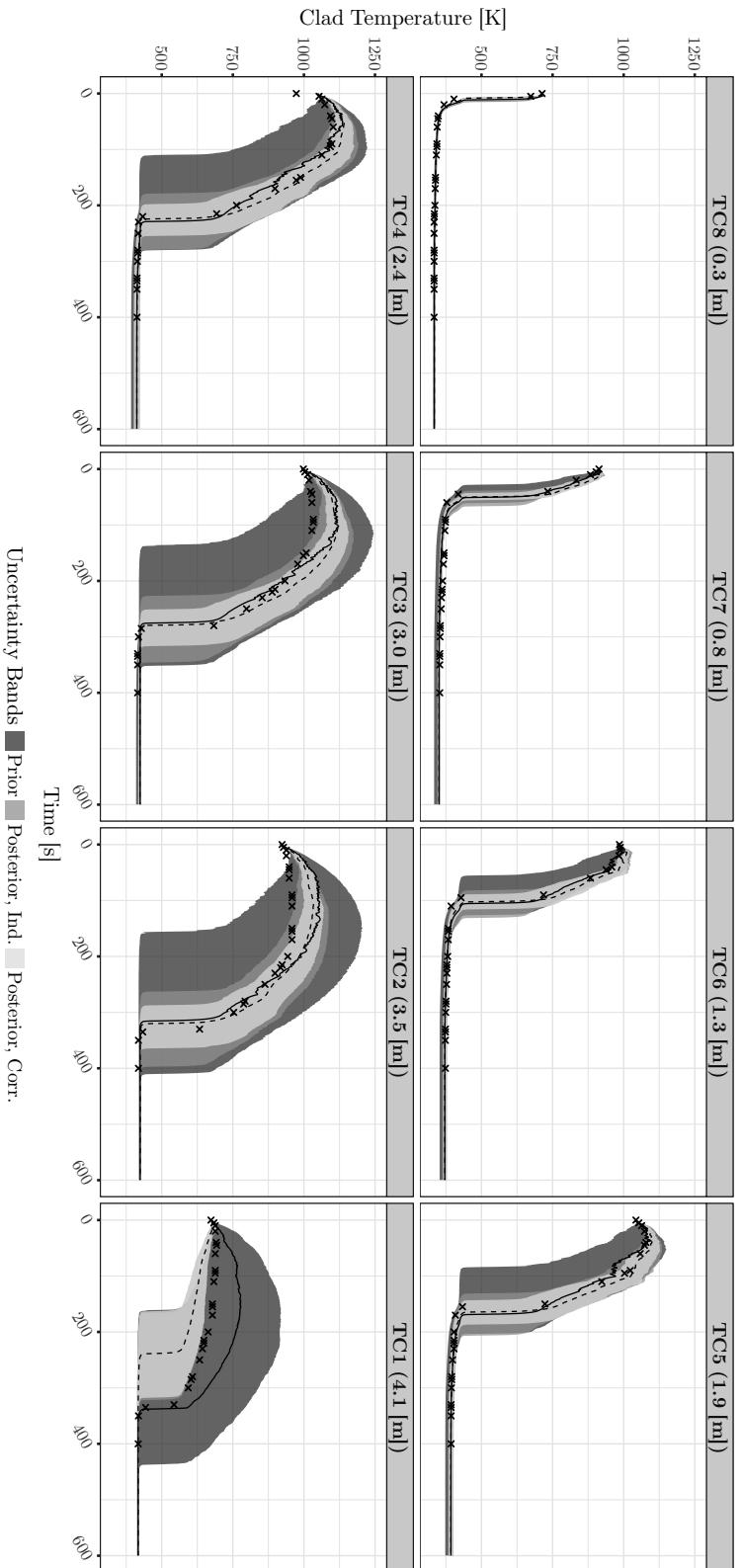


Figure B.31: Propagation of the model parameters uncertainty on FEBA test No. 214 for the clad temperature output (TC) at different axial locations. The uncertainty bands refer to the symmetric 95% probabilities. Solid lines, dashed lines, and crosses indicate the simulation with the nominal parameters values, the median of the posterior, and the experimental data, respectively. The posterior samples are from the calibration without model bias term and considering all types of output (w/o Bias).

B.7.3 FEBA Test No. 223, clad Temperature Output (TC)

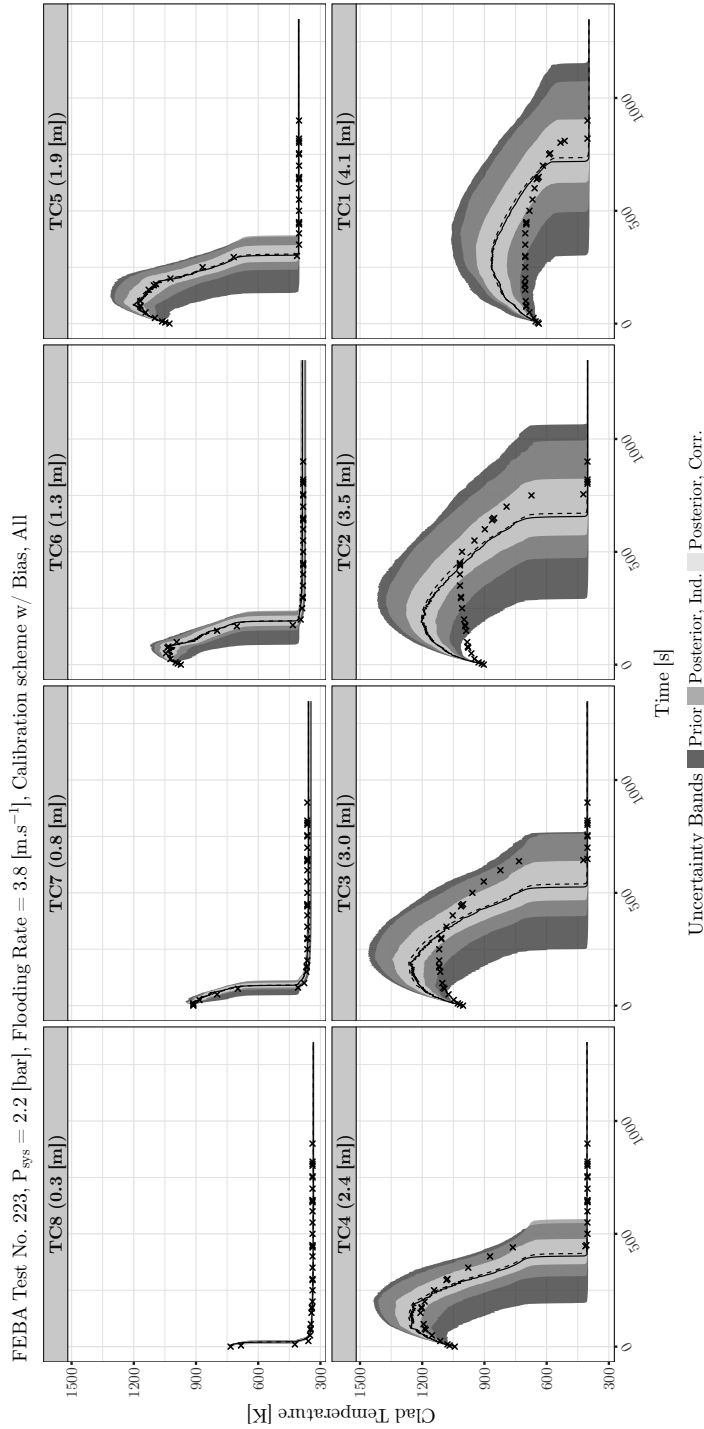


Figure B.32: Propagation of the model parameters uncertainty on FEBA test No. 223 for the clad temperature output (TC) at different axial locations. The uncertainty bands refer to the symmetric 95% probabilities. Solid lines, dashed lines, and crosses indicate the simulation with the nominal parameters values, the median of the posterior, and the experimental data, respectively. The posterior samples are from the calibration with model bias term and considering all types of output (w/ Bias, All).

FEBA Test No. 223, $P_{sys} = 2.2$ [bar], Flooding Rate = 3.8 [m.s⁻¹], Calibration scheme w/ Bias, no dfbVHTT

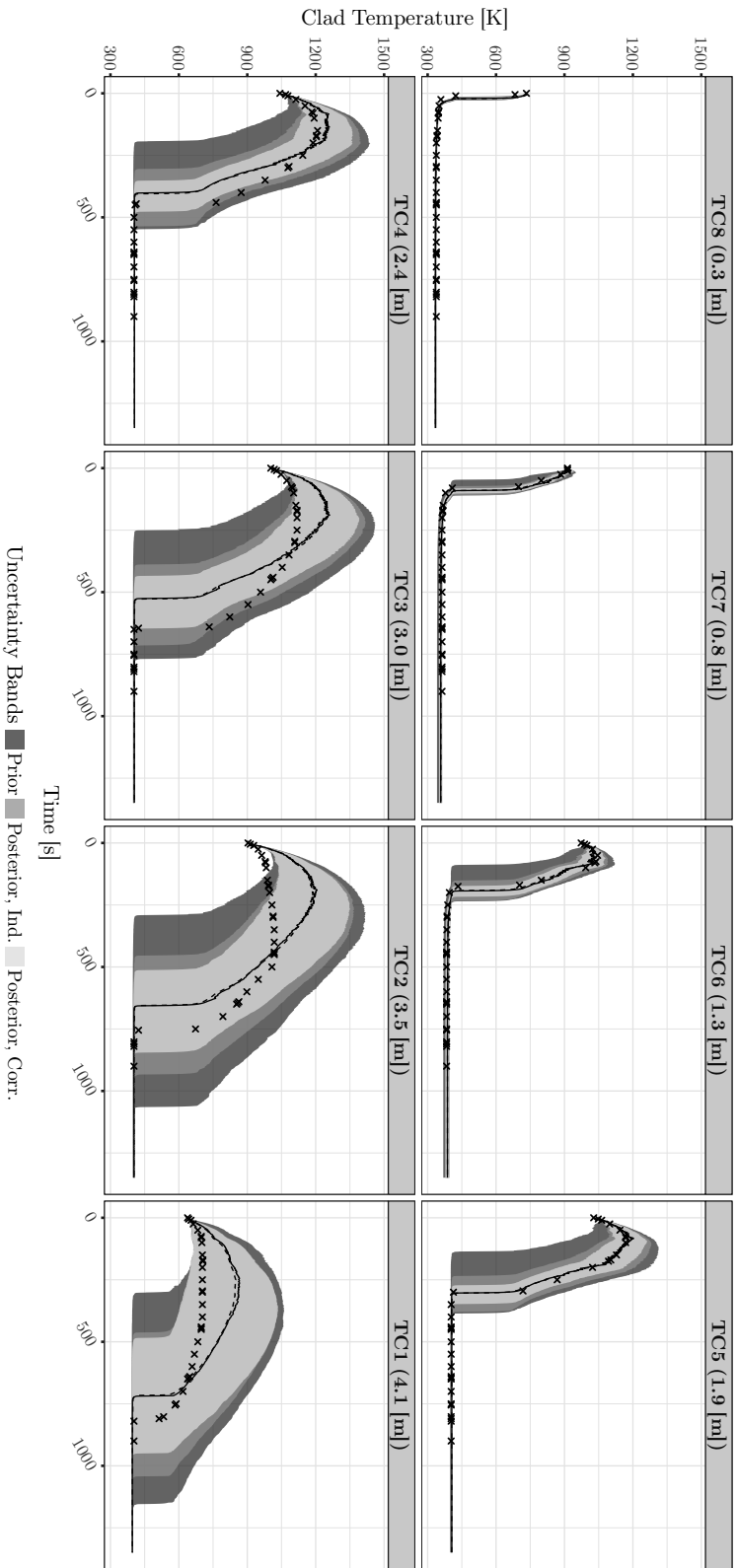


Figure B.33: Propagation of the model parameters uncertainty on FEBA test No. 223 for the clad temperature output (TC) at different axial locations. The uncertainty bands refer to the symmetric uncertainty 95% probabilities. Solid lines, dashed lines, and crosses indicate the simulation with the nominal parameters values, the median of the posterior, and the experimental data, respectively. The posterior samples are from the calibration with model bias term, considering all types of output, but excluding the parameter dfbVHTT (w/ Bias, no dfbVHTT).

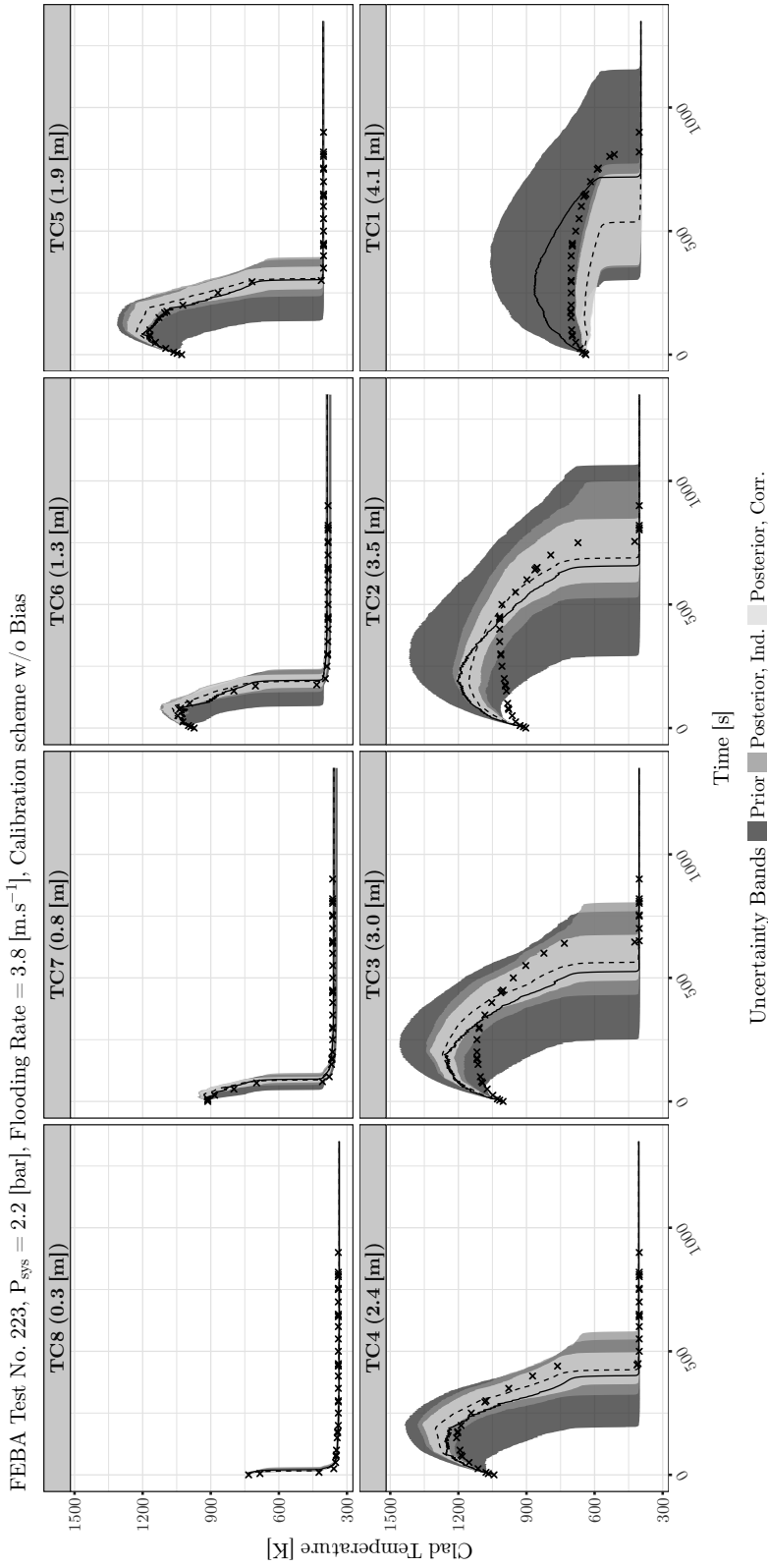


Figure B.34: Propagation of the model parameters uncertainty on FEBA test No. 223 for the clad temperature output (TC) at different axial locations. The uncertainty bands refer to the symmetric 95% probabilities. Solid lines, dashed lines, and crosses indicate the simulation with the nominal parameters values, the median of the posterior, and the experimental data, respectively. The posterior samples are from the calibration without model bias term and considering all types of output (w/o Bias).

B.7.4 FEBA Test No. 218, clad Temperature Output (TC)

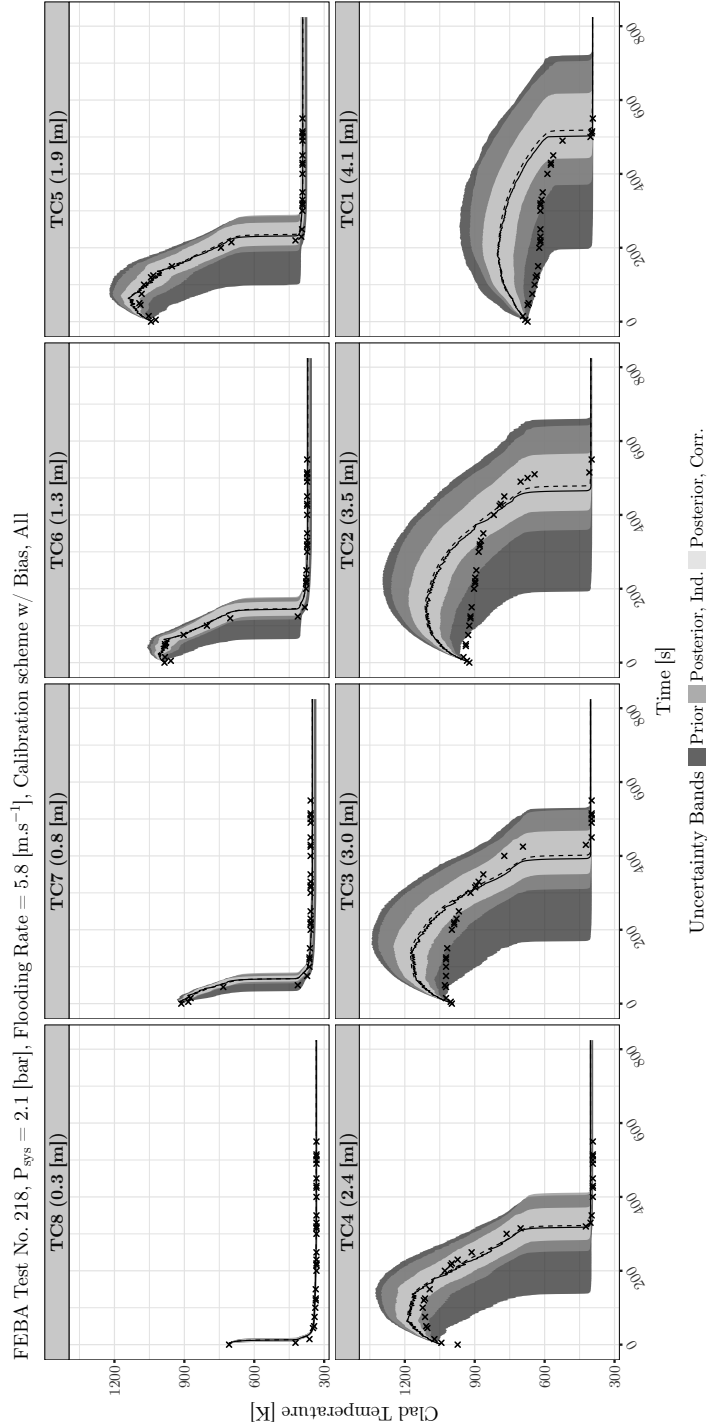


Figure B.35: Propagation of the model parameters uncertainty on FEBA test No. 218 for the clad temperature output (TC) at different axial locations. The uncertainty bands refer to the symmetric 95% probabilities. Solid lines, dashed lines, and crosses indicate the simulation with the nominal parameters values, the median of the posterior, and the experimental data, respectively. The posterior samples are from the calibration with model bias term and considering all types of output (w/ Bias, All).

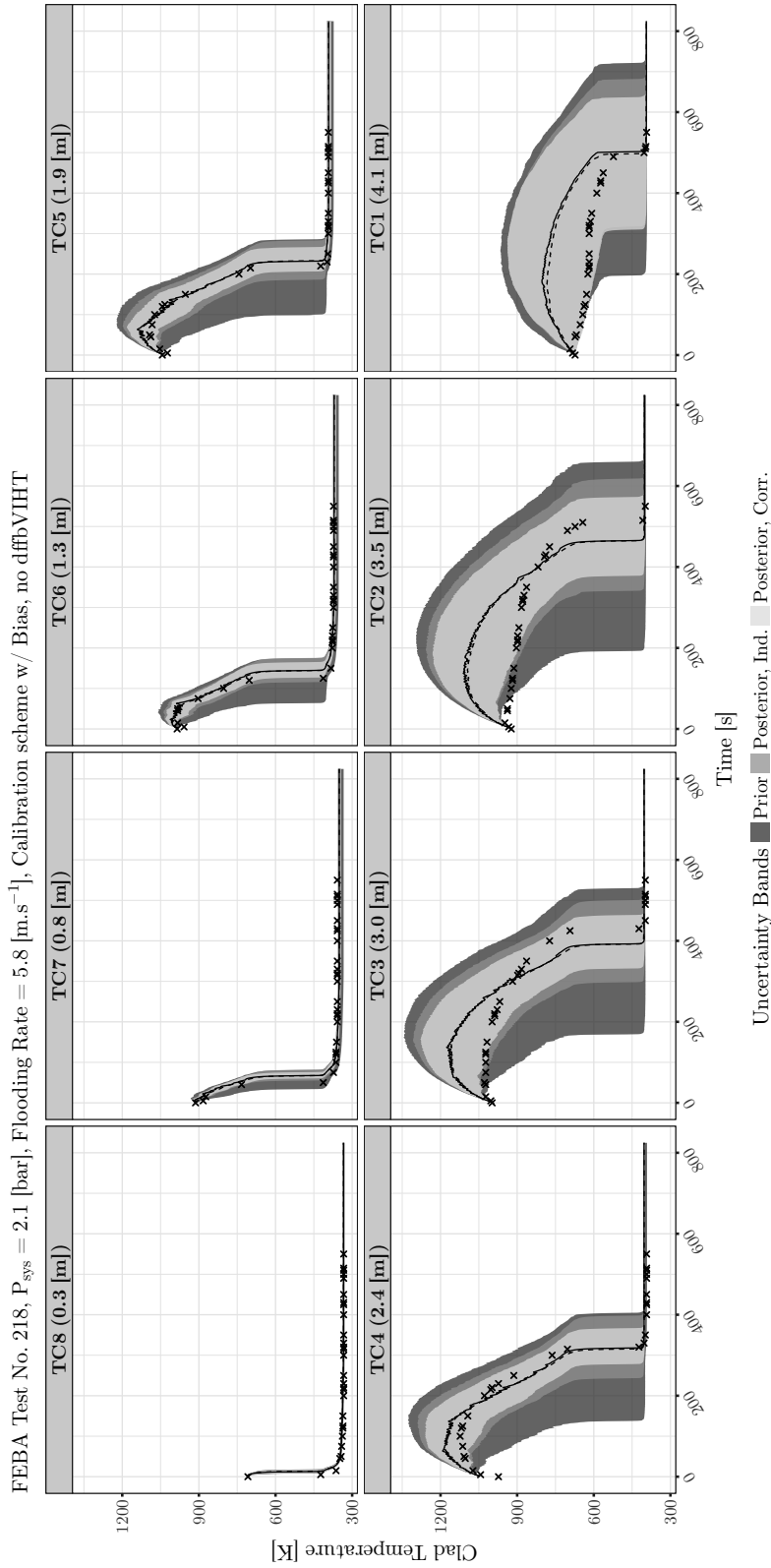


Figure B.36: Propagation of the model parameters uncertainty on FEBA test No. 218 for the clad temperature output (TC) at different axial locations. The uncertainty bands refer to the symmetric 95% probabilities. Solid lines, dashed lines, and crosses indicate the simulation with the nominal parameters values, the median of the posterior, and the experimental data, respectively. The posterior samples are from the calibration with model bias term, considering all types of output, but excluding the parameter dffbVIHT (w/ Bias, no dffbVIHT).

FEBA Test No. 218, $P_{sys} = 2.1$ [bar], Flooding Rate = 5.8 [m.s⁻¹], Calibration scheme w/o Bias

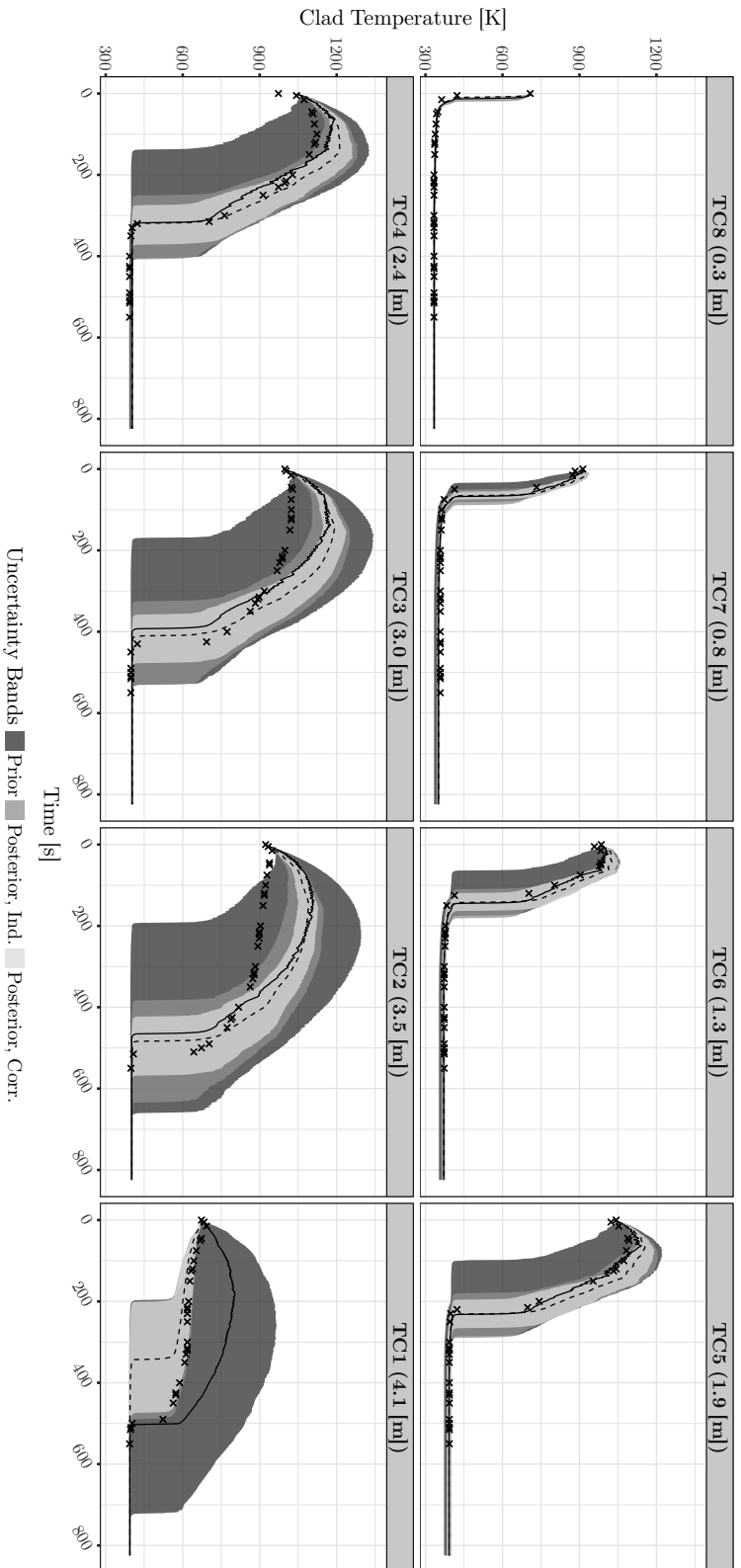


Figure B.37: Propagation of the model parameters uncertainty on FEBA test No. 218 for the clad temperature output (TC) at different axial locations. The uncertainty bands refer to the symmetric uncertainty 95% probabilities. Solid lines, dashed lines, and crosses indicate the simulation with the nominal parameters values, the median of the posterior, and the experimental data, respectively. The posterior samples are from the calibration without model bias term and considering all types of output (w/o Bias).

B.7.5 FEBA Test No. 220, clad Temperature Output (TC)

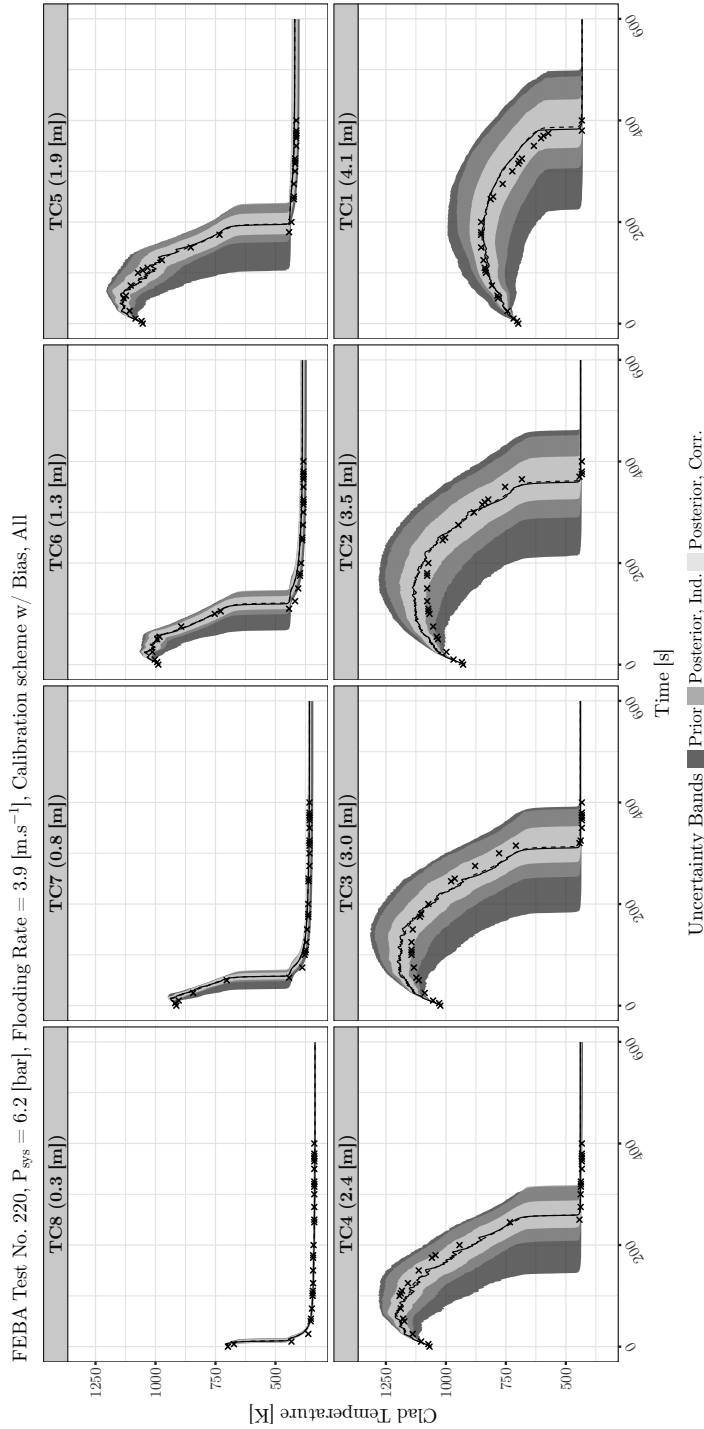


Figure B.38: Propagation of the model parameters uncertainty on FEBA test No. 220 for the clad temperature output (TC) at different axial locations. The uncertainty bands refer to the symmetric 95% probabilities. Solid lines, dashed lines, and crosses indicate the simulation with the nominal parameters values, the median of the posterior, and the experimental data, respectively. The posterior samples are from the calibration with model bias term and considering all types of output (w/ Bias, All).

FEBA Test No. 220, $P_{sys} = 6.2$ [bar], Flooding Rate = 3.9 [$\text{m}\cdot\text{s}^{-1}$], Calibration scheme w/ Bias, no dfbVHTT

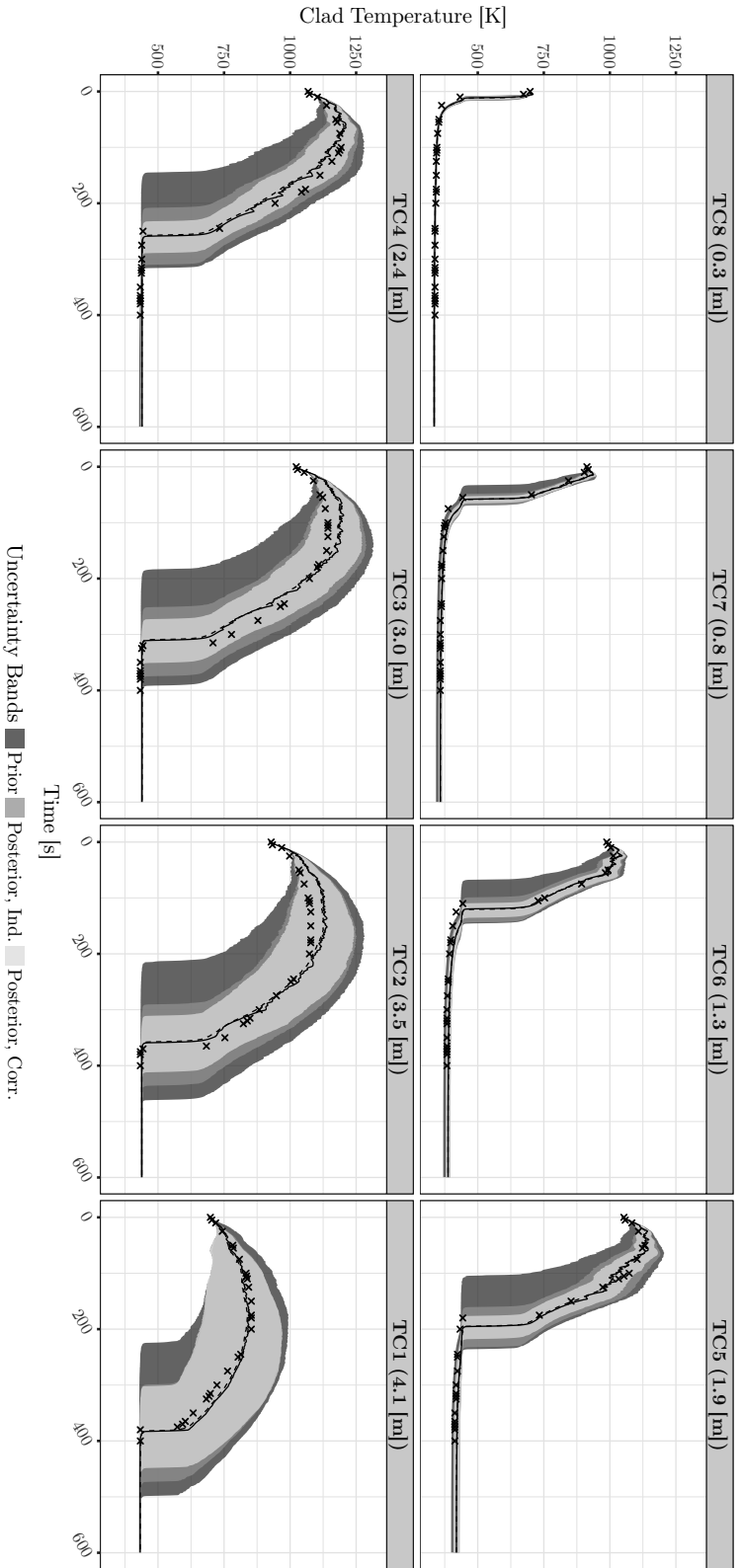


Figure B.39: Propagation of the model parameters uncertainty on FEBA test No. 220 for the clad temperature output (TC) at different axial locations. The uncertainty bands refer to the symmetric uncertainty 95% probabilities. Solid lines, dashed lines, and crosses indicate the simulation with the nominal parameters values, the median of the posterior, and the experimental data, respectively. The posterior samples are from the calibration with model bias term, considering all types of output, but excluding the parameter dfbVHTT (w/ Bias, no dfbVHTT).

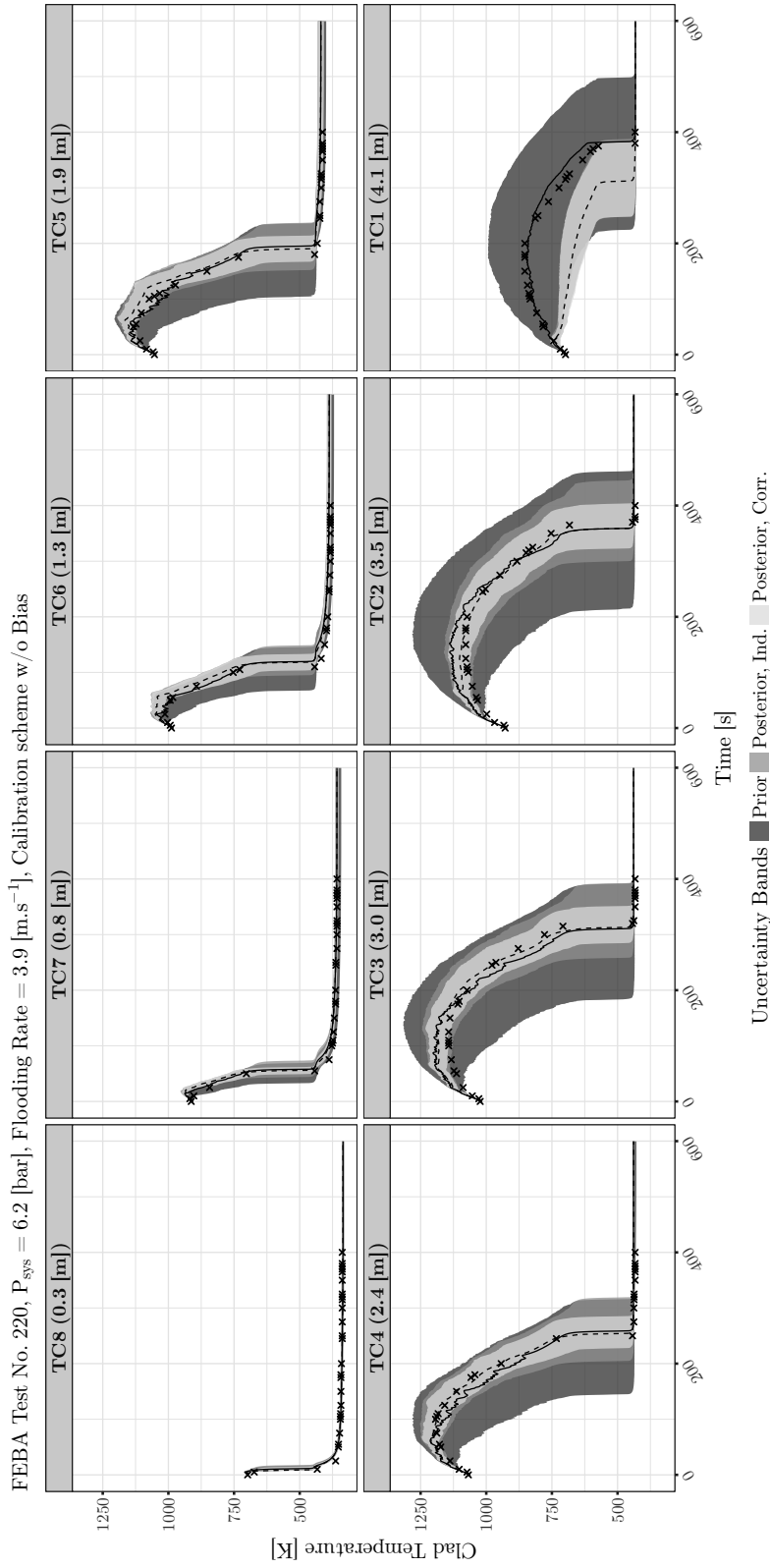


Figure B.4o: Propagation of the model parameters uncertainty on FEBA test No. 220 for the clad temperature output (TC) at different axial locations. The uncertainty bands refer to the symmetric 95% probabilities. Solid lines, dashed lines, and crosses indicate the simulation with the nominal parameters values, the median of the posterior, and the experimental data, respectively. The posterior samples are from the calibration without model bias term and considering all types of output (w/o Bias).

B.7.6 FEBA Test No. 222, clad Temperature Output (TC)

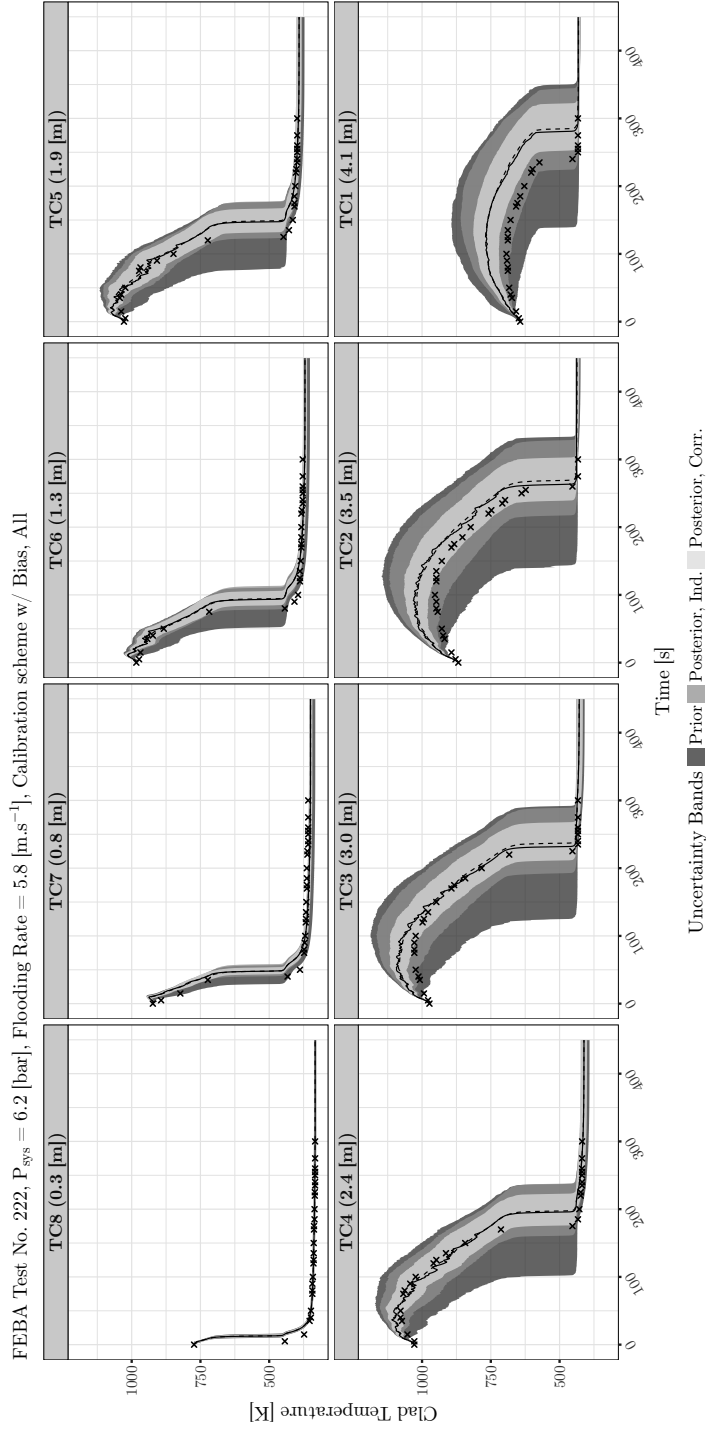


Figure B.41: Propagation of the model parameters uncertainty on FEBA test No. 222 for the clad temperature output (TC) at different axial locations. The uncertainty bands refer to the symmetric 95% probabilities. Solid lines, dashed lines, and crosses indicate the simulation with the nominal parameters values, the median of the posterior, and the experimental data, respectively. The posterior samples are from the calibration with model bias term and considering all types of output (w/ Bias, All).

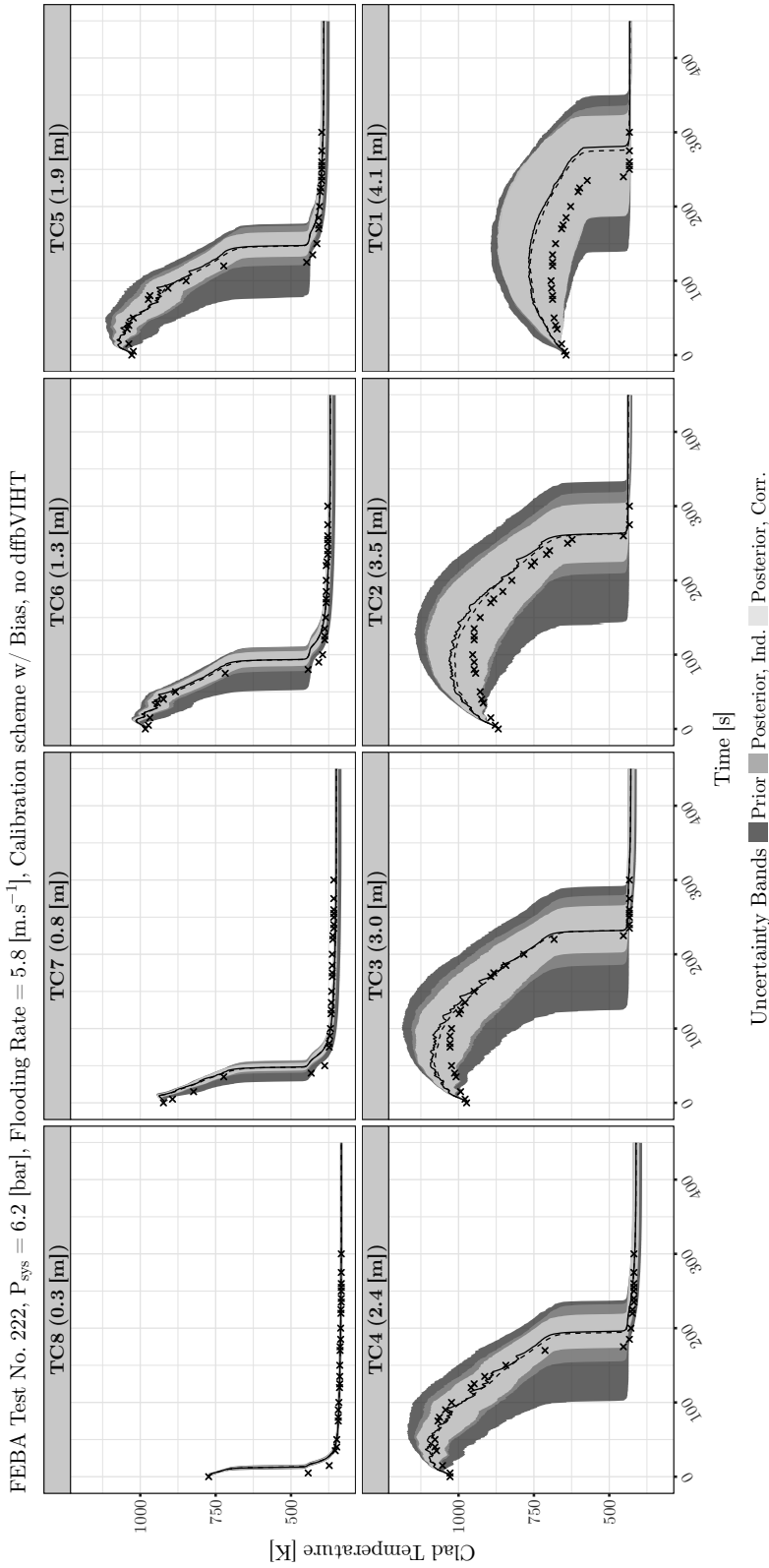


Figure B.42: Propagation of the model parameters uncertainty on FEBA test No. 222 for the clad temperature output (TC) at different axial locations. The uncertainty bands refer to the symmetric 95% probabilities. Solid lines, dashed lines, and crosses indicate the simulation with the nominal parameters values, the median of the posterior, and the experimental data, respectively. The posterior samples are from the calibration with model bias term, considering all types of output, but excluding the parameter dffbVIHT (w/ Bias, no dffbVIHT).

FEBA Test No. 222, $P_{sys} = 6.2$ [bar], Flooding Rate = 5.8 [m.s⁻¹], Calibration scheme w/o Bias

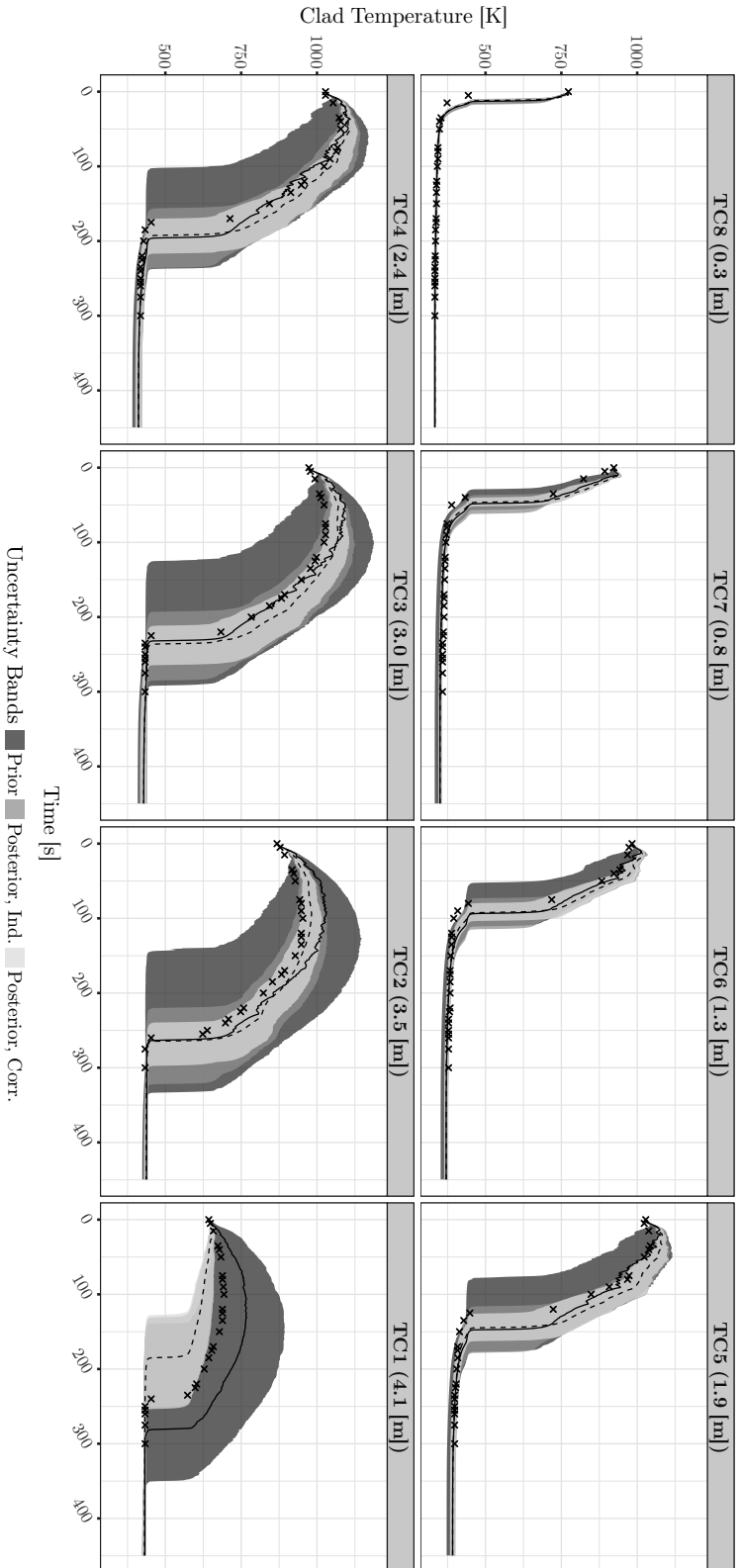


Figure B.43: Propagation of the model parameters uncertainty on FEBA test No. 222 for the clad temperature output (TC) at different axial locations. The uncertainty bands refer to the symmetric uncertainty 95% probabilities. Solid lines, dashed lines, and crosses indicate the simulation with the nominal parameters values, the median of the posterior, and the experimental data, respectively. The posterior samples are from the calibration without model bias term and considering all types of output (w/o Bias).

B.7.7 FEBA Test No. 216, Pressure Drop Output (DP)

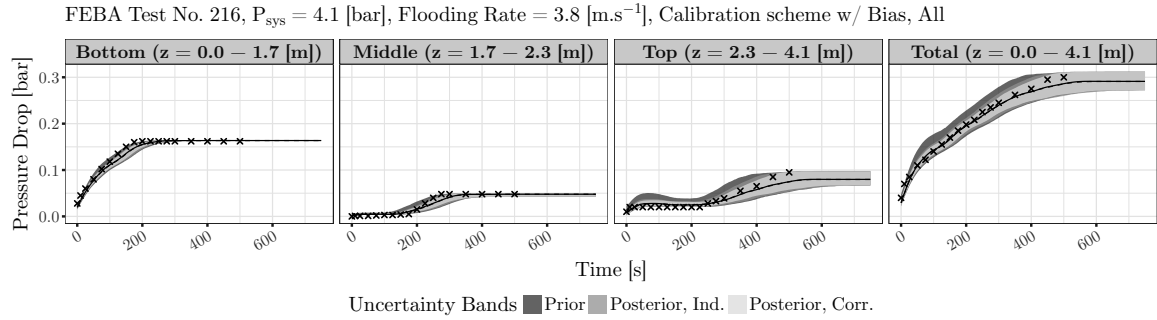


Figure B.44: Propagation of the model parameters uncertainty on FEBA test No. 216 for the pressure drop output (DP) at different axial segments. The uncertainty bands refer to the symmetric 95% probabilities. Solid lines, dashed lines, and crosses indicate the simulation with the nominal parameters values, the median of the posterior, and the experimental data, respectively. The posterior samples are from the calibration with model bias term and considering all types of output (w/ Bias, All).

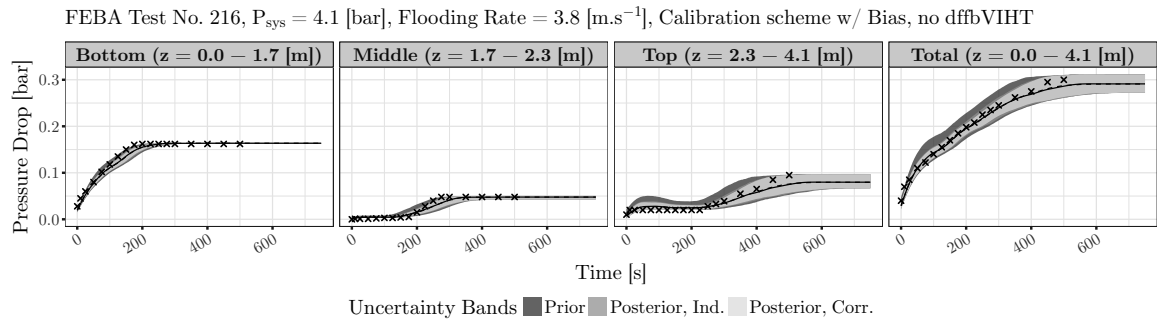


Figure B.45: The posterior samples are from the calibration with model bias term, considering all types of output, but excluding the parameter dffbVIHT (w/ Bias, no dffbVIHT).

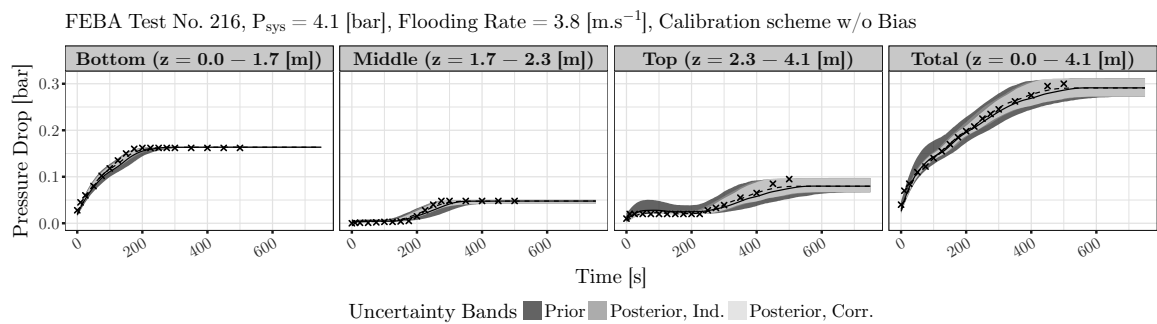


Figure B.46: The posterior samples are from the calibration without model bias term and considering all types of output (w/o Bias).

B.7.8 FEBA Test No. 214, Pressure Drop Output (DP)

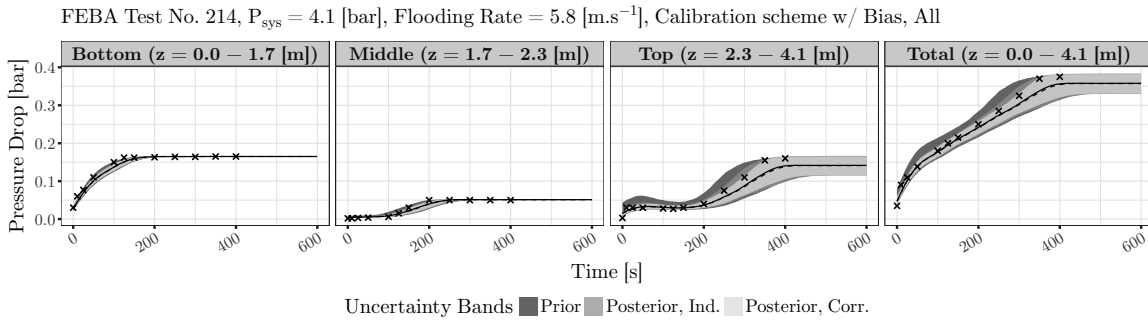


Figure B.47: Propagation of the model parameters uncertainty on FEBA test No. 214 for the pressure drop output (DP) at different axial segments. The uncertainty bands refer to the symmetric 95% probabilities. Solid lines, dashed lines, and crosses indicate the simulation with the nominal parameters values, the median of the posterior, and the experimental data, respectively. The posterior samples are from the calibration with model bias term and considering all types of output (w/ Bias, All).

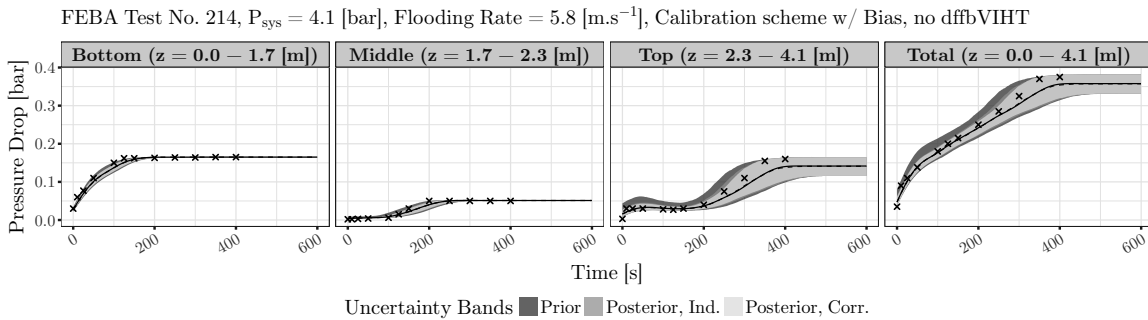


Figure B.48: The posterior samples are from the calibration with model bias term, considering all types of output, but excluding the parameter dffbVIHT (w/ Bias, no dffbVIHT).

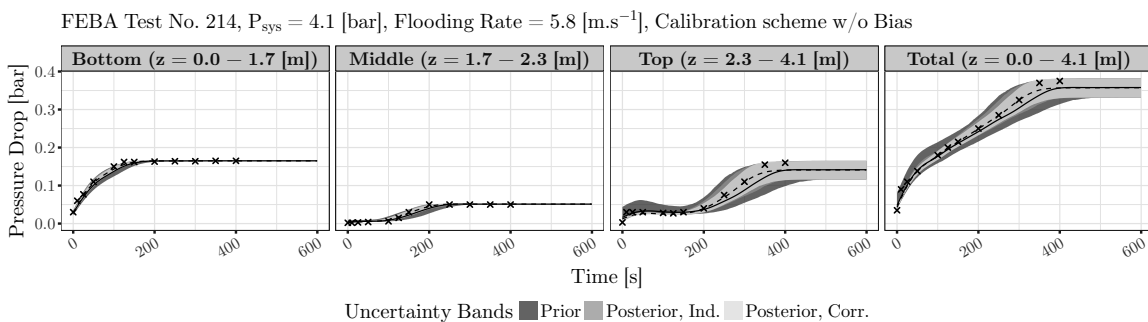


Figure B.49: The posterior samples are from the calibration without model bias term and considering all types of output (w/o Bias).

B.7.9 FEBA Test No. 223, Pressure Drop Output (DP)

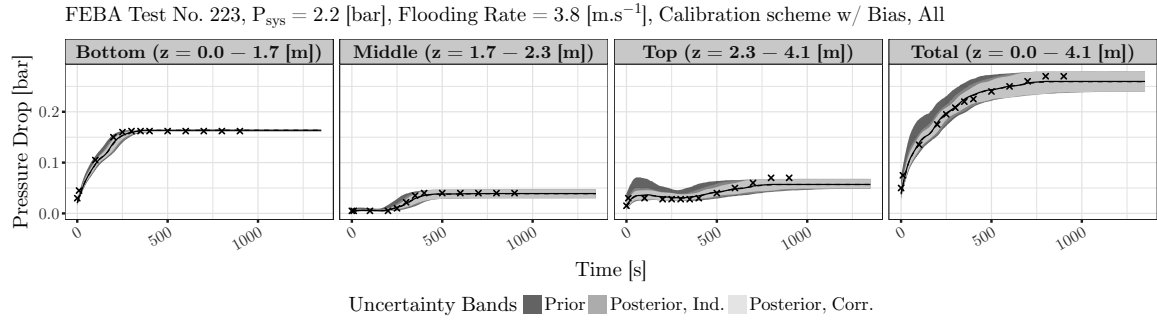


Figure B.50: Propagation of the model parameters uncertainty on FEBA test No. 223 for the pressure drop output (DP) at different axial segments. The uncertainty bands refer to the symmetric 95% probabilities. Solid lines, dashed lines, and crosses indicate the simulation with the nominal parameters values, the median of the posterior, and the experimental data, respectively. The posterior samples are from the calibration with model bias term and considering all types of output (w/ Bias, All).

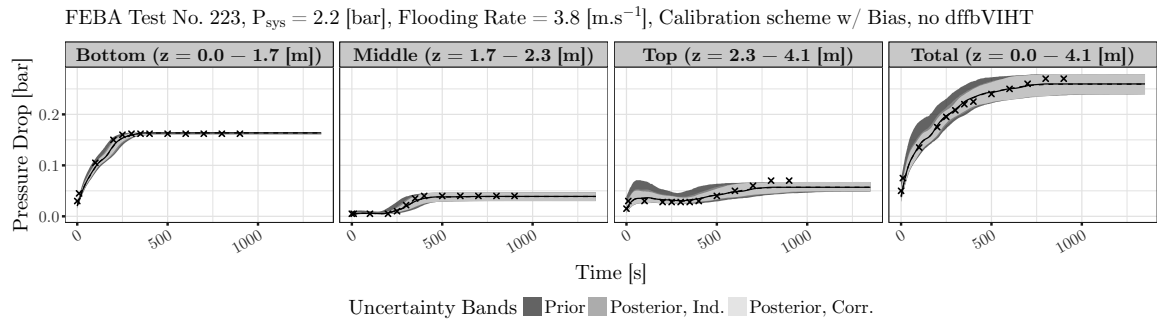


Figure B.51: The posterior samples are from the calibration with model bias term, considering all types of output, but excluding the parameter dffbVIHT (w/ Bias, no dffbVIHT).

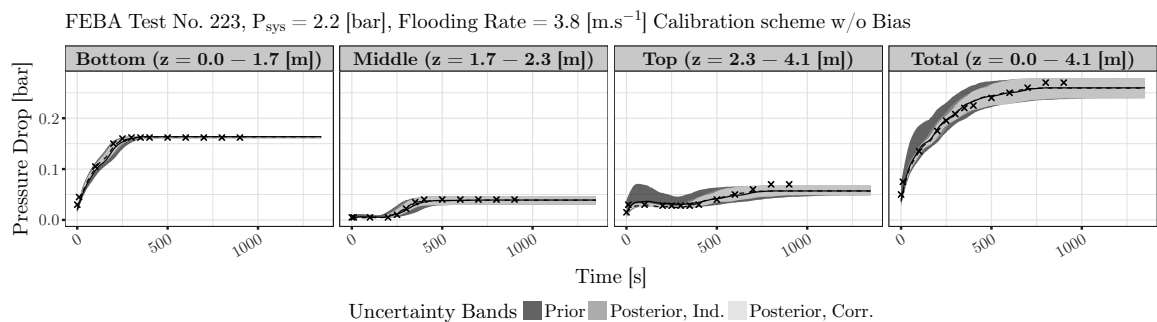


Figure B.52: The posterior samples are from the calibration without model bias term and considering all types of output (w/o Bias).

B.7.10 FEBA Test No. 218, Pressure Drop Output (DP)

FEBA Test No. 218, $P_{\text{sys}} = 2.1$ [bar], Flooding Rate = 5.8 [m.s⁻¹], Calibration scheme w/ Bias, All

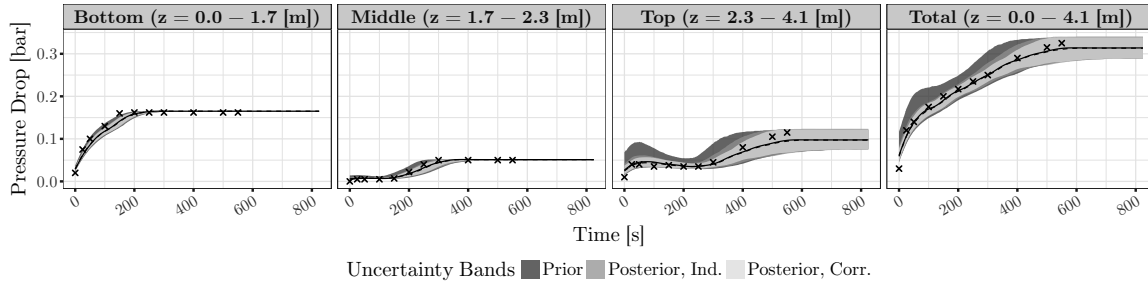


Figure B.53: Propagation of the model parameters uncertainty on FEBA test No. 218 for the pressure drop output (DP) at different axial segments. The uncertainty bands refer to the symmetric 95% probabilities. Solid lines, dashed lines, and crosses indicate the simulation with the nominal parameters values, the median of the posterior, and the experimental data, respectively. The posterior samples are from the calibration with model bias term and considering all types of output (w/ Bias, All).

FEBA Test No. 218, $P_{\text{sys}} = 2.1$ [bar], Flooding Rate = 5.8 [m.s⁻¹], Calibration scheme w/ Bias, no dffbVIHT

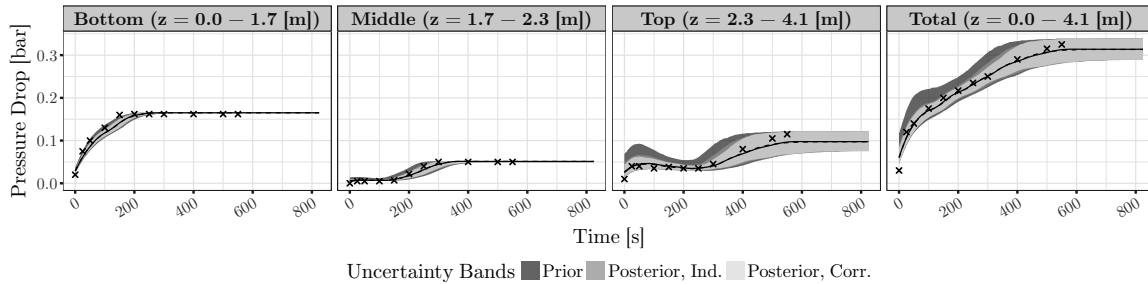


Figure B.54: The posterior samples are from the calibration with model bias term, considering all types of output, but excluding the parameter dffbVIHT (w/ Bias, no dffbVIHT).

FEBA Test No. 218, $P_{\text{sys}} = 2.1$ [bar], Flooding Rate = 5.8 [m.s⁻¹], Calibration scheme w/o Bias

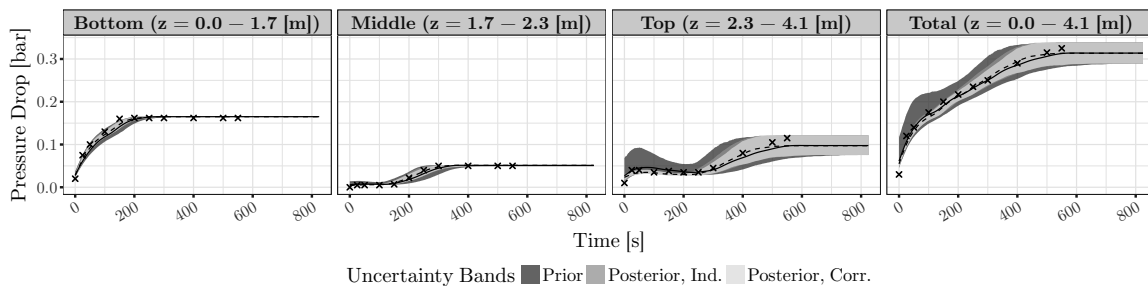


Figure B.55: The posterior samples are from the calibration without model bias term and considering all types of output (w/o Bias).

B.7.11 FEBA Test No. 220, Pressure Drop Output (DP)

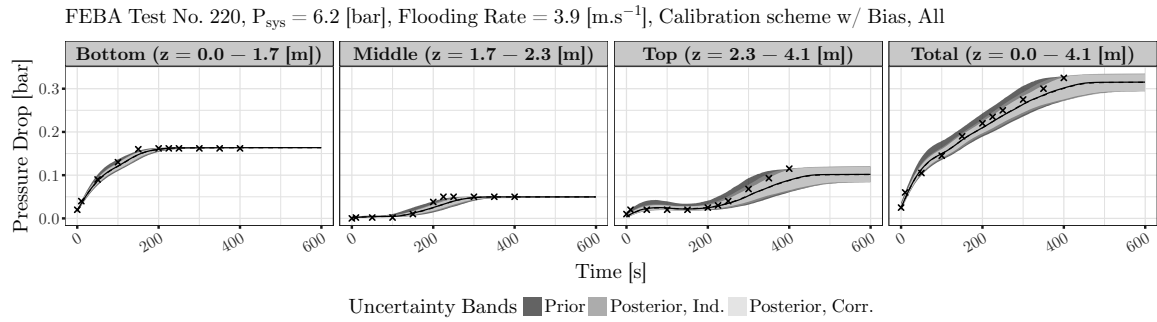


Figure B.56: Uncertainty propagation of the parameters uncertainty of FEBA Test No. 220 for the pressure drop output (DP) at different axial segments. The uncertainty bands refer to the symmetric 95% probabilities. Solid lines, dashed lines, and crosses indicate the simulation with the nominal parameters values, the median of the posterior, and the experimental data, respectively. The posterior samples are from the calibration with model bias term and considering all types of output (w/ Bias, All).

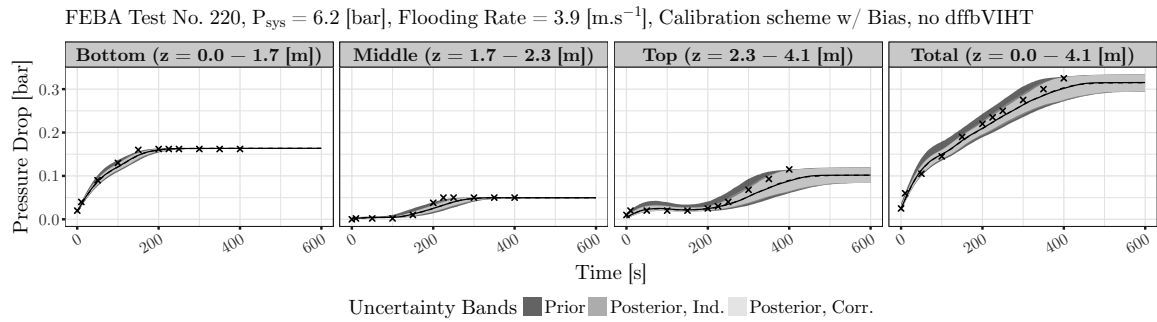


Figure B.57: The posterior samples are from the calibration with model bias term, considering all types of output, but excluding the parameter dffbVIHT (w/ Bias, no dffbVIHT).

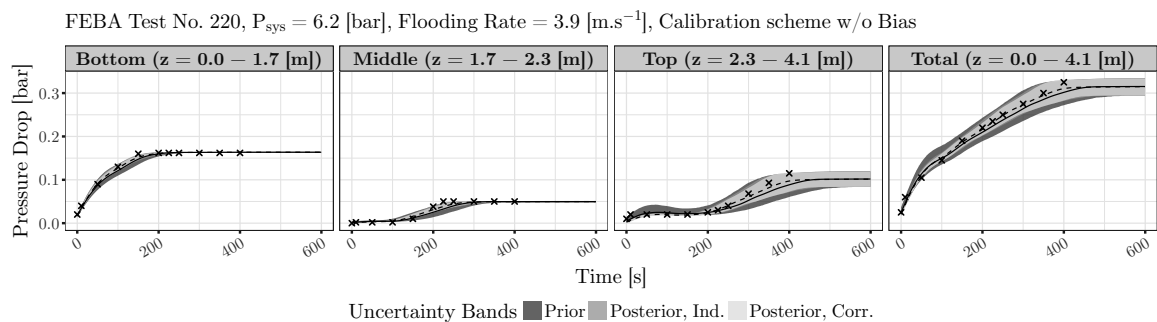


Figure B.58: The posterior samples are from the calibration without model bias term and considering all types of output (w/o Bias).

B.7.12 FEBA Test No. 222, Pressure Drop Output (DP)

FEBA Test No. 222, $P_{\text{sys}} = 6.2$ [bar], Flooding Rate = 5.8 [m.s⁻¹], Calibration scheme w/ Bias, All

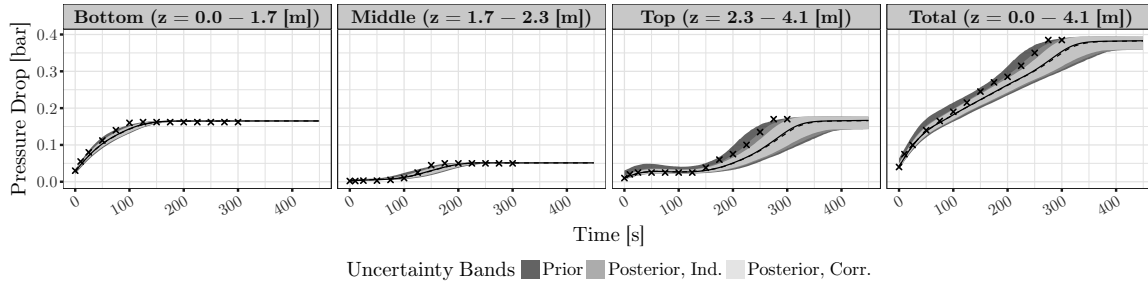


Figure B.59: Propagation of the model parameters uncertainty on FEBA test No. 222 for the pressure drop output (DP) at different axial segments. The uncertainty bands refer to the symmetric 95% probabilities. Solid lines, dashed lines, and crosses indicate the simulation with the nominal parameters values, the median of the posterior, and the experimental data, respectively. The posterior samples are from the calibration with model bias term and considering all types of output (w/ Bias, All).

FEBA Test No. 222, $P_{\text{sys}} = 6.2$ [bar], Flooding Rate = 5.8 [m.s⁻¹], Calibration scheme w/ Bias, no dffbVIHT

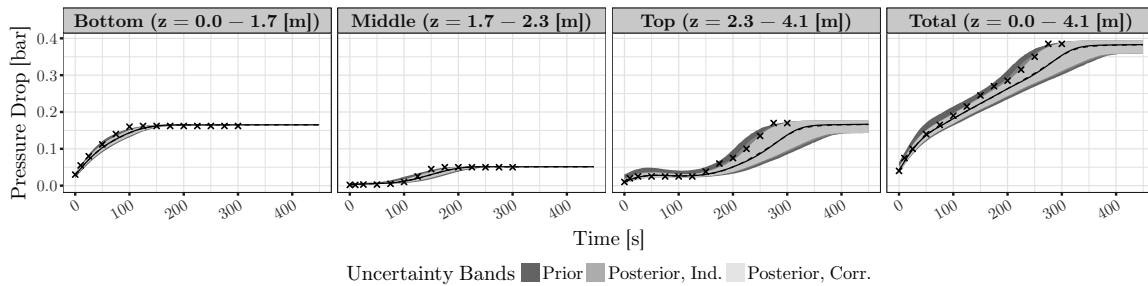


Figure B.60: The posterior samples are from the calibration with model bias term, considering all types of output, but excluding the parameter dffbVIHT (w/ Bias, no dffbVIHT).

FEBA Test No. 222, $P_{\text{sys}} = 6.2$ [bar], Flooding Rate = 5.8 [m.s⁻¹], Calibration scheme w/o Bias

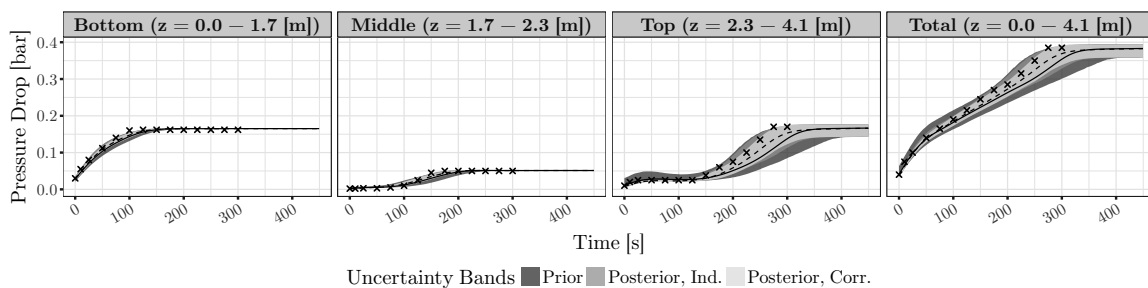


Figure B.61: The posterior samples are from the calibration without model bias term and considering all types of output (w/o Bias).

B.7.13 FEBA Test No. 216, Liquid Carryover Output (CO)

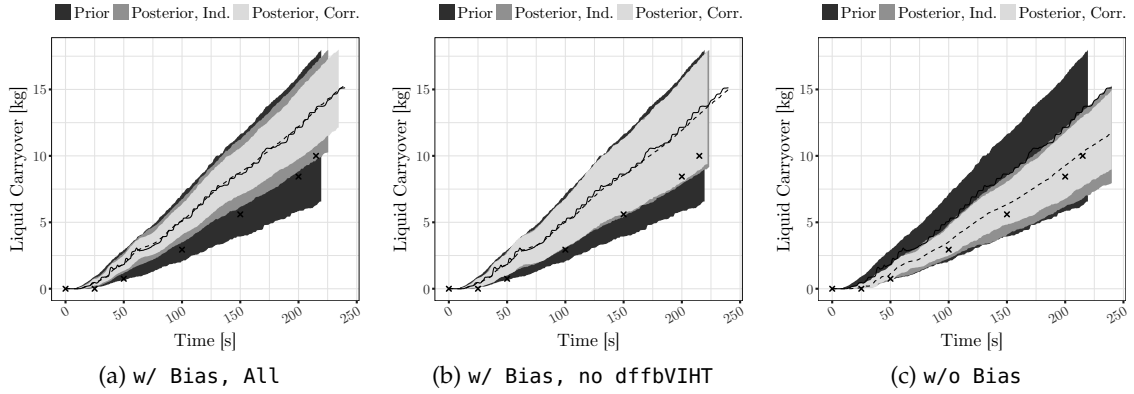


Figure B.62: Propagation of the model parameters uncertainty on FEBA test No. 216 for the liquid carryover outputs (CO) from three different calibration schemes. The uncertainty bands refer to the symmetric 95% probabilities. Solid lines, dashed lines, and crosses indicate the simulation with the nominal parameters values, the median of the posterior, and the experimental data, respectively.

B.7.14 FEBA Test No. 214, Liquid Carryover Output (CO)

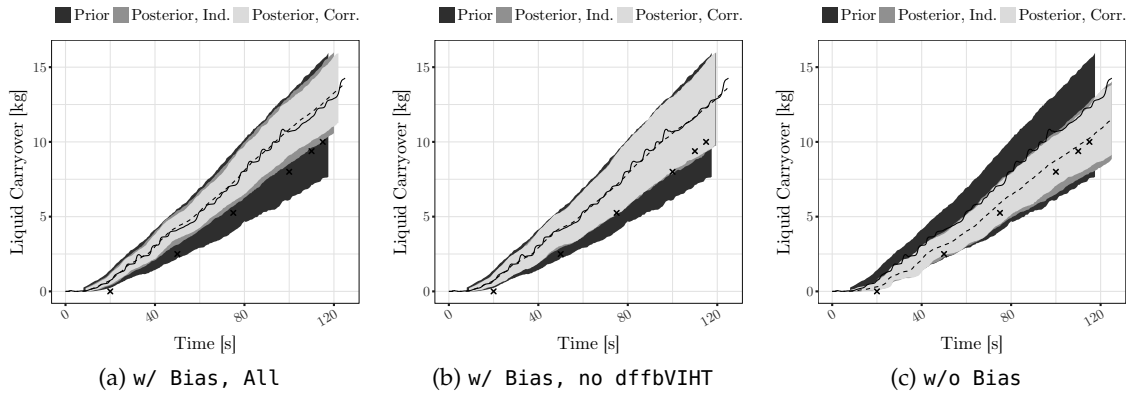


Figure B.63: Propagation of the model parameters uncertainty on FEBA test No. 214 for the liquid carryover outputs (CO) from three different calibration schemes. The uncertainty bands refer to the symmetric 95% probabilities. Solid lines, dashed lines, and crosses indicate the simulation with the nominal parameters values, the median of the posterior, and the experimental data, respectively.

B.7.15 FEBA Test No. 223, Liquid Carryover Output (CO)

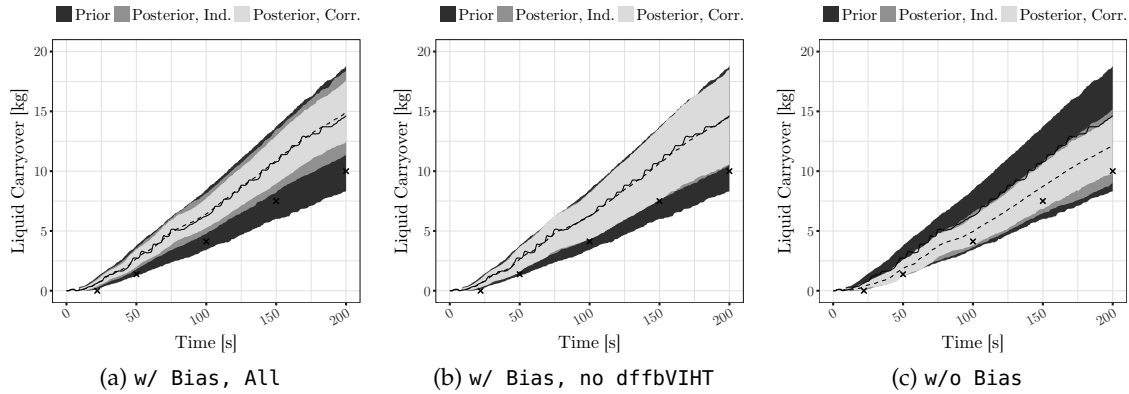


Figure B.64: Propagation of the model parameters uncertainty on FEBA test No. 223 for the liquid carryover outputs (CO) from three different calibration schemes. The uncertainty bands refer to the symmetric 95% probabilities. Solid lines, dashed lines, and crosses indicate the simulation with the nominal parameters values, the median of the posterior, and the experimental data, respectively.

B.7.16 FEBA Test No. 218, Liquid Carryover Output (CO)

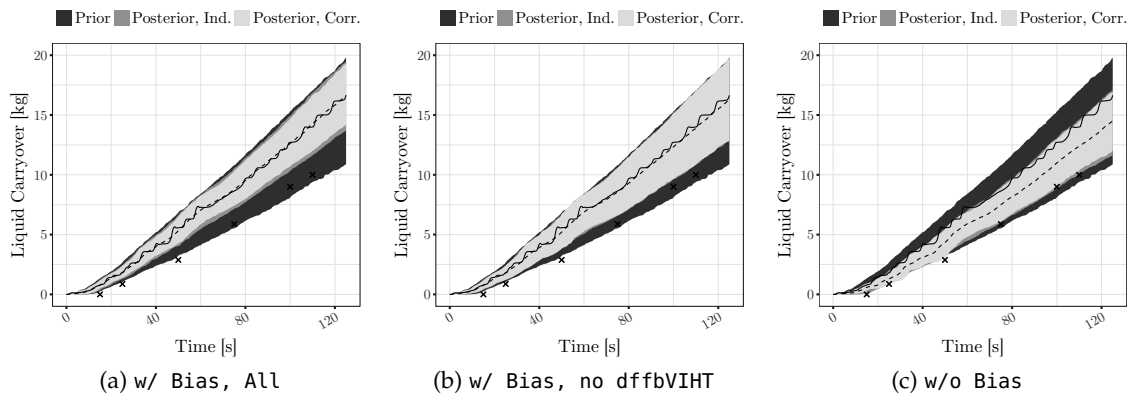


Figure B.65: Propagation of the model parameters uncertainty on FEBA test No. 218 for the liquid carryover outputs (CO) from three different calibration schemes. The uncertainty bands refer to the symmetric 95% probabilities. Solid lines, dashed lines, and crosses indicate the simulation with the nominal parameters values, the median of the posterior, and the experimental data, respectively.

B.7.17 FEBA Test No. 220, Liquid Carryover Output (CO)

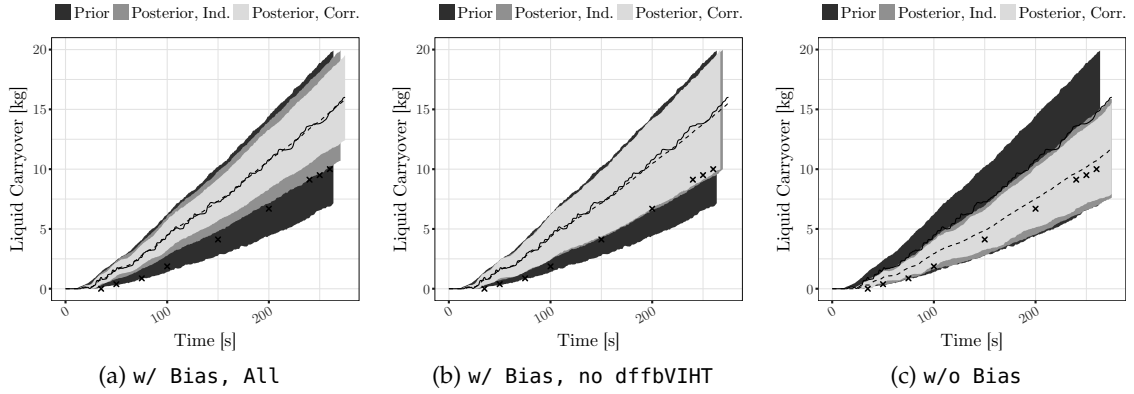


Figure B.66: Propagation of the model parameters uncertainty on FEBA test No. 220 for the liquid carryover outputs (CO) from three different calibration schemes. The uncertainty bands refer to the symmetric 95% probabilities. Solid lines, dashed lines, and crosses indicate the simulation with the nominal parameters values, the median of the posterior, and the experimental data, respectively.

B.7.18 FEBA Test No. 222, Liquid Carryover Output (CO)

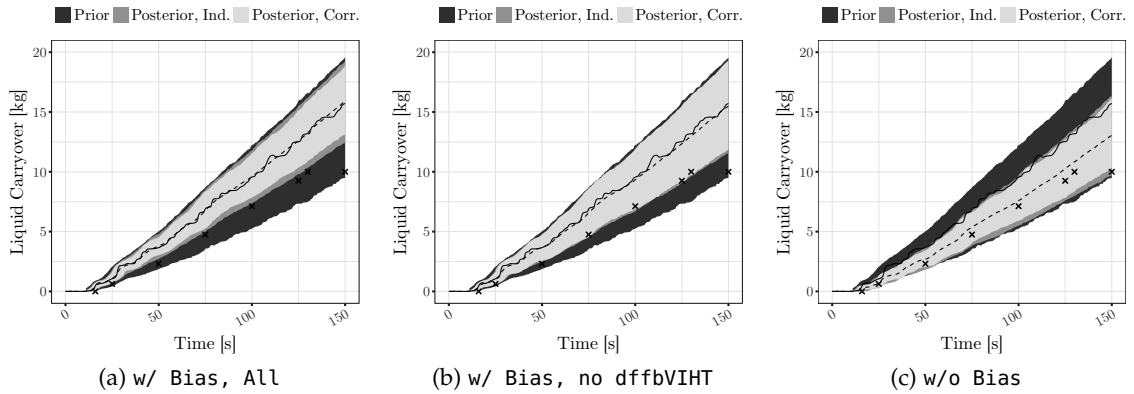


Figure B.67: Propagation of the model parameters uncertainty on FEBA test No. 222 for the liquid carryover outputs (CO) from three different calibration schemes. The uncertainty bands refer to the symmetric 95% probabilities. Solid lines, dashed lines, and crosses indicate the simulation with the nominal parameters values, the median of the posterior, and the experimental data, respectively.

COMPUTATIONAL TOOLS

C.1 gsa-module: PYTHON3 IMPLEMENTATION OF GLOBAL SENSITIVITY ANALYSIS METHODS

gsa-module is a Python3 package implementing several global sensitivity analysis methods for computer/simulation experiments. The implementation is based on a black-box approach where the computer model (or any generic function) is externally implemented to the module itself. The module accepts the model outputs and the design of experiment (optional, only for certain methods) and compute the associated sensitivity measures. The package also includes routines to generate normalized design of experiment file to be used in the simulation experiment based on several algorithms (such as simple random sampling or latin hypercube) as well as simple routines to post-processed multivariate raw code output such as its maximum, minimum, or average.

The general calculation flowchart involved in using the gsa-module can be seen in the figure below.

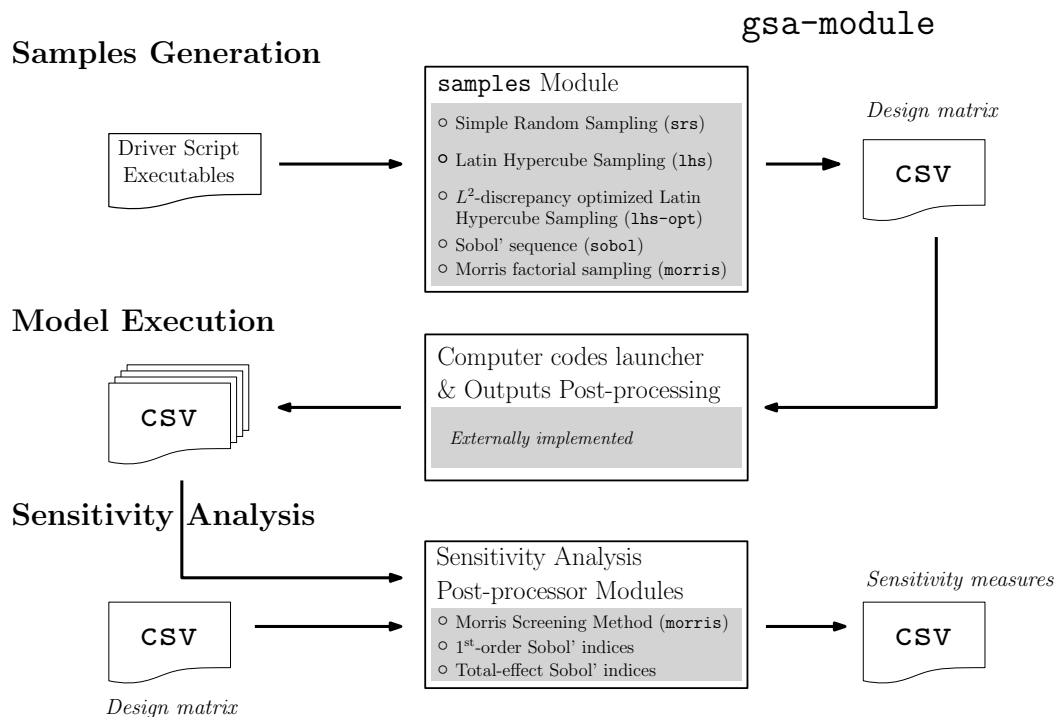


Figure C.1: Flowchart of gsa-module.

MAIN FEATURES

- Capability to generate design of computer experiments using 4 different methods: simple random sampling (SRS), latin hypercube sampling (LHS), sobol' sequence, and optimized LHS using either command line interface `gsa_create_sample` or the module API via `import gsa_module`.
- Sobol' quasi-random number sequence generator is natively implemented in Python3 based on C++ implementation of Joe and Kuo [152].
- Randomization of the Sobol' quasi-random number using random shift procedure.
- Optimization of the latin hypercube design is done via evolutionary stochastic algorithm (ESE) [180].
- Generation of separate test points based on a given design using Hammersley quasi-random sequence [181].
- Capability to generate design of computer experiments for screening analysis (One-at-a-time design), based on the trajectory design [44] and radial design [132]
- Capability to compute the statistics of elementary effects, standardized or otherwise both for trajectory and radial designs. The statistics (mean, mean of absolute, and standard deviation) are used as the basis of parameter importance ranking.
- Capability to estimate the first-order (main effect) Sobol' sensitivity indices using two different estimators (Saltelli et al. [133] and Janon et al. [148]).
- Capability to estimate the total effect Sobol' sensitivity indices using two different estimators (Sobol-Homma [145] and Jansen [149]).
- All estimated quantities are equipped with their bootstrap samples.

REQUIREMENTS, INSTALLATION, AND DOCUMENTATION

The module was developed and tested using the Anaconda Python distribution of Python v3.5. No additional package except the base installation of the distribution is required.

`gsa-module` is hosted on BitBucket. Installation instruction and detailed documentation can be found in the project page.

LICENSE

`gsa-module` is licensed under the MIT License.

C.2 trace-simexp: COMPUTER EXPERIMENT FOR TRACE CODE

A computer experiment, loosely defined, is a multiple computer model runs using different values of the model parameters. Its design, in particular the selection of the design points at which the model will be evaluated; as well as its analysis, in particular the analysis of the output variation in relation to the inputs variation, are useful for sensitivity and uncertainty analyses of the model subjected to the experimentation.

An important prerequisite of carrying out such experiment is the availability of a supporting tool able to handle the related logistical aspects. A Python3-based scripting utility has been developed to assist in carrying such experiments for the thermal-hydraulics system code TRACE. The scope of the utility is ranging from the pre-processing of the TRACE input deck amenable for batch parallel execution to the post-treatment of the resulting binary xtv / dmx file amenable to subsequent sensitivity and uncertainty analyses.

A user interacts with the utility via command line interface. A set of of command line applications corresponding to each of the three processes involved. For reproducibility, an explicit set of parameters are required to be supplied and after successful execution of each, a log file is produced. The log files are also used as a connection between two successive steps. The general flowchart of the processes involved in trace-simexp package is Fig. C.2.

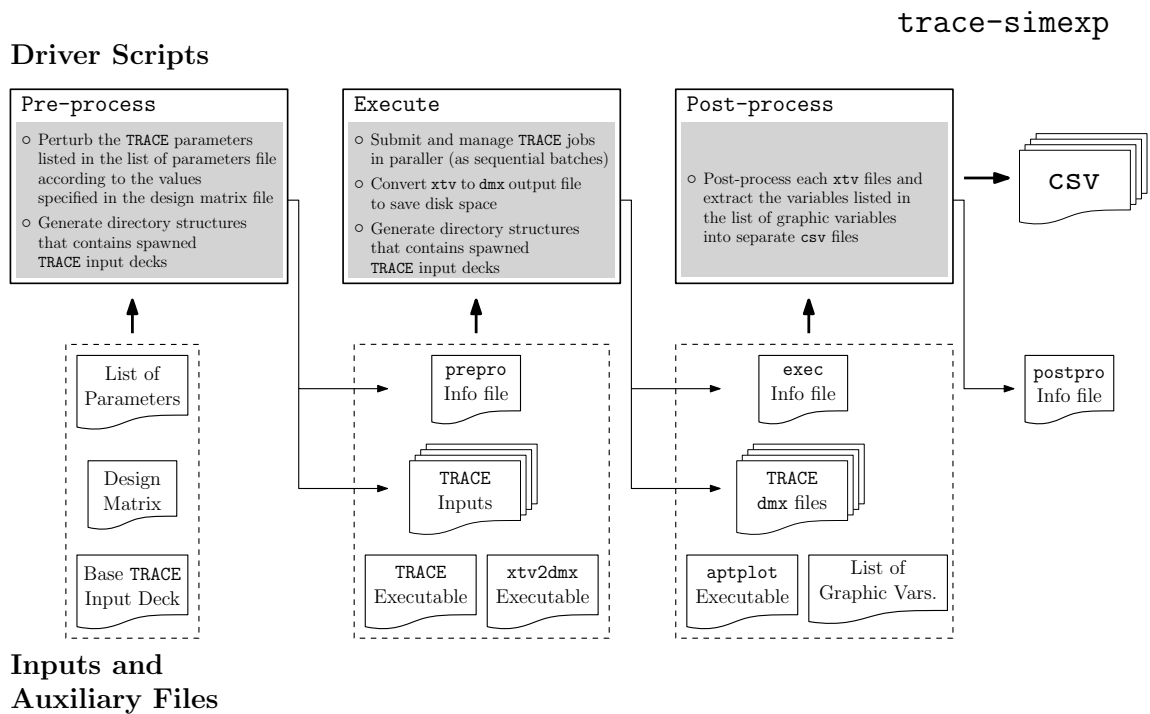


Figure C.2: Flowchart of trace-simexp.

MAIN FEATURES

- Complete separation of the processes in 3 different steps: prepro, exec, and postpro. Interaction is conducted via command line interface.
- Specification of a computer experiment for TRACE by the users is done through a set of input files (list of parameters file, design matrix file, and list of graphic variables file).
- Three modes of parameter perturbation are supported: additive, multiplicative, and substitutive.
- Four categories of TRACE variables in the input deck can be perturbed: spacer grid, material properties, sensitivity coefficient, and components.
- For TRACE components, five are supported: PIPE, VESSEL, POWER, FILL, BREAK.
- Iso-probabilistic transformation of the normalized design matrix is available for uniform, discrete, log-uniform, and normal distributions.

REQUIREMENTS, INSTALLATION, AND DOCUMENTATION

The module was developed and tested using the Anaconda Python distribution of Python v3.5. No additional package except the base installation of the distribution is required.

`trace-simexp` is hosted on BitBucket. Installation instruction and detailed documentation can be found in the project page.

LICENSE

`trace-simexp` is licensed under the MIT License.

SOME USEFUL MATHEMATICAL RESULTS AND RECIPES

D.1 THE SOBOLOV'-SALTIELLI METHOD FOR ESTIMATING VARIANCE-BASED SENSITIVITY INDICES

Sobol' [41] and Saltelli [133] proposed an alternative approach that circumvent the nested structure of Monte Carlo (MC) simulation to estimate sensitivity indices (see Algorithm 1). The formulation starts by expressing the expectation and variance operators in their integral form. As the following formulation is defined on a unit hypercube of D -dimension input parameter space where each parameter is a uniform and independent random variable, explicit writing of the distribution within the integration as well as the integration range are excluded for conciseness.

First, the variance operator shown in the numerator of Eq. (3.24) is written as

$$\begin{aligned} \mathbb{V}_d[\mathbb{E}_{\sim d}[y|\mathcal{X}_d]] &= \mathbb{E}_d[\mathbb{E}_{\sim d}^2[y|\mathcal{X}_d]] - (\mathbb{E}_d[\mathbb{E}_{\sim d}[y|\mathcal{X}_d]])^2 \\ &= \int \mathbb{E}_{\sim d}^2[y|\mathcal{X}_d] dx_d - \left(\int \mathbb{E}_{\sim d}[y|\mathcal{X}_d] dx_d \right)^2 \end{aligned} \quad (\text{D.1})$$

The notation $\mathbb{E}_{\sim o}[\circ|\circ]$ was already explained in Section 3.4.1, while $\mathbb{E}_o[\circ]$ corresponds to the marginal expectation operator where the integration is carried out over the range of parameters specified in the subscript.

Next, consider the term conditional expectation shown in Eq. (D.1), which per definition reads

$$\mathbb{E}_{\sim d}[y|\mathcal{X}_d] = \int f(\mathbf{x}_{\sim d}, x_d) dx_{\sim d} \quad (\text{D.2})$$

Note that $\mathbf{x} = \{\mathbf{x}_{\sim d}, x_d\}$, such that

$$\int \mathbb{E}_{\sim d}[y|\mathcal{X}_d] dx_d = \int \int f(\mathbf{x}_{\sim d}, x_d) dx_{\sim d} dx_d = \int f(\mathbf{x}) dx \quad (\text{D.3})$$

Following the first term of Eq. (D.1), by squaring Eq. (D.2) and by defining a dummy vector variable $\mathbf{x}'_{\sim d}$, the product of the two integrals can be written in terms of a single multiple integrals

$$\begin{aligned} \mathbb{E}_{\sim d}^2[y|\mathcal{X}_d] &= \int f(\mathbf{x}_{\sim d}, x_d) dx_{\sim d} \cdot \int f(\mathbf{x}_{\sim d}, x_d) dx_{\sim d} \\ &= \int \int f(\mathbf{x}'_{\sim d}, x_d) f(\mathbf{x}_{\sim d}, x_d) dx'_{\sim d} dx_{\sim d} \end{aligned} \quad (\text{D.4})$$

Returning to the full definition of variance of conditional expectation in Eq. (D.1),

$$\begin{aligned} \mathbb{V}_d[\mathbb{E}_{\sim d}[y|\mathcal{X}_d]] &= \iint f(\mathbf{x}'_{\sim d}, x_d) f(\mathbf{x}_{\sim d}, x_d) d\mathbf{x}'_{\sim d} dx_d \\ &\quad - \left(\int f(\mathbf{x}) d\mathbf{x} \right)^2 \end{aligned} \quad (\text{D.5})$$

Finally, the main-effect sensitivity index can be written as an integral as follows:

$$\begin{aligned} S_d &= \frac{\mathbb{V}_d[\mathbb{E}_{\sim d}[y|\mathcal{X}_d]]}{\mathbb{V}[y]} \\ &= \frac{\iint f(\mathbf{x}'_{\sim d}, x_d) f(\mathbf{x}_{\sim d}, x_d) d\mathbf{x}'_{\sim d} dx_d - \left(\int f(\mathbf{x}) d\mathbf{x} \right)^2}{\int f(\mathbf{x})^2 d\mathbf{x} - \left(\int f(\mathbf{x}) d\mathbf{x} \right)^2} \end{aligned} \quad (\text{D.6})$$

The integral form given above dispenses with the nested structure of multiple integrals in the original definition of main-effect index. The multidimensional integration is over $(2 \times D - 1)$ -dimension and it is the basis of estimating sensitivity index using Monte Carlo (MC) simulation in this thesis, hereinafter referred to as the Sobol'-Saltelli method. The same procedure applies to derive the total-effect index which yields,

$$\begin{aligned} ST_d &= \frac{\mathbb{E}_{\sim d}[\mathbb{V}_d[y|\mathcal{X}_{\sim d}]]}{\mathbb{V}[y]} \\ &= \frac{\int f^2(\mathbf{x}) d\mathbf{x} - \iint f(\mathbf{x}_{\sim d}, x'_d) f(\mathbf{x}_{\sim d}, x_d) dx'_d dx_d}{\int f^2(\mathbf{x}) d\mathbf{x} - \left(\int f(\mathbf{x}) d\mathbf{x} \right)^2} \end{aligned} \quad (\text{D.7})$$

D.2 MULTIVARIATE RANDOM VARIABLE (RANDOM VECTOR)

A collection of finite D continuous random variables (or random vector) $\mathcal{X} = [X_1, X_2, \dots, X_D] \in \mathbf{X} \subseteq \mathbb{R}^D$ is *jointly continuous* if a non-negative *joint probability density function* $p_{\mathcal{X}} : \mathcal{X} \subseteq \mathbb{R}^D \mapsto \mathbb{R}_{\geq 0}$ exists such that, for any set of $B \in \mathbf{X} \subseteq \mathbb{R}^D$, the probability of \mathcal{X} belonging to B is defined as,

$$\mathbb{P}(\mathcal{X} \in B) = \int_{\mathbf{x} \in B} p_{\mathcal{X}}(\mathbf{x}) d\mathbf{x} \quad (\text{D.8})$$

Additionally, the joint density function is also required to sum up to 1.0 over the whole domain \mathcal{X} for it to be a valid probability density function. In other words, the probability of \mathcal{X} belonging to the domain \mathbf{X} is 1.0,

$$\mathbb{P}(\mathcal{X} \in \mathbf{X}) = \int_{\mathbf{x} \in \mathbf{X}} p_{\mathcal{X}}(\mathbf{x}) d\mathbf{x} = 1.0 \quad (\text{D.9})$$

In this thesis, the type of random variable is restricted to continuous random variable and the term *probability* is often used referring to

the *probability density*. When the distinction is required (such as in the definition Eq. (D.8) and the condition Eq. (D.9) above) the notations used are p and \mathbb{P} for density and probability, respectively. Furthermore, the density function of random vector \mathbf{X} written as $p_{\mathbf{X}}(\mathbf{x})$ is shortened simply to $p(\mathbf{x})$ as it is often clear from the context.

partitioning random vector

Now suppose that \mathbf{X} is partitioned into two disjoint sets \mathbf{X}_A and \mathbf{X}_B whose number of elements $\text{card}(\mathbf{X}_A)$ and $\text{card}(\mathbf{X}_B)$ are non-zero such that $\mathbf{X} = [\mathbf{X}_A, \mathbf{X}_B]$; $p(\mathbf{x}) = p(\mathbf{x}_a, \mathbf{x}_b)$; and $\mathbf{x}_A \in \mathbf{X}_A \subseteq \mathbb{R}^{D_1}$, $\mathbf{x}_B \in \mathbf{X}_B \subseteq \mathbb{R}^{D_2}$, with $D_1 + D_2 = D$.

marginal probability

The *marginal probability* of \mathbf{X}_A is defined as

$$p(\mathbf{x}_A) = \int p(\mathbf{x}_A, \mathbf{x}_B) d\mathbf{x}_B \quad (\text{D.10})$$

where the integration is carried out only on the domain of random variables \mathbf{X}_B , $\mathbf{X}_B \subseteq \mathbb{R}^{D_2}$. Note that if $\text{card}(\mathbf{X}_A) \geq 1$ then $p(\mathbf{x}_A)$ itself is a joint probability. The marginal probability of \mathbf{X}_B follows suit,

$$p(\mathbf{x}_B) = \int p(\mathbf{x}_A, \mathbf{x}_B) d\mathbf{x}_A \quad (\text{D.11})$$

where now the integration is carried out only on the domain of random variables \mathbf{X}_A , $\mathbf{X}_A \subseteq \mathbb{R}^{D_1}$.

conditional probability

The *conditional probability* of \mathbf{X}_A given (or conditioned on) \mathbf{X}_B is defined as,

$$p(\mathbf{x}_A|\mathbf{x}_B) = \frac{p(\mathbf{x}_A, \mathbf{x}_B)}{p(\mathbf{x}_B)} \quad (\text{D.12})$$

for $p(\mathbf{x}_B) > 0$. That is, the notion of conditional probability cannot be defined given an impossible event, $p(\mathbf{x}_B) = 0$. The definition of the conditional probability of \mathbf{X}_B given (or conditioned on) \mathbf{X}_A follows suit,

$$p(\mathbf{x}_B|\mathbf{x}_A) = \frac{p(\mathbf{x}_A, \mathbf{x}_B)}{p(\mathbf{x}_A)} \quad (\text{D.13})$$

for $p(\mathbf{x}_A) > 0$.

Random variables \mathbf{X}_A and \mathbf{X}_B are said to be *independent* of each other if and only if their joint probability $p(\mathbf{x}_A, \mathbf{x}_B)$ is defined as,

independence

$$p(\mathbf{x}_A, \mathbf{x}_B) = p(\mathbf{x}_A)p(\mathbf{x}_B) \quad (\text{D.14})$$

that is, it is the product of the marginals $p(\mathbf{x}_A)$ and $p(\mathbf{x}_B)$. The random variables \mathbf{X}_A and \mathbf{X}_B are said to be *dependent* otherwise. Also, following Eq. (D.12) and Eq. (D.13), the two random variables are independent from each other if and only the marginal is equal to the conditional, $p(\mathbf{x}_A) = p(\mathbf{x}_A|\mathbf{x}_B)$ and $p(\mathbf{x}_B) = p(\mathbf{x}_B|\mathbf{x}_A)$.

Random variables \mathbf{X}_A and \mathbf{X}_B are said to be *exchangeable* if and only if their joint probability is symmetric [262], that is

exchangeability

$$p(\mathbf{x}_A, \mathbf{x}_B) = p(\mathbf{x}_B, \mathbf{x}_A) \quad (\text{D.15})$$

Bayes' Theorem

The exchangeability of random variables (Eq. (D.15)) combined with the definition of conditional probability (Eqs. (D.12) and (D.13)) lead to the *Bayes' Theorem*,

$$\begin{aligned} p(\mathbf{x}_A|\mathbf{x}_B) &= \frac{p(\mathbf{x}_B|\mathbf{x}_A)p(\mathbf{x}_A)}{p(\mathbf{x}_B)} \\ p(\mathbf{x}_B|\mathbf{x}_A) &= \frac{p(\mathbf{x}_A|\mathbf{x}_B)p(\mathbf{x}_B)}{p(\mathbf{x}_A)} \end{aligned} \tag{D.16}$$

D.3 GAUSSIAN RANDOM VECTOR (MULTIVARIATE NORMAL RANDOM VARIABLE)

A Gaussian random vector is a vector with random elements that are *jointly* Gaussian. That is, the random variables have a multivariate normal (MVN) distribution. It is the most widely studied and applied multivariate random variable. There are a couple of reasons for this. From a practical viewpoint, the MVN distribution is tractable and its special properties are well known [263]. From an epistemological viewpoint, modeling a variable as a MVN distribution is a particular way of quantifying uncertainty about that variable. Specifically, if only the mean and variance are of interest then the MVN distribution is the most consistent and parsimonious distribution to describe the variable [95]. This section reviews the definition and some of the most important properties of MVN random variable relevant in the present study.

multivariate normal distribution

A D-dimensional random vector \mathcal{Z} whose elements are random variables, $\mathcal{Z} = [z_1, \dots, z_D] \in \mathbb{R}^D$, is said to have an MVN distribution with mean vector $\boldsymbol{\mu} \in \mathbb{R}^D$ and variance-covariance matrix $\boldsymbol{\Sigma}$, if its joint probability density function is given by,

$$p(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2} |\boldsymbol{\Sigma}|^{1/2}} \exp \left[-\frac{1}{2} (\mathbf{z} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{z} - \boldsymbol{\mu}) \right] \tag{D.17}$$

The joint distribution of a Gaussian random vector is parameterized and fully specified by the mean vector $\boldsymbol{\mu}$ and the variance-covariance matrix $\boldsymbol{\Sigma}$. The symbol “;” separates the value of the variates \mathbf{z} from the parameters of the distribution. A D-dimensional random vector \mathcal{Z} distributed as a joint Gaussian is denoted by,

$$\mathcal{Z} \sim \mathcal{N}_D (\boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{D.18}$$

mean vector

The mean vector $\boldsymbol{\mu}$ is defined as,

$$\boldsymbol{\mu} = [\mathbb{E}[z_1], \dots, \mathbb{E}[z_D]]^T \tag{D.19}$$

where $\mathbb{E}[\circ]$ is the expectation operator, such that $\mathbb{E}[z] = \int_{\mathbf{z}} zp(\mathbf{z})d\mathbf{z}$.

The variance-covariance matrix $\boldsymbol{\Sigma}$ is an element in the space of symmetric positive semi-definite (PSD) $D \times D$ matrices S_{++}^D , which is defined as

$$S_{++}^D = \{\boldsymbol{\Sigma} \in \mathbb{R}^{D \times D} : \boldsymbol{\Sigma} = \boldsymbol{\Sigma}^T, \mathbf{z}^T \boldsymbol{\Sigma} \mathbf{z} \geq 0, \forall \mathbf{z} \in \mathbb{R}^D \text{ and } \mathbf{z} \neq 0\} \tag{D.20}$$

The diagonal elements of the variance-covariance matrix, $\Sigma_{i,i}$, describe the variance of a single random variable, while the off-diagonal elements, $\Sigma_{i,j}$, describe the covariation between a pair of random variables,

variance-covariance matrix

$$\Sigma = \begin{pmatrix} \mathbb{V}[Z_1] & \cdots & \text{Cov}[Z_1, Z_D] \\ \vdots & \ddots & \vdots \\ \text{Cov}[Z_D, Z_1] & \cdots & \mathbb{V}[Z_D] \end{pmatrix} \quad (\text{D.21})$$

where $\mathbb{V}[\circ]$ is the variance operator, such that $\mathbb{V}[Z] = \mathbb{E}[(Z - \mathbb{E}[Z])^2]$; and $\text{Cov}[\circ, \circ]$ is the covariance operator, such that $\text{Cov}[Z, Z^*] = \mathbb{E}[(Z - \mathbb{E}[Z])(Z^* - \mathbb{E}[Z^*])]$.

Suppose that the D -dimensional random vector \mathcal{Z} is partitioned into two sub-vectors (disjoint sets) of D_1 -dimensional random vector \mathcal{Z}_A and D_2 -dimensional random vector \mathcal{Z}_B , such that $\mathcal{Z} = [\mathcal{Z}_A, \mathcal{Z}_B]$ and $D = D_1 + D_2$ (see Appendix D.2). Then the Gaussian random vector $[\mathcal{Z}_A, \mathcal{Z}_B]$ is written,

Gaussian random vector partition

$$\begin{bmatrix} \mathcal{Z}_A \\ \mathcal{Z}_B \end{bmatrix} \sim \mathcal{N} \left(\begin{bmatrix} \boldsymbol{\mu}_A \\ \boldsymbol{\mu}_B \end{bmatrix}, \begin{pmatrix} \boldsymbol{\Sigma}_{A,A} & \boldsymbol{\Sigma}_{A,B} \\ \boldsymbol{\Sigma}_{B,A} & \boldsymbol{\Sigma}_{B,B} \end{pmatrix} \right) \quad (\text{D.22})$$

where $\boldsymbol{\mu}_A$ and $\boldsymbol{\mu}_B$ are the D_1 -dimensional and D_2 -dimensional mean vectors of \mathcal{Z}_A and \mathcal{Z}_B , respectively; and $\boldsymbol{\Sigma}_{A,A}$, $\boldsymbol{\Sigma}_{A,B}$, $\boldsymbol{\Sigma}_{B,A}$, $\boldsymbol{\Sigma}_{B,B}$ are the $D_1 \times D_1$, $D_1 \times D_2$, $D_2 \times D_1$, and $D_2 \times D_2$ sub-matrices of the partitioned covariance matrix, respectively. So for instance,

$$\boldsymbol{\Sigma}_{A,B} = \begin{pmatrix} \text{Cov}[Z_1, Z_{D_1+1}] & \cdots & \text{Cov}[Z_1, Z_D] \\ \vdots & \ddots & \vdots \\ \text{Cov}[Z_{D_1}, Z_{D_1+1}] & \cdots & \text{Cov}[Z_{D_1}, Z_D] \end{pmatrix}$$

The marginal density of \mathcal{Z}_A follows an MVN distribution given by,

Gaussian identity: marginal density

$$p(\mathbf{z}_A) = \frac{1}{(2\pi)^{D_1/2} |\boldsymbol{\Sigma}_{A,A}|^{1/2}} \exp \left[-\frac{1}{2} (\mathbf{z}_A - \boldsymbol{\mu}_A)^\top \boldsymbol{\Sigma}_{A,A}^{-1} (\mathbf{z}_A - \boldsymbol{\mu}_A) \right] \quad (\text{D.23})$$

The results are analogous for the marginal of \mathcal{Z}_B , $p(\mathbf{z}_B)$ of which all the subscripts A are replaced by B .

The conditional density of \mathcal{Z}_A conditioned on \mathcal{Z}_B , again, also follows an MVN distribution given as,

Gaussian identity: conditional density

$$\begin{aligned} p(\mathbf{z}_A | \mathbf{z}_B) &= \frac{1}{(2\pi)^{D_1/2} |\boldsymbol{\Sigma}_{A,A}^*|^{1/2}} \exp \left[-\frac{1}{2} (\mathbf{z}_A - \boldsymbol{\mu}_A^*)^\top \boldsymbol{\Sigma}_{A,A}^{-1*} (\mathbf{z}_A - \boldsymbol{\mu}_A^*) \right] \\ \boldsymbol{\mu}_A^* &= \boldsymbol{\mu}_A + \boldsymbol{\Sigma}_{A,B} \boldsymbol{\Sigma}_{B,B}^{-1} (\mathbf{z}_B - \boldsymbol{\mu}_B) \\ \boldsymbol{\Sigma}_{A,A}^* &= \boldsymbol{\Sigma}_{A,A} - \boldsymbol{\Sigma}_{A,B} \boldsymbol{\Sigma}_{B,B}^{-1} \boldsymbol{\Sigma}_{A,B}^\top \end{aligned} \quad (\text{D.24})$$

The results are analogous for the conditional \mathcal{Z}_B given \mathcal{Z}_A , $p(\mathbf{z}_B | \mathbf{z}_A)$ of which all the subscripts A are replaced by B , and vice versa.

D.4 INVERSE TRANSFORM SAMPLING

The inverse transform sampling provides a simple approach to generate samples of a univariate non-uniform random variable. The following justification of the method is adapted from the proposition that can be found in (pp. 432, [216]). Here, it is assumed that the (pseudo)-random number generator for a uniform variable $\mathcal{U} \sim \text{unif}(0, 1)$ is readily accessible.

Let \mathcal{X} be a random variable with distribution function $F(x)$ where $F: x \in \mathbf{X} \mapsto [0, 1]$ and a non-decreasing function. Then:

- If $F(x)$ continuous then $F(\mathcal{X}) \sim \text{unif}(0, 1)$
- For non-continuous $F(x)$ the condition $\mathbb{P}(F(\mathcal{X}) \leq t) \leq t, \forall t \in [0, 1]$ holds nevertheless.
- If $F^{-1}(y) = \inf\{x : F(x) \geq y, 0 < y < 1\}$ and if $\mathcal{U} \sim \text{unif}[0, 1]$ then

$$F^{-1}(\mathcal{U}) \sim \mathcal{X} \quad (\text{D.25})$$

where F^{-1} the inverse of F is called the *quantile* function of \mathcal{X} .

The proof of these propositions can be found in [216]. What is important here is that a non-uniform random variable is distributed as a transformed uniform random variable and the transformation is done through the quantile function of the non-uniform random variable. This provides a basis for generating samples of a non-uniform random variable given in Algorithm 4. The method requires the quantile function $F^{-1}(x)$ and a uniform random generator $\mathcal{U} \sim \text{unif}[0, 1]$.

Algorithm 4 Inverse Transform Sampling

Generate N samples of \mathcal{X} given $F^{-1}(x)$ and $\mathcal{U} \sim \text{unif}[0, 1]$

Require: $N > 0$, $F^{-1}(x)$, and \mathcal{U}

for $n = 1$ to N **do**
 sample u from \mathcal{U}
 $x^{(n)} \leftarrow F^{-1}(u)$
end for

As an illustrative example of this method, consider the problem of generating samples from a long-tail distribution called the Gumbel distribution parameterized by location parameter x_0 and scale parameter β [264],

$$\begin{aligned} \mathcal{X} &\sim \text{Gumbel}(x_0, \beta); x_0 \in \mathbb{R}, \beta \in \mathbb{R}_{\geq 0} \\ p(x) &= \frac{1}{\beta} \exp[-(z + \exp[-z])] \\ F(x) &= \exp[-\exp[-z]] \\ z &= \frac{x - x_0}{\beta}; x \in \mathbb{R} \\ F^{-1}(u) &= x_0 - \beta \ln\left(\ln \frac{1}{u}\right); u \in [0, 1] \end{aligned} \quad (\text{D.26})$$

To generate samples from the above distribution, first generate N samples from a uniform distribution and then transform the sampled values using F^{-1} . The resulting transformed values are samples distributed as the specified Gumbel distribution. Fig. D.1 illustrates this procedure and its result for $x_0 = 0.0$ and $\beta = 10.0$.

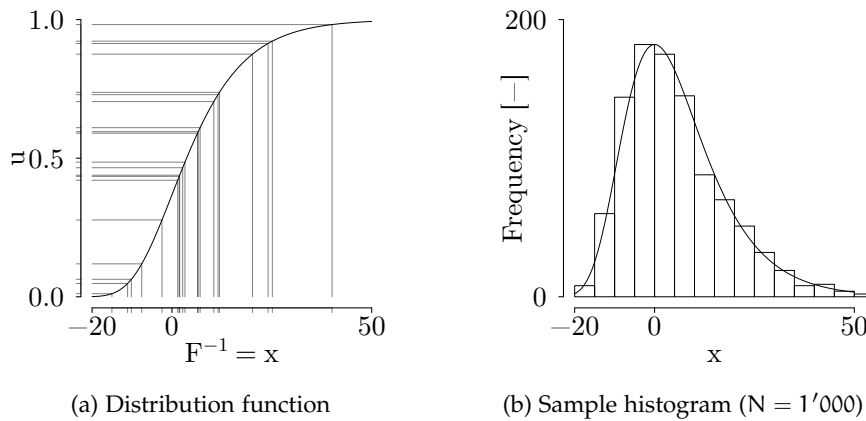


Figure D.1: Illustration of inverse transform sampling for the density given in Eq.(D.26). First, samples from uniform random variable $u \sim \text{unif}(0, 1)$ are generated. The transformation of the uniform samples by F^{-1} (Left) will then yield samples distributed as required (Right). The analytical density have been normalized to match the peak of the histogram.

D.5 GENERATING SAMPLES FROM A MULTIVARIATE NORMAL DISTRIBUTION

Drawing n number of samples from an m -variate Normal distribution, $\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ having an arbitrary m -dimensional mean vector $\boldsymbol{\mu}$ and an arbitrary, $m \times m$ covariance matrix $\boldsymbol{\Sigma}$ can be achieved by using a (univariate) standard normal random number generator available in most numerical computing environment. The procedure is as follows [216]:

1. Factorize the covariance matrix $\boldsymbol{\Sigma}$ using the Cholesky decomposition,

$$\boldsymbol{\Sigma} = \mathbf{L}\mathbf{L}^T \tag{D.27}$$

where \mathbf{L} and \mathbf{L}^T are the Cholesky factor and its transpose, respectively. \mathbf{L} is a lower triangular matrix.

2. Generate vector $\mathbf{z} = (z_1, z_2, \dots, z_m)^T$ by taking m random draws from a standard normal random generator, $u \sim \mathcal{N}(0, 1)$.
3. Transform the vector \mathbf{z} by the following formula,

$$\mathbf{x} = \boldsymbol{\mu} + \mathbf{L}\mathbf{z} \tag{D.28}$$

where the m -dimensional vector \mathbf{x} is a single realization of the specified m -variate normal random variable.

4. Repeat Step 2 and Step 3 n times to obtain the desired number of samples

D.6 LANDMARK REGISTRATION AND TIME WARPING FUNCTION

The most straightforward curve registration procedure is the landmark registration/marker registration [265]. *Landmarks* are salient features of a curve that can be observed or expected to occur in a set of curves. In the context of reflood simulations, examples of such landmarks are the time of maximum temperature and the time of quenching. A transformation of time for each curve is carried out such that these features are aligned with respect to a reference curve.

The landmark registration problem can be expressed as the following: Let $\{y_i(t); i = 1, 2, \dots, N; t \in [t_a, t_b]\}$ be a set of continuous functions defined over the domain t_a to t_b . Let $\{y_{ref}(t_{ref,j}); j = 1, \dots, M\}$ be a set of M landmarks of a given reference function $y_{ref}(t)$. Then a set of time warping functions $\{h_i(t); i = 1, 2, \dots, N; t \in [t_a, t_b]\}$ for each curve in the data set can be defined. These functions have the following properties:

1. Each $h_i(t)$ is defined in the same domain as the domain of the original curve $y_i(t)$. That is, $t \in [t_a, t_b]$;
2. Each $h_i(t)$ satisfies the boundary conditions,

$$\begin{aligned} h_i(t_a) &= t_a \\ h_i(t_b) &= t_b; \end{aligned} \tag{D.29}$$

3. Each $h_i(t)$ is a strictly increasing function. The first implication of this property is that the time transformation process cannot alter the ordering of the landmarks. In other words, time is strictly increasing both in the original and the transformed frames. The second implication is that the time warping function is an invertible function such that for the same event there exists a unique pair of time and its transformed value.
4. Each $h_i(t)$ transforms the time t_{ref} of the reference curve $y_{ref}(t)$ such that the timing of the M reference landmarks are aligned with respect to the landmarks of each curve,

$$\begin{aligned} h_i(t_{ref,j}) = t_{i,j} &\iff h_i^{-1}[t_{i,j}] = t_{ref,j}; \forall i, j \\ y_i[h_i(t_{ref})] &= y_i(t_i) \equiv y_i^*(t_{ref}); \forall i \end{aligned} \tag{D.30}$$

where y_i^* is the *registered* function of curve i , whose time scale is the same as the reference curve; $t_{i,j}$ is the timing of the land-

mark j of curve i ; and $h_i^{-1}(t)$, the inverse of the warping function, is the *aligning* function, as it *aligns* the timing of the landmark j of curve i to the timing of the same landmark of the reference curve. The argument t_{ref} in $y_i^*(t_{ref})$ implies that the registered curve is in the time scale of the reference, which is the same for all the curves. For instance, if $h_i(t_{ref,j}) > t_{ref,j}$ then the landmark j for curve i is delayed and the aligning function accelerates the time for curve i to conform to the reference timing. On the other hand, if $h_i(t_{ref,j}) < t_{ref,j}$ then the landmark j for curve i occurs earlier and the aligning function retards the time for curve i to conform to the reference timing.

The registration problem can then be posed as an estimation problem of each time warping function $h_i(t)$ constrained by the above properties. Following [140], it is solved by using penalized least square regression method. In accordance to the functional data analysis (FDA) framework, the warping function is also represented as a linear combination of B-spline basis functions.

D.7 KARHUNEN-LOÉVE THEOREM

The Karhunen-Loéve theorem establishes that for any centered mean-square continuous stochastic process \mathcal{Y} defined by a sample space Ω and on a domain $\mathcal{D} \subseteq \mathbb{R}$, there exists a set of basis functions ξ_j defined on \mathcal{D} such that for all $t \in \mathcal{D}$,

$$\mathcal{Y} = \sum_{j=1}^{+\infty} \theta_j \cdot \xi_j(t) \quad (\text{D.31})$$

The scalar coefficients θ_j in Eq. (D.31) are given for each $\omega \in \Omega$ by $\theta_j(\omega) = \int_{\mathcal{D}} \mathcal{Y}(\omega) \xi_j(t) dt$ and satisfy the following:

$$\begin{aligned} \mathbb{E}[\theta_j] &= 0 \\ \mathbb{V}[\theta_j] &= \rho_j \\ \mathbb{E}[\theta_j \cdot \theta_k] &= \delta_{jk} \rho_j \end{aligned} \quad (\text{D.32})$$

where $\mathbb{E}[\circ]$ and $\mathbb{V}[\circ]$ are the expectation and the variance operators, respectively; δ is the Kronecker delta; ρ_j is the eigenvalue associate with basis function $\xi_j(t)$. Eqs. (D.31) and (D.32) imply that θ_j is independent and identically distributed (i.i.d) with mean 0 and variance ρ_j [266].

The basis function, in turn, is defined as the eigenfunction of the functional operator $\mathbb{K}[f(\circ)]$ on some function $f(\circ)$ applied to $\xi_j(t)$

$$\mathbb{K}[\xi_j(t)] = \int_{\mathcal{D}} R(t, s) \cdot \xi_j(s) ds = \rho_j \cdot \xi_j(t); \forall t \in \mathcal{D} \quad (\text{D.33})$$

where $R(t, s)$ is the covariance function of the stochastic process \mathcal{Y} for the covariance between time t and s , i.e., $R(t, s) \equiv \mathbb{E}[\mathcal{Y}_t \cdot \mathcal{Y}_s]$.

The Karhunen-Loève theorem is applicable to the functional deviation from the proper mean and therefore allows for each element of the data set to be represented as a series that is optimal in the root-mean-square-of-error sense:

$$y_n(t) = \bar{y}(t) + \sum_{j=1}^{+\infty} \theta_{j,n} \cdot \xi_j(t); n = 1, \dots, N \quad (\text{D.34})$$

where $\xi_j(t)$ is the series of orthogonal eigenfunctions (or fPC), and the corresponding fPC score $\theta_{j,n}$ associated with each function realization is defined by the orthogonality condition

$$\theta_{j,n} = \int_{\mathcal{D}} [y_n(t) - \bar{y}(t)] \cdot \xi_j(t) dt \quad (\text{D.35})$$

As can be seen in Eq. (D.34), the transformation is exact if the set of eigenfunctions is infinite, but truncation is needed for practical application. Such details of the actual implementation of functional principal component analysis (fPCA) can be found in Refs. [49, 142, 154].

D.8 DISCRETE-STATE MARKOV CHAIN

This section of the appendix complements the most important theorems of Markov chain in relation to MCMC simulation presented in Section 5.3.2. In the following some of the important notions are first introduced for the discrete-state Markov chain. It provides a more intuitive entry to the theory of Markov chain through matrix notation and a graphical representation. The associated theorems are presented without proof though the list of references are provided.

Markov chain A Markov chain on a discrete-state space \mathcal{S} is defined as a sequence of random variables $\{\mathcal{X}^{(i)}; i \geq 0\}$ where the indices represents successive iterations, *such that the conditional probability distribution of $\mathcal{X}^{(i+1)}$ follows the Markov assumption.* That is,

$$\mathcal{X}^{(i+1)} | \mathcal{X}^{(i)}, \mathcal{X}^{(i-1)}, \dots, \mathcal{X}^{(0)} = \mathcal{X}^{(i+1)} | \mathcal{X}^{(i)} \quad (\text{D.36})$$

Put differently, the future value depends on the past only through the present value [105, 219].

A discrete-state Markov chain is fully defined by its joint probability [219].

$$\begin{aligned} \mathbb{P}(\mathcal{X}^{(i+1)} = x^{(i+1)}, \mathcal{X}^{(i)} = x^{(i)}, \dots, \mathcal{X}^{(0)} = x^{(0)}) = \\ \mathbb{P}(\mathcal{X}^{(0)} = x^{(0)}) \cdot \mathbb{P}(\mathcal{X}^{(1)} = x^{(1)} | \mathcal{X}^{(0)} = x^{(0)}) \cdot \dots \quad (\text{D.37}) \\ \cdot \mathbb{P}(\mathcal{X}^{(i+1)} | \mathcal{X}^{(i)} = x^{(i)}). \end{aligned}$$

The specification consists of three main components:

- The *state space* \mathcal{S} which is the set of all possible outcomes of the random variables $\{\mathcal{X}^{(i)}\}$. The state space considered here is discrete with D elements, $\mathcal{S} = \{x_1, x_2, \dots, x_D\}$. *Discrete state space*
- The *initial* probability distribution $\pi^{(0)}$. This is the (marginal) probability distribution of $\mathcal{X}^{(0)}$ (or the marginal distribution of the chain at $i = 0$). That is, *Initial distribution*

$$\pi^{(0)} = \{\mathbb{P}(\mathcal{X}^{(0)} = x)\} = \{\pi_x\} \quad \forall x \in \mathcal{S} \tag{D.38}$$

In discrete-state Markov chain, the distribution can be expressed as a D -dimensional vector.

- The *transition probability matrix* P which is a $D \times D$ matrix with elements $p_{x,y} \geq 0.0$ and $\sum_y p_{x,y} = 1.0$. Each element is the conditional probability between two states. That is, *Transition probability*

$$p_{x,y} = \mathbb{P}(\mathcal{X}^{(i+1)} = y | \mathcal{X}^{(i)} = x) \quad \forall x, y \in \mathcal{S} \tag{D.39}$$

The transition probability matrix is said to be *stationary* if it does not depend on a particular iteration i . In practice, most MCMC algorithms rely on a stationary transition probability [105]. *Stationary transition probability*

As an example of a discrete-state Markov chain, consider a 3-state Markov chain representing changes of human health condition with $\mathcal{S} = \{\text{Healthy}, \text{Sick}, \text{Dead}\}$ and a transition probability matrix P ,

$$P = \begin{pmatrix} \mathbb{P}(H|H) & \mathbb{P}(S|H) & \mathbb{P}(D|H) \\ \mathbb{P}(H|S) & \mathbb{P}(S|S) & \mathbb{P}(D|S) \\ \mathbb{P}(H|D) & \mathbb{P}(S|D) & \mathbb{P}(D|D) \end{pmatrix} = \begin{pmatrix} 0.75 & 0.20 & 0.05 \\ 0.65 & 0.15 & 0.20 \\ 0.00 & 0.00 & 1.00 \end{pmatrix} \tag{D.40}$$

The Markov chain is graphically represented in Fig. D.2 using a *state transition diagram*.

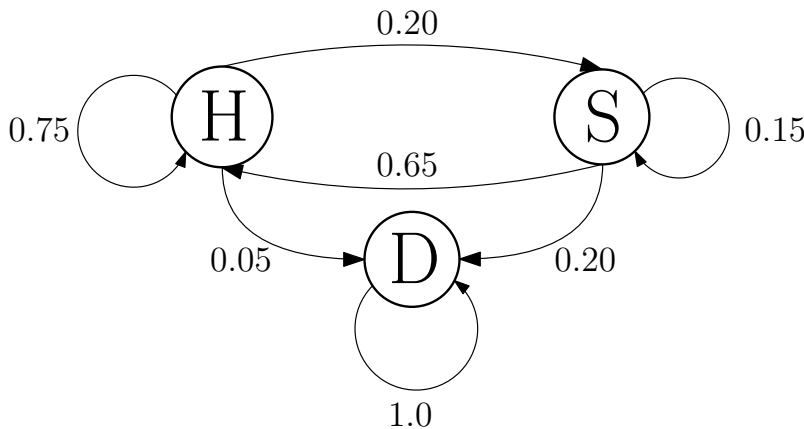


Figure D.2: An illustration of a 3-State Markov chain with the transition probability given by the matrix P in Eq. (D.40).

The probability of transition from state x to y in one iteration is given by,

$$\begin{aligned} \mathbb{P}(\mathcal{X}^{(i+1)} = y) &= \sum_{x \in \mathcal{S}} \mathbb{P}(\mathcal{X}^{(i)} = x) \cdot \mathbb{P}(\mathcal{X}^{(i+1)} = y | \mathcal{X}^{(i)} = x) \\ \Leftrightarrow \pi_y^{(i+1)} &= \sum_{x \in \mathcal{S}} \pi_x^{(i)} p_{x,y} \quad \forall y \in \mathcal{S} \\ \Leftrightarrow \pi^{(i+1)} &= \pi^{(i)} \mathbf{P} \end{aligned} \tag{D.41}$$

Marginal distribution of state at iteration n

Thus, given the three main components, the marginal probability distribution at any given iteration can be defined recursively. That is, the probability distribution at iteration n is given by $\pi^{(n)} = \pi^{(0)} \mathbf{P}^n$.

Irreducibility

A Markov chain is said to be *irreducible* if each state in the state space \mathcal{S} can be reached eventually from any other state [221, 222]. Irreducibility is a property of the transition probability matrix \mathbf{P} (i.e., having an irreducible transition probability matrix). Formally,

$$\forall x, y \in \mathcal{S}, \exists n \geq 0 \text{ for which } p_{x,y}^{(n)} > 0 \tag{D.42}$$

Based on this definition the transition matrix of Eq. (D.40) is not irreducible as the state of being *Dead* does not allow transition to any of the two other states. An example of irreducible chain is given in a graphical representation of Fig. D.3. Note that while state B is not directly connected to state A, the state can eventually be reached from state B through the connection of state C (in this case, n is equal to 2).

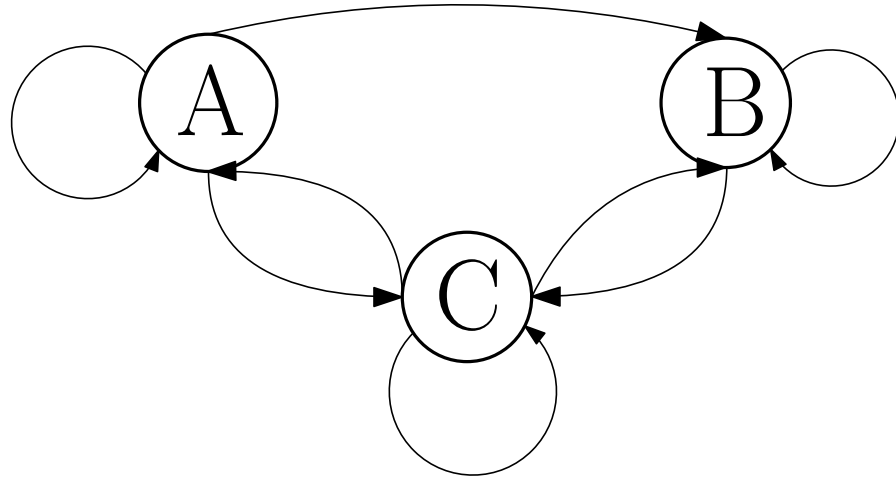


Figure D.3: An illustration of an irreducible 3-State Markov chain.

period of a state

A period of a state $x \in \mathcal{S}$ denoted as d_x is defined for each state in the chain as follows [219],

$$d_x = \text{GCD} \{n : p_{x,x}^{(n)} > 0, n > 0\} \tag{D.43}$$

where GCD stands for the *Greatest Common Divisor* of the set.

In an arbitrary discrete-state Markov chain, different states might have different periods. A state is called *aperiodic* if its period is equal to 1 and it is called *periodic* otherwise. If a chain has the same period $d > 1$ for each of its states then the chain is called *periodic* (see Fig. D.4a for a periodic chain with period 3). A periodic chain exhibits a non-stochastic behavior in their dynamic. On the contrary, a chain having the same period of 1 for each of its states is called a *aperiodic* chain (see Fig. D.4b for an example of a aperiodic chain).

periodic, aperiodic chain

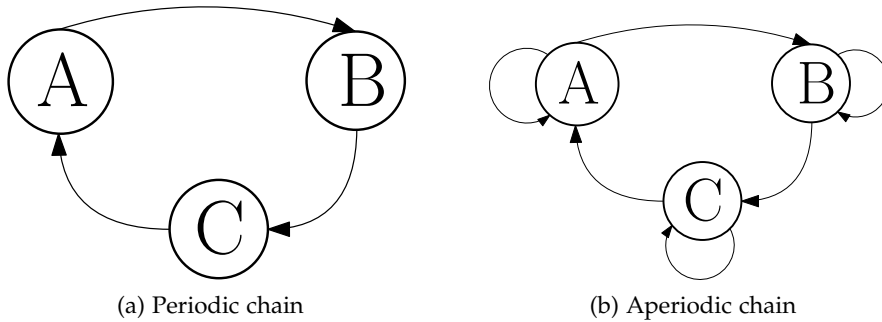


Figure D.4: Examples of periodic and aperiodic chains. (Left) an example of a periodic 3-state Markov chain. In this case, all states have the the same period of 3 iterations. (Right) an example of aperiodic 3-state Markov chain, that is all states are aperiodic.

Some distributions are *stationary* with respect to a transition probability matrix [219, 221, 222]. Specifically, π^* is *stationary for P* if,

Stationary distribution

$$\pi^* = \pi^* P \tag{D.44}$$

Put differently, the distribution is *invariant* under transition. Consequently, if stationary distribution exists, once the chain reaches the stationary distribution, it will remain there and the chain itself becomes stationary. Stationary distribution need not exist for a given P, but in the application of MCMC algorithms, the existence of stationary distribution is guaranteed [105].

Stationary Markov chain

As an example of a stationary distribution, consider once more the transition probability matrix P in Eq. (D.40). For this transition, the distribution $\pi = [0.0, 0.0, 1.0]$ is stationary with respect to P such that

$$\pi = \pi P \Leftrightarrow [0.0, 0.0, 1.0] = [0.0, 0.0, 1.0] \begin{pmatrix} 0.75 & 0.20 & 0.05 \\ 0.65 & 0.15 & 0.20 \\ 0.00 & 0.00 & 1.00 \end{pmatrix} \tag{D.45}$$

Stating that a stationary distribution is being dead, eventually and definitely.

The notions of irreducibility, aperiodicity, and stationarity are cobbled together to arrive at an important result in the discrete-state Markov chain and it is stated here without proof [221]. Let P be a transition probability matrix, irreducible and aperiodic, then P has

Fundamental theorem of Markov chain

exactly one stationary distribution π^* and for any initial distribution $\pi^{(0)}$

$$\lim_{I \rightarrow \infty} |\pi^{(0)} P^I - \pi^*| = 0 \tag{D.46}$$

Ergodic theorem That is, the chain converges *in distribution* to the stationary distribution regardless of its initial distribution. The theorem also indicates the existence of a *limiting distribution* $\lim_{I \rightarrow \infty} \pi^{(0)} P^I$.

Furthermore, under the above condition,

$$\lim_{I \rightarrow \infty} \frac{1}{I} \sum_{i=1}^I \mathbb{I}_{\{\mathcal{X}^{(i)}=x\}} = \pi_x \quad \forall x \in \mathcal{S} \tag{D.47}$$

where \mathbb{I} is the indicator function where it is 1.0 if the condition in the subscript holds, and 0 otherwise. In other words, the chain converges *in probability*. These theorems provide the justification for using samples generated from a Markov chain (whose properties stated above) as samples for Monte Carlo calculation. For proofs of these theorems, refer to [222].

Detailed balance condition

In generating samples from a target distribution, the engineering is done somewhat in reverse (“Given a distribution, construct P”). Thus it is worthwhile to note the *detailed balance* condition which is a central condition for an MCMC algorithm. A Markov chain with a transition probability matrix P satisfies the detailed balance condition if there exists a probability distribution π such that,

$$p_{x,y} \pi_x = p_{y,x} \pi_y \quad \forall x, y \in \mathcal{S} \tag{D.48}$$

Reversible chain

As a result, the chain is said to be *reversible*. Formally,

$$\mathcal{X}^{(i+1)} | \mathcal{X}^{(i)} = x \sim \mathcal{X}^{(i)} | \mathcal{X}^{(i+1)} = x \quad \forall x \in \mathcal{S} \tag{D.49}$$

A reversible chain is a stationary chain. Consequently, in an MCMC algorithm, if the transition probability satisfies the detailed balance condition with respect to the target distribution, it ensures the reversibility of the process and ultimately the stationarity of the chain. Finally, imposing the conditions of irreducibility and aperiodicity, the stationary distribution of the chain converges to the target distribution. For proof of this theorem see Refs. [222, 267].

BIBLIOGRAPHY

- [1] Joshua S. Kaizer, A. Kevin Heller, and William L. Oberkampf. "Scientific computer simulation review." In: *Reliability Engineering & System Safety* 138 (2015), pp. 210–218. DOI: 10.1016/j.res.2015.01.020.
- [2] Keith Beven. *Environmental Modelling: An Uncertain Future*. London: Routledge, 2009.
- [3] T.G. Trucano, L.P. Swiler, T. Igusa, W.L. Oberkampf, and M. Pilch. "Calibration, validation, and sensitivity analysis: What's what." In: *Reliability Engineering & System Safety* 91.10-11 (2006), pp. 1331–1357. DOI: 10.1016/j.res.2005.11.031.
- [4] Anthony O'Hagan. "Bayesian analysis of computer code outputs: A tutorial." In: *Reliability Engineering and System Safety* 91 (10–11 2006), pp. 1290–1300. DOI: 10.1016/j.res.2005.11.025. URL: <http://dx.doi.org/10.1016/j.res.2005.11.025>.
- [5] International Atomic Energy Agency (IAEA). *Deterministic Safety Analysis for Nuclear Power Plants*. IAEA Safety Standard Series SSG-2. Vienna, 2009.
- [6] Francesco D'Auria. "Perspectives in System Thermal-Hydraulics." In: *Nuclear Engineering and Technology* 44.8 (2012), pp. 855–870.
- [7] United States Nuclear Regulatory Commission. *50.46 Acceptance criteria for emergency core cooling systems for light-water nuclear power reactors*. Title 10, Code of Federal Regulations. Nov. 1, 2017. URL: <https://www.nrc.gov/reading-rm/doc-collections/cfr/part050/part050-0046.html>.
- [8] D. Bestion. "System Code Models and Capabilities." In: *THICKET 2008. Seminar on the Transfer of Competence, Knowledge, and Experience Gained through CSNI Activities in the Field of Thermal-Hydraulics*. Pisa, 2008, pp. 81–106.
- [9] F. Barre and M. Bernard. "The CATHARE code strategy and assessment." In: *Nuclear Engineering and Design* 124.3 (1990), pp. 257–284. DOI: 10.1016/0029-5493(90)90296-a.
- [10] Ralph Nelson and Cetin Unal. "A phenomenological model of the thermal hydraulics of convective boiling during the quenching of hot rod bundles Part I: Thermal hydraulic model." In: *Nuclear Engineering and Design* 136.3 (1992), pp. 277–298. DOI: 10.1016/0029-5493(92)90029-u.
- [11] Wolfgang Wulff. "Simulation of two-phase in complex systems." In: *Nuclear Technology* 159 (2007), pp. 292–309.

- [12] International Atomic Energy Agency (IAEA). *Accident Analysis for Nuclear Power Plants*. Tech. rep. SRS No. 23. Vienna, 2002.
- [13] G.S. Lellouche et al. "Quantifying reactor safety margins part 4: Uncertainty evaluation of lbloca analysis based on trac-pf1/mod 1." In: *Nuclear Engineering and Design* 119.1 (1990), pp. 67–95. DOI: 10.1016/0029-5493(90)90074-8.
- [14] H. Glaeser, E. Hofer, M. Kloos, and T. Skorek. "Uncertainty and sensitivity analysis of a post-experiment calculation in thermal hydraulics." In: *Reliability Engineering & System Safety* 45.1-2 (1994), pp. 19–33. DOI: 10.1016/0951-8320(94)90073-6.
- [15] Graham B. Wallis. "Uncertainties and probabilities in nuclear reactor regulation." In: *Nuclear Engineering and Design* 237.15-17 (2007), pp. 1586–1592. DOI: 10.1016/j.nucengdes.2006.12.013.
- [16] Horst Glaeser. "GRS Method for Uncertainty and Sensitivity Evaluation of Code Results and Applications." In: *Science and Technology of Nuclear Installations* 2008 (2008), pp. 1–7. DOI: 10.1155/2008/798901.
- [17] Damar Wicaksono. *Summary of PSI Contribution in the OECD/NEA PREMIUM Benchmark*. Tech. rep. SB-RND-ACT-006-13.007. Paul Scherrer Institut, 2016.
- [18] M. Perez et al. "Uncertainty and sensitivity analysis of a LBLOCA in a PWR Nuclear Power Plant: Results of the Phase V of the BEMUSE programme." In: *Nuclear Engineering and Design* 241.10 (2011), pp. 4206–4222. DOI: 10.1016/j.nucengdes.2011.08.019.
- [19] A. Kovtonyuk, A. Petrucci, and Francesco D'Auria. *Post-BEMUSE Reflood Model Input Uncertainty Methods (PREMIUM) Benchmark Phase II: Identification of Influential Parameters*. Tech. rep. NEA/CSNI/R(2014)14. Nuclear Energy Agency (NEA) / Organisation for Economic Co-operation and Development (OECD), 2015.
- [20] Francesc Reventós, Elsa de Alfonso, and Rafael Mendizabál Sanz. *PREMIUM: A Benchmark on the Quantification of the Uncertainty of the Physical Models in System Thermal-Hydraulic Codes: Methodologies and Data Review*. Tech. rep. NEA/CSNI/R(2016)9. Nuclear Energy Agency (NEA) / Organisation for Economic Co-operation and Development (OECD), 2016.
- [21] Rafael Mendizabál Sanz, Elsa de Alfonso, Jordi Freixa, and Francesc Reventós. *Post-BEMUSE Reflood Model Input Uncertainty Methods (PREMIUM) Benchmark: Final Report*. Tech. rep. NEA/CSNI/R(2016)18. Nuclear Energy Agency (NEA) / Organisation for Economic Co-operation and Development (OECD), 2017.

- [22] A. Forge, R. Pochard, H. Glaeser, W. Hobbhan, E. Hofer, and V. Teschendorff. *Review study on uncertainty methods for thermal-hydraulics system codes*. Tech. rep. S. European Commission, 1994.
- [23] Damar Wicaksono. *Trip Report – WGAMA Task Group on the PREMIUM Benchmark: Phase V Meeting 11-13.05-2015, OECD/NEA, Issy-les-Moulineaux, France*. Tech. rep. SB-LRS-DOC-005-15. Paul Scherrer Institut, 2015.
- [24] Agnès de Crècy. “Determination of the Uncertainties of the Constitutive Relationships of the CATHARE 2 Code.” In: *International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering (M&C)*. Salt Lake City, 2001. Utah.
- [25] B.E. Boyack et al. “Quantifying reactor safety margins part 1: An overview of the code scaling, applicability, and uncertainty evaluation methodology.” In: *Nuclear Engineering and Design* 119.1 (1990), pp. 1–15. DOI: 10.1016/0029-5493(90)90071-5.
- [26] D. Bestion. “The physical closure laws in the CATHARE code.” In: *Nuclear Engineering and Design* 124.3 (1990), pp. 229–245. DOI: 10.1016/0029-5493(90)90294-8.
- [27] Alessandro Petruzzi and Francesco D’Auria. “Approaches, Relevant Topics, and Internal Method for Uncertainty Evaluation in Predictions of Thermal-Hydraulic System Codes.” In: *Science and Technology of Nuclear Installations* 2008 (2008), pp. 1–17. DOI: 10.1155/2008/325071.
- [28] United States Nuclear Regulatory Commission. *TRACE v5.0 Theory Manual*. 2012. Washington, DC.
- [29] Paul Scherrer Institut. *Steady-state and Transient Analysis Research for the Swiss Reactors (STARS Program)*. Oct. 29, 2017. URL: <https://www.psi.ch/stars/>.
- [30] Bertrand Iooss and Paul Lemaître. “A review on global sensitivity analysis methods.” In: *Uncertainty Management in Simulation-Optimization of Complex Systems*. Springer Nature, 2015, pp. 101–122. DOI: 10.1007/978-1-4899-7547-8_5.
- [31] H. Christopher Frey and Sumeet R. Patil. “Identification and Review of Sensitivity Analysis Methods.” In: *Risk Analysis* 22.3 (2002), pp. 553–578. DOI: 10.1111/0272-4332.00039.
- [32] Mihaela Ionescu-Bujor and Dan G. Cacuci. “A Comparative Review of Sensitivity and Uncertainty Analysis of Large-Scale Systems - I: Deterministic Methods.” In: *Nuclear Science and Engineering* 147.3 (2004), pp. 189–203. DOI: 10.13182/nse03-105cr.

- [33] Dan G. Cacuci and Mihaela Ionescu-Bujor. "A Comparative Review of Sensitivity and Uncertainty Analysis of Large-Scale Systems - II: Statistical Methods." In: *Nuclear Science and Engineering* 147.3 (2004), pp. 204–217. DOI: 10.13182/04-54cr.
- [34] Andrea Saltelli, Marco Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, and S. Tarantola. *Global Sensitivity Analysis, The Primer*. West Sussex: John Wiley & Sons, Inc., 2008. DOI: 10.1002/9780470725184.
- [35] Thomas J. Santner, Brian J. Williams, and William I. Notz. *The Design and Analysis of Computer Experiments*. New York: Springer, 2003. DOI: 10.1007/978-1-4757-3799-8.
- [36] Dan G. Cacuci. *Sensitivity and Uncertainty Analysis, Volume I: Theory*. Boca Raton: Chapman & Hall/CRC, 2003. Florida.
- [37] D. G. Cacuci and Mihaela Ionescu-Bujor. "Sensitivity and Uncertainty Analysis, Data Assimilation, and Predictive Best-Estimate Model Calibration." In: *Handbook of Nuclear Engineering*. Ed. by D. G. Cacuci. Springer, 2010, pp. 1913–2051. DOI: 10.1007/978-0-387-98149-9_17. New York.
- [38] Saman Razavi and Hoshin V. Gupta. "What do we mean by sensitivity analysis? The need for comprehensive characterization of "global" sensitivity in Earth and Environmental systems models." In: *Water Resources Research* 51.5 (2015), pp. 3070–3092. DOI: 10.1002/2014WR016527.
- [39] Andrea Saltelli, Stefano Tarantola, Francesca Campolongo, and Marco Ratto. *Sensitivity Analysis in Practice: A Guide to Assessing Scientific Models*. West Sussex: John Wiley & Sons, Inc., 2004.
- [40] Andrea Saltelli, Marco Ratto, Stefano Tarantola, and Francesca Campolongo. "Sensitivity analysis practices: Strategies for model-based inference." In: *Reliability Engineering & System Safety* 91 (2006), pp. 1109–1125. DOI: 10.1016/j.res.2005.11.014.
- [41] I.M Sobol'. "Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates." In: *Mathematics and Computers in Simulation* 55.1-3 (2001), pp. 271–280. DOI: 10.1016/S0378-4754(00)00270-6.
- [42] R.I Cukier, H.B Levine, and K.E Shuler. "Nonlinear Sensitivity Analysis of Multiparameter Model Systems." In: *Journal of Computational Physics* 26.1 (1978), pp. 1–42. DOI: 10.1016/0021-9991(78)90097-9.
- [43] Jon C Helton. "Uncertainty and sensitivity analysis techniques for use in performance assessment for radioactive waste disposal." In: *Reliability Engineering & System Safety* 42 (1993), pp. 327–367. DOI: 10.1016/0951-8320(93)90097-i.

- [44] Max D. Morris. "Factorial Sampling Plans for Preliminary Computational Experiments." In: *Technometrics* 33.2 (1991), pp. 161–174. DOI: 10.2307/1269043.
- [45] D. G. Cacuci and M. Ionescu-Bujor. "Adjoint Sensitivity Analysis of the RELAP5/MOD3.2 Two-Fluid Thermal-Hydraulic Code System - I: Theory." In: *Nuclear Science and Engineering* 136.1 (2000), pp. 59–84. DOI: 10.13182/nse136-59.
- [46] M. Ionescu-Bujor and D. G. Cacuci. "Adjoint Sensitivity Analysis of the RELAP5/MOD3.2 Two-Fluid Thermal-Hydraulic Code System - II: Applications." In: *Nuclear Science and Engineering* 136.1 (2000), pp. 85–121. DOI: 10.13182/nse136-85.
- [47] William T. Nutt and Graham B. Wallis. "Evaluation of nuclear safety from the outputs of computer codes in the presence of uncertainties." In: *Reliability Engineering & System Safety* 83.1 (2004), pp. 57–77. DOI: 10.1016/j.res.2003.08.008.
- [48] Katherine Campbell, Michael D. McKay, and Brian J. Williams. "Sensitivity Analysis when Model Outputs are Functions." In: *Reliability Engineering & System Safety* 91 (2006), pp. 1468–1472. DOI: 10.1016/j.res.2005.11.049.
- [49] J. O. Ramsay and B. W. Silverman. *Functional Data Analysis*. 2nd Edition. Springer, 2005. DOI: 10.1007/978-1-4757-7107-7. New York.
- [50] Mihaela Ionescu-Bujor, Xuezhou Jin, and Dan G. Cacuci. "Deterministic Local Sensitivity Analysis of Augmented Systems - II: Applications to the QUENCH-04 Experiment Using the RELAP5/MOD3.2 Code System." In: *Nuclear Science and Engineering* 151.1 (2005), pp. 67–81. DOI: 10.13182/nse151-67.
- [51] Benjamin Auder, Agnès De Crecy, Bertrand Iooss, and Michel Marquès. "Screening and metamodeling of computer experiments with functional outputs. Application to thermal-hydraulic computations." In: *Reliability Engineering & System Safety* 107 (2012), pp. 122–131. DOI: 10.1016/j.res.2011.10.017.
- [52] Andrej Prošek and Matjaž Leskovar. "Use of FFTBM by signal mirroring for sensitivity study." In: *Annals of Nuclear Energy* 76 (2015), pp. 253–262. DOI: 10.1016/j.anucene.2014.09.051.
- [53] Jerome Sacks, William J. Welch, Toby Mitchell, and Henry P. Wynn. "Design and Analysis of Computer Experiments." In: *Statistical Science* 4.4 (1989), pp. 409–423.
- [54] Kai-Tai Fang, Runze Li, and Agus Sudjianto. *Design and Modeling for Computer Experiments*. Chapman and Hall/CRC, 2006. DOI: 10.1201/9781420034899.

- [55] M. J. Asher, B. F. W. Croke, A. J. Jakeman, and L. J. M. Peeters. "A review of surrogate models and their application to ground-water modeling." In: *Water Resources Research* 51.8 (2015), pp. 5957–5973. DOI: 10.1002/2015wr016967.
- [56] T. W. Simpson, J. D. Peplinski, P. N. Koch, and J. K. Allen. "Metamodels for Computer-based Engineering Design: Survey and Recommendations." In: *Engineering with Computers* 17 (2001), pp. 129–150. DOI: 10.1007/pl00007198.
- [57] Saman Razavi, Bryan A. Tolson, and Donald H. Burn. "Review of surrogate modeling in water resources." In: *Water Resources Research* 48.7 (2012). DOI: 10.1029/2011wr011527.
- [58] T. W. Simpson, D. Lin, and W. Chen. "Sampling Strategies for Computer Experiments: Design and Analysis." In: *International Journal of Reliability and Applications* 2.3 (2001), pp. 209–240.
- [59] Jerome Sacks, Susannah B. Schiller, and William J. Welch. "Designs for Computer Experiments." In: *Technometrics* 31.1 (1989), pp. 41–47. DOI: 10.1080/00401706.1989.10488474.
- [60] Daniel G. Krige. "A statistical approach to some mine valuation and allied problems on the Witwatersrand." MA thesis. University of the Witwatersrand, 1951.
- [61] George Matheron. "Principles of Geostatistics." In: *Economic Geology* 58.8 (1963), pp. 1246–1266. DOI: 10.2113/gsecongeo.58.8.1246.
- [62] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. Cambridge, Massachusetts: The MIT Press, 2006. ISBN: 026218253X.
- [63] George E. P. Box and Norman R. Draper. *Response Surfaces, Mixtures, and Ridge Analyses*. 2nd. Wiley-Blackwell, 2007. DOI: 10.1002/0470072768.
- [64] J. D. Kerrigan and D. R. Coleman. "Application of the response surface method of uncertainty analysis to establish distributions of FRAP-S₃ calculated stored energy for PWR-type fuels." In: *Nuclear Engineering and Design* 54.2 (1979), pp. 211–224. DOI: 10.1016/0029-5493(79)90168-7.
- [65] A. C. Lucia. "Response surface methodology approach for structural reliability analysis: an outline of typical applications performed at CEC-JRC, ISPRA." In: *Nuclear Engineering and Design* 71 (1982), pp. 281–286. DOI: 10.1016/0029-5493(82)90092-9.
- [66] Lucia Faravelli. "Response-Surface Approach for Reliability Analysis." In: *Journal of Engineering Mechanics* 115.12 (1989), pp. 2763–2781. DOI: 10.1061/(asce)0733-9399(1989)115:12(2763).

- [67] Ronald E. Engel, John M. Sorensen, Randall S. May, Kenneth J. Doran, N. G. Trikouros, and Eugene S. Mozias. "Response surface development using RETRAN." In: *Nuclear Technology* 93 (1991), pp. 65–81.
- [68] Jack P. C. Kleijnen and Robert G. Sargent. "A methodology for fitting and validating metamodels in simulation." In: *European Journal of Operational Research* 120 (2000), pp. 14–29.
- [69] Timothy W. Simpson, Timothy M. Mauery, John J. Korte, and Farrokh Mistree. "Comparison Of Response Surface And Kriging Models For Multidisciplinary Design Optimization." In: *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*. American Institute of Aeronautics and Astronautics (AIAA), 1998. DOI: 10.2514/6.1998-4755.
- [70] D.J. Fonseca, D.O. Navaresse, and G.P. Moynihan. "Simulation metamodeling through artificial neural networks." In: *Engineering Applications of Artificial Intelligence* 16 (2003), pp. 177–183. DOI: 10.1016/S0952-1976(03)00043-5.
- [71] Bruno Sudret. "Global sensitivity analysis using polynomial chaos expansions." In: *Reliability Engineering & System Safety* 93 (2008), pp. 964–979. DOI: 10.1016/j.res.2007.04.002.
- [72] Bruno Sudret. "Meta-models for structural reliability and uncertainty quantification." In: *Proceedings of the 5th Asian-Pacific Symposium on Structural Reliability and its Applications (APSSRA 2012)*. 2012.
- [73] Elsevier B.V. *Scopus: the largest database of peer-reviewed literature*. May 5, 2017. URL: <https://www.scopus.com>.
- [74] Neil D. Cox. "Comparison of two uncertainty analysis methods." In: *Nuclear Science and Engineering* 64 (1977), pp. 258–265.
- [75] T. Ishigami, E. Cazzoli, M. Khatib-Rahbar, and S. D. Unwin. "Techniques to Quantify the Sensitivity of Deterministic Model Uncertainties." In: *Nuclear Science and Engineering* 101.4 (1989), pp. 371–383. DOI: 10.13182/nse89-a23625.
- [76] R. Meyder. "Modelling of Transient Fuel Rod Behavior and Core Damage during Loss of Coolant Accidents in a Light Water Reactor." In: *Nuclear Engineering and Design* 100.3 (1986), pp. 307–314. DOI: 10.1016/0029-5493(87)90082-3.
- [77] M. Khatib-Rahbar, E. Cazzoli, M. Lee, H. Nourbakhsh, R. Davis, and E. Schmidt. "A Probabilistic Approach to Quantifying Uncertainties in the Progression of Severe Accidents." In: *Nuclear Science and Engineering* 102.3 (1989), pp. 219–259. DOI: 10.13182/nse89-a27476.

- [78] Wasim Raza and Kwang-Yong Kim. "Shape optimization of wire-wrapped fuel assembly using Kriging metamodeling technique." In: *Nuclear Engineering and Design* 238.6 (2008), pp. 1332–1341. DOI: 10.1016/j.nucengdes.2007.10.018.
- [79] Kwang-Yong Kim and Jun-Woo Seo. "Numerical Optimization for the Design of a Spacer Grid with Mixing Vanes in a Pressurized Water Reactor Fuel Assembly." In: *Nuclear Technology* 149.1 (2005), pp. 62–70. DOI: 10.13182/nt05-a3579.
- [80] Xu Wu, Travis Mui, Guojun Hu, Hadi Meidani, and Tomasz Kozlowski. "Inverse uncertainty quantification of TRACE physical model parameters using sparse grid stochastic collocation surrogate model." In: *Nuclear Engineering and Design* 319 (2017), pp. 185–200. DOI: 10.1016/j.nucengdes.2017.05.011.
- [81] Dave Higdon, Ken Geelhood, Brian Williams, and Cetin Unal. "Calibration of tuning parameters in the FRAPCON model." In: *Annals of Nuclear Energy* 52 (2013), pp. 95–102. DOI: 10.1016/j.anucene.2012.06.018.
- [82] Katherine Campbell. "Statistical calibration of computer simulations." In: *Reliability Engineering & System Safety* 91.10-11 (2006), pp. 1358–1363. DOI: 10.1016/j.res.2005.11.032.
- [83] Marc C. Kennedy and Anthony O'Hagan. "Bayesian calibration of computer models." In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 63.3 (2001), pp. 425–464. DOI: 10.1111/1467-9868.00294.
- [84] Jari Kaipio and Colin Fox. "The Bayesian Framework for Inverse Problem in Heat Transfer." In: *Heat Transfer Engineering* 32.9 (2011), pp. 718–753. DOI: 10.1080/01457632.2011.525137.
- [85] Jim W. Hall, Lucy J. Manning, and Robin K. S. Hankin. "Bayesian calibration of a flood inundation model using spatial data." In: *Water Resources Research* 47.5 (2011). DOI: 10.1029/2009wr008541.
- [86] You Ling, Joshua Mullins, and Sankaran Mahadevan. "Selection of model discrepancy priors in Bayesian calibration." In: *Journal of Computational Physics* 276 (2014), pp. 665–680. DOI: 10.1016/j.jcp.2014.08.005.
- [87] Andrew Gelman, John B. Carlin, Hal S. Stern, David B. Dunson, and Aki Vehtari. *Bayesian Data Analysis*. Taylor & Francis Ltd, 2013. ISBN: 9781439840955.
- [88] Dave Higdon, James Gattiker, Brian Williams, and Maria Rightley. "Computer Model Calibration Using High-Dimensional Output." In: *Journal of the American Statistical Association* 103.482 (2008), pp. 570–583. DOI: 10.1198/016214507000000888.

- [89] M. J. Bayarri, J. O. Berger, J. Cafeo, G. Garcia-Donato, F. Liu, J. Palomo, R. J. Parthasarathy, R. Paulo, J. Sacks, and D. Walsh. "Computer model validation with functional output." In: *The Annals of Statistics* 35.5 (2007), pp. 1874–1906. DOI: 10.1214/009053607000000163.
- [90] R. D. Wilkinson. "Bayesian Calibration of Expensive Multivariate Computer Experiments." In: *Large-Scale Inverse Problems and Quantification of Uncertainty*. John Wiley & Sons, Ltd, 2010, pp. 195–215. DOI: 10.1002/9780470685853.ch10.
- [91] Paul D. Arendt, Daniel W. Apley, Wei Chen, David Lamb, and David Gorsich. "Improving Identifiability in Model Calibration Using Multiple Responses." In: *Journal of Mechanical Design* 134.10 (2012), p. 100909. DOI: 10.1115/1.4007573.
- [92] Paul D. Arendt, Daniel W. Apley, and Wei Chen. "Quantification of Model Uncertainty: Calibration, Model Discrepancy, and Identifiability." In: *Journal of Mechanical Design* 134.10 (2012), p. 100908. DOI: 10.1115/1.4007390.
- [93] Maria J. Bayarri, James O. Berger, Rui Paulo, Jerry Sacks, John A. Cafeo, James Cavendish, Chin-Hsu Lin, and Jian Tu. "A Framework for Validation of Computer Models." In: *Technometrics* 49.2 (2007), pp. 138–154.
- [94] Michael Goldstein. "Bayes Linear Analysis for Complex Physical Systems Modeled by Computer Simulators." In: *IFIP Advances in Information and Communication Technology*. 2012, pp. 78–94. DOI: 10.1007/978-3-642-32677-6_6.
- [95] Richard McElreath. *Statistical Rethinking: a Bayesian Course with Examples in R and Stan*. 1st Edition. Boca Raton: CRC Press, 2015. 487 pp. ISBN: 9781482253443. URL: http://www.ebook.de/de/product/24465987/richard_mcelreath_statistical_rethinking.html. Florida.
- [96] Andrew Gelman. "Prior Distribution." In: *Encyclopedia of Environmetrics*. Ed. by Abdel H. El-Shaarawi. Ed. by Walter W. Piegorsch. Vol. 3. Chichester: John Wiley & Sons, Ltd, 2002, pp. 1634–1637. DOI: 10.1002/9780470057339.vap039.
- [97] Andrew Gelman. "Prior distributions for variance parameters in hierarchical models (comment on article by Browne and Draper)." In: *Bayesian Analysis* 1.3 (2006), pp. 515–534. DOI: 10.1214/06-ba117a.
- [98] David J. C. Mackay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2005. Cambridge.
- [99] D. S. Sivia and J. Skilling. *Data Analysis: a Bayesian Tutorial*. 2nd. Oxford: Oxford University Press, 2012.

- [100] Nicholas Metropolis and S. Ulam. "The Monte Carlo Method." In: *Journal of the American Statistical Association* 44.247 (1949), pp. 335–341. DOI: 10.1080/01621459.1949.10483310.
- [101] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. "Equation of State Calculations by Fast Computing Machines." In: *The Journal of Chemical Physics* 21.6 (1953), pp. 1087–1092. DOI: 10.2172/4390578.
- [102] Wilfred K. Hastings. "Monte carlo sampling methods using Markov chains and their applications." In: *Biometrika* 57.1 (1970), pp. 97–109. DOI: 10.2307/2334940.
- [103] Stuart Geman and Donald Geman. "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images." In: *Journal of Applied Statistics* 20.5-6 (1984), pp. 25–62. DOI: 10.1080/02664769300000058.
- [104] Rupert Allison and Joanna Dunkley. "Comparison of sampling techniques for Bayesian parameter estimation." In: *Monthly Notices of the Royal Astronomical Society* 437.4 (2013), pp. 3918–3928. DOI: 10.1093/mnras/stt2190.
- [105] Charles J. Geyer. "Introduction to Markov Chain Monte Monte." In: *Handbook of Markov Chain Monte Carlo*. Ed. by Steve Brooks. Ed. by Andrew Gelman. Ed. by Galin L. Jones. Ed. by Xiao-Li Meng. Chapman and Hall/CRC, 2011. DOI: 10.1201/b10905-2.
- [106] Christophe Andrieu and Johannes Thoms. "A tutorial on adaptive MCMC." In: *Statistics and Computing* 18.4 (2008), pp. 343–373. DOI: 10.1007/s11222-008-9110-y.
- [107] Radford M. Neal. "MCMC Using Hamiltonian Dynamics." In: *Handbook of Markov Chain Monte Carlo*. Ed. by Steve Brooks. Ed. by Andrew Gelman. Ed. by Galin L. Jones. Ed. by Xiao-Li Meng. Chapman and Hall/CRC, 2011. DOI: 10.1201/b10905-6.
- [108] Jonathan Goodman and Jonathan Weare. "Ensemble samplers with affine invariance." In: *Communications in Applied Mathematics and Computational Science* 5.1 (2010), pp. 65–80. DOI: 10.2140/camcos.2010.5.65.
- [109] Daniel Foreman-Mackey, David W. Hogg, Dustin Lang, and Jonathan Goodman. "emcee: The MCMC Hammer." In: *Publications of the Astronomical Society of the Pacific* 125.925 (2013), pp. 306–312. DOI: 10.1086/670067.

- [110] C. Unal, B. Williams, F. Hemez, S.H. Atamturktur, and P. McClure. "Improved best estimate plus uncertainty methodology, including advanced validation concepts, to license evolving nuclear reactors." In: *Nuclear Engineering and Design* 241.5 (2011), pp. 1813–1833. DOI: 10.1016/j.nucengdes.2011.01.048.
- [111] Jordi Freixa, Elsa de Alfonso, and Francesc Reventós. "Testing methodologies for quantifying physical models uncertainties. A comparative exercise using CIRCE and IPREM (FFTBM)." In: *Nuclear Engineering and Design* 305 (2016), pp. 653–665. DOI: 10.1016/j.nucengdes.2016.05.037.
- [112] Damar Wicaksono, Omar Zerkak, and Andreas Pautz. "Bayesian Calibration of Thermal-Hydraulics Model with Time-Dependent Output." In: *Proceeding of the 11th International Topical Meeting on Nuclear Thermal-Hydraulics, Operation and Safety (NUTHOS-10)*. (Oct. 9, 2016–Oct. 13, 2014). Gyeongju, Korea, 2016.
- [113] Dong Li, Xiaojing Liu, and Yanhua Yang. "Investigation of uncertainty quantification method for BE models using MCMC approach and application to assessment with FEBA data." In: *Annals of Nuclear Energy* 107 (2017), pp. 62–70. DOI: 10.1016/j.anucene.2017.04.020.
- [114] Jaeseok Heo, Kyung Doo Kim, and Seung-Wook Lee. "Validation and uncertainty quantification for FEBA, FLECHT-SEASET, and PERICLES tests incorporating multi-scaling effects." In: *Annals of Nuclear Energy* 111 (2018), pp. 499–508. DOI: 10.1016/j.anucene.2017.08.033.
- [115] G. F. Hewitt and John G. Collier. *Introduction to Nuclear Power*. CRC Press, 2000. ISBN: 9781560324546.
- [116] Glenn A. Roth and Fatih Aydogan. "Theory and implementation of nuclear safety system codes – Part II: System code closure relations, validation, and limitations." In: *Progress in Nuclear Energy* 76 (2014), pp. 55–72. DOI: 10.1016/j.pnucene.2014.05.003.
- [117] Alessandro Petruzzi and Francesco D'Auria. "Thermal-Hydraulic System Codes in Nuclear Reactor Safety and Qualification Procedures." In: *Science and Technology of Nuclear Installations* 2008 (2008), pp. 1–16. DOI: 10.1155/2008/460795.
- [118] A. Petruzzi, M. Cherubini, and F. D'Auria. "Thirty Years' Experience in RELAP5 Applications at GRNSPG & NINE." In: *Nuclear Technology* 193.1 (2016), pp. 47–87. DOI: 10.13182/nt14-144.
- [119] S. Mostafa Ghiaasiaan. *Two-Phase Flow, Boiling and Condensation: In Conventional and Miniature Systems*. New York: Cambridge University Press, 2007. DOI: 10.1017/cbo9780511619410.

- [120] D. Bestion. "The difficult challenge of a two-phase CFD modelling for all flow regimes." In: *Nuclear Engineering and Design* 279 (2014), pp. 116–125. DOI: 10.1016/j.nucengdes.2014.04.006.
- [121] Mamoru Ishii and Takashi Hibiki. *Thermo-Fluid Dynamics of Two-Phase Flow*. New York: Springer, 2011. DOI: 10.1007/978-1-4419-7985-8.
- [122] Y. J. Zeng, C. P. Hale, Geoffrey F. Hewitt, and S. P. Walker. "Flow and Heat Transfer in Pressurized Water Reactor Reflood." In: *Multiphase Science and Technology* 22.4 (2010), pp. 279–370. DOI: 10.1615/multscientechn.v22.i4.10.
- [123] P. Ihle and K. Rust. *FEBA—Flooding Experiments with Blocked Arrays Evaluation Report*. Tech. rep. Kernforschungszentrum Karlsruhe, 1984. URL: <http://bibliothek.fzk.de/zb/kfkberichte/KFK3657.pdf> (visited on 02/02/2016).
- [124] Tomasz Skorek and Agnès de Crècy. "PREMIUM - Benchmark on the quantification of the uncertainty of the physical models in the system thermal-hydraulic codes." In: *OECD/CSNI Workshop on Best Estimate Methods and Uncertainty Evaluations - Workshop Proceedings*. Barcelone, 2013.
- [125] United States Nuclear Regulatory Commission. *TRACE Pressurized Water Reactor Modeling Guidance, Preliminary Draft Report*. 2012. Washington, DC.
- [126] Omar Zerkak and Damar Wicaksono. "Methodology used by PSI as Part of Contribution to PREMIUM." In: Francesc Reventós, Elsa de Alfonso, and Rafael Mendizabál Sanz. *PREMIUM: A Benchmark on the Quantification of the Uncertainty of the Physical Models in System Thermal-Hydraulic Codes: Methodologies and Data Review*. Nuclear Energy Agency (NEA) / Organisation for Economic Co-operation and Development (OECD), 2016.
- [127] D. J. Miller, F. B. Cheung, and S. M. Bajorek. "Investigation of grid-enhanced two-phase convective heat transfer in the dispersed flow film boiling regime." In: *Nuclear Engineering and Design* 265 (2013), pp. 35–44. DOI: 10.1016/j.nucengdes.2013.07.013.
- [128] S. C. Yao, L. E. Hochreiter, and W. J. Leech. "Heat-Transfer Augmentation in Rod Bundles Near Spacer Grid." In: *Journal of Heat Transfer* 104 (1982), pp. 76–81.
- [129] A. J. Wickett, J. C. Birchley, and B. J. Holmes. *Quantification of Large LOCA Uncertainties*. Tech. rep. United Kingdom Atomic Energy Authority (UK AEA), 1991.

- [130] H. Glaeser, B. Krzykacz-Hausmann, W. Luther, S. Schwarz, and T. Skorek. *Methodenentwicklung und Exemplarische Anwendungen zur Bestimmung der Aussagersicherheit von Rechenprogrammergebnissen*. Tech. rep. Gesellschaft für Anlagen- und Reaktorsicherheit (GRS), 2008.
- [131] Institut de protection et de sûreté nucléaire (IPSN). *Rapport d'activité 2000 de l'IPSN*. Tech. rep. 2001.
- [132] F. Campolongo, Andrea Saltelli, and J. Cariboni. "From Screening to Quantitative Analysis: a Unified Approach." In: *Computational Physics Communications* 182.4 (2011), pp. 978–988.
- [133] Andrea Saltelli. "Making best use of model evaluations to compute sensitivity indices." In: *Computer Physics Communications* 145.2 (2002), pp. 280–297. DOI: 10.1016/S0010-4655(02)00280-1.
- [134] Duncan Gillies. "B-splines." In: *Wiley Interdisciplinary Reviews: Computational Statistics* 2 (2010), pp. 237–242. DOI: 10.1002/wics.77.
- [135] Paul H. C. Eilers and Brian D. Marx. "Flexible smoothing with B-splines and penalties." In: *Statistical Science* 11.2 (1996), pp. 89–121. DOI: 10.1214/ss/1038425655.
- [136] Paul H. C. Eilers and Brian D. Marx. "Splines, knots, and penalties." In: *Wiley Interdisciplinary Reviews: Computational Statistics* 2 (2010), pp. 637–653. DOI: 10.1002/wics.125.
- [137] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria, 2017. URL: <https://www.R-project.org>.
- [138] Alois Kneip and Theo Gasser. "Statistical Tools to Analyze Data Representing a Sample of Curves." In: *The Annals of Statistics* 20.3 (1992), pp. 1266–1305. DOI: 10.1214/aos/1176348769.
- [139] Kongming Wang and Theo Gasser. "Alignment of curves by dynamic time warping." In: *The Annals of Statistics* 25.3 (1997), pp. 1251–1276. DOI: 10.1214/aos/1069362747.
- [140] J. O. Ramsay and Xiaochun Li. "Curve registration." In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 60.2 (1998), pp. 351–363. DOI: 10.1111/1467-9868.00129.
- [141] Ruye Wang. *Introduction to Orthogonal Transforms with Applications in Data Processing and Analysis*. Cambridge University Press, 2012, pp. 412–460. DOI: 10.1017/cbo9781139015158.

- [142] Damar Wicaksono, Omar Zerkak, and Andreas Pautz. "Exploring Variability in Reflood Simulation Results: an Application of Functional Data Analysis." In: *Proceeding of the 10th International Topical Meeting on Nuclear Thermal-Hydraulics, Operation and Safety (NUTHOS-10)*. (Dec. 14–18, 2014). Okinawa, Japan, 2014.
- [143] M.V. Ruano, J. Ribes, A. Seco, and J. Ferrer. "An improved sampling strategy based on trajectory design for application of the Morris method to systems with many input factors." In: *Environmental Modelling & Software* 37 (2012), pp. 103–109. DOI: 10.1016/j.envsoft.2012.03.008.
- [144] Genyuan Li, Carey Rosenthal, and Herschel Rabitz. "High Dimensional Model Representations." In: *The Journal of Physical Chemistry A* 105.33 (2001), pp. 7765–7777. DOI: 10.1021/jp010450t.
- [145] Toshimitsu Homma and Andrea Saltelli. "Importance measures in global sensitivity analysis of nonlinear models." In: *Reliability Engineering & System Safety* 52.1 (1996), pp. 1–17. DOI: 10.1016/0951-8320(96)00002-6.
- [146] Python Core Team. *Python: a Dynamic, open source programming language*. Python Software Foundation. 2017. URL: <https://www.python.org/>.
- [147] Francesca Campolongo, Jessica Cariboni, and Andrea Saltelli. "An effective screening design for sensitivity analysis of large models." In: *Environmental Modelling & Software* 22.10 (2007), pp. 1509–1518. DOI: 10.1016/j.envsoft.2006.10.004.
- [148] Alexandre Janon, Thierry Klein, Agnès Lagnoux, Maëlle Nodet, and Clémentine Prieur. "Asymptotic normality and efficiency of two Sobol index estimators." In: *ESAIM: Probability and Statistics* 18 (2014), pp. 342–364. DOI: 10.1051/ps/2013040.
- [149] Michiel J.W. Jansen. "Analysis of variance designs for model output." In: *Computer Physics Communications* 117.1-2 (1999), pp. 35–43. DOI: 10.1016/s0010-4655(98)00154-4.
- [150] Andrea Saltelli, Paola Annoni, Ivano Azzini, Francesca Campolongo, Marco Ratto, and Stefano Tarantola. "Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index." In: *Computer Physics Communications* 181.2 (2010), pp. 259–270. DOI: 10.1016/j.cpc.2009.09.018.
- [151] Andrea Saltelli and Paola Annoni. "How to avoid a perfunctory sensitivity analysis." In: *Environmental Modelling & Software* 25.12 (2010), pp. 1508–1517. DOI: 10.1016/j.envsoft.2010.04.012.

- [152] Stephen Joe and Frances Y. Kuo. "Constructing Sobol Sequences with Better Two-Dimensional Projections." In: *SIAM Journal on Scientific Computing* 30.5 (2008), pp. 2635–2654. DOI: 10.1137/070709359.
- [153] Damar Wicaksono, Omar Zerkak, and Andreas Pautz. "Sensitivity Analysis of a Bottom Reflood Simulation using the Morris Screening Method." In: *Proceeding of the 10th International Topical Meeting on Nuclear Thermal-Hydraulics, Operation and Safety (NUTHOS-10)*. (Dec. 14–18, 2014). Okinawa, Japan, 2014.
- [154] J. O. Ramsay, Hadley Wickham, Spencer Graves, and Giles Hooker. *fda: Functional Data Analysis*. R Package version 2.4.4. 2014. URL: <https://CRAN.R-project.org/package=fda> (visited on 02/02/2016).
- [155] B. Efron and R. Tibshirani. "Bootstrap Methods for Standard Errors, Confidence Intervals, and Other Measures of Statistical Accuracy." In: *Statistical Science* 1.1 (1986), pp. 54–77. DOI: 10.1214/ss/1177013815.
- [156] M. Andreani and G. Yadigaroglu. "Prediction Methods for Dispersed Flow Film Boiling." In: *International Journal of Multiphase Flow* 20.1 (1994), pp. 1–51. DOI: 10.1016/0301-9322(94)90069-8.
- [157] G. Yadigaroglu, R. A. Nelson, V. Teschendorff, Y. Murao, J. Kelly, and D. Bestion. "Modeling of Reflooding." In: *Nuclear Engineering and Design* 145 (1993), pp. 1–35.
- [158] W. Jaeger and V. H. Sánchez-Espinoza. "Uncertainty and sensitivity study in the frame of TRACE validation for reflood experiment." In: *Nuclear Technology* 184 (2013), pp. 333–350. DOI: 10.13182/nt184-333.
- [159] Bradley Jones and Rachel T. Johnson. "Design and analysis for the Gaussian process model." In: *Quality and Reliability Engineering International* 25 (2009), pp. 515–524. DOI: 10.1002/qre.1044.
- [160] Carla Currin, Toby Mitchell, Max Morris, and Don Ylvisaker. "Bayesian Prediction of Deterministic Functions, with Applications to the Design and Analysis of Computer Experiment." In: *Journal of the American Statistical Association* 86.416, 953–963 (1991).
- [161] E. H. Isaaks and R. M. Srivastava. "12. Ordinary Kriging." In: *An Introduction to Applied Geostatistics*. New York: Oxford University Press, 1989. Chap. 12, pp. 278–332.

- [162] S. Roberts, M. Osborne, M. Ebdon, S. Reece, N. Gibson, and S. Aigrain. "Gaussian processes for time-series modelling." In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 371.1984 (2012), pp. 20110550–20110550. DOI: 10.1098/rsta.2011.0550. URL: <http://dx.doi.org/10.1098/rsta.2011.0550>.
- [163] R. Syski. *Stochastic Process*. In: *Wiley StatsRef: Statistics Reference Online*. John Wiley & Sons, Inc., 2014. URL: <http://onlinelibrary.wiley.com/doi/10.1002/9781118445112.stat03025/abstract>.
- [164] Krzysztof Débicki. *Gaussian Process: Overview*. In: *Wiley StatsRef: Statistics Reference Online*. John Wiley & Sons, Inc., 2014. URL: <http://onlinelibrary.wiley.com/doi/10.1002/9781118445112.stat03025/abstract>.
- [165] François Bachoc. "Parametric Estimation of Covariance Function in Gaussian-Process Based Kriging Models. Application to Uncertainty Quantification for Computer Experiments." PhD thesis. Université Paris - Diderot, 2013.
- [166] Amandine Marrel, Bertrand Iooss, François Van Dorpe, and Elena Volkova. "An Efficient Methodology for Modeling Complex Computer Codes with Gaussian Processes." In: *Computational Statistics and Data Analysis* 52 (10 2008), pp. 4731–4744. DOI: 10.1016/j.csda.2008.03.026. URL: <http://dx.doi.org/10.1016/j.csda.2008.03.026>.
- [167] David Ginsbourger, Delphine Dupuy, Anca Badea, Laurent Carraro, and Olivier Roustant. "A note on the choice and the estimation of Kriging models for the analysis of deterministic computer experiments." In: *Applied Stochastic Models in Business and Industry* 25.2 (2009), pp. 115–131. DOI: 10.1002/asmb.741.
- [168] Robert B Gramacy and Herbert K. H Lee. "Bayesian Treed Gaussian Process Models With an Application to Computer Modeling." In: *Journal of the American Statistical Association* 103.483 (2008), pp. 1119–1130. DOI: 10.1198/016214508000000689.
- [169] P. Abrahamsen. *A review of Gaussian random fields and correlation functions*. Tech. rep. 917. Oslo: Norwegian Computing Center, 1997.
- [170] Olivier Roustant, David Ginsbourger, and Yves Deville. "DiceKriging, DiceOptim: Two R Packages for the Analysis of Computer Experiments by Kriging-based Metamodeling and Optimization." In: *Journal of Statistical Software* 51.1 (2012).
- [171] Marc C. Kennedy, Clive W. Anderson, Stefano Conti, and Anthony O'Hagan. "Case studies in Gaussian process modelling of computer codes." In: *Reliability Engineering & System Safety*

- 91.10-11 (2006), pp. 1301–1309. DOI: 10.1016/j.res.2005.11.028.
- [172] Michael L. Stein. “Comment on “Design and Analysis of Computer Experiments.”” In: *Statistical Science* 4.4 (1989), pp. 432–433.
- [173] Dave Higdon. “Space and Space-Time Modeling using Process Convolutions.” In: *Quantitative Methods for Current Environmental Issues*. Springer Nature, 2002, pp. 37–56. DOI: 10.1007/978-1-4471-0657-9_2.
- [174] François Bachoc, Guillaume Bois, Josselin Garnier, and Jean-Marc Martinez. “Calibration And Improved Prediction Of Computer Models By Universal Kriging.” In: *Nuclear Science and Engineering* 176.1 (2014). DOI: 10.13182/nse12-55. URL: <http://dx.doi.org/10.13182/NSE12-55>.
- [175] J.R. Koehler and A.B. Owen. “9. Computer experiments.” In: *Handbook of Statistics*. Elsevier BV, 1996, pp. 261–308. DOI: 10.1016/S0169-7161(96)13011-X.
- [176] Jay D. Martin and Timothy W. Simpson. “Use of Kriging Models to Approximate Deterministic Computer Models.” In: *AIAA Journal* 43.4 (2005), pp. 853–863. DOI: 10.2514/1.8650.
- [177] A. G. Journel and M. E. Rossi. “When Do We Need a Trend Model in Kriging?” In: *Mathematical Geology* 21.7, 715–739 (1989). DOI: 10.1007/bf00893318.
- [178] Hao Chen, Jason L. Loepky, Jerome Sacks, and William J. Welch. “Analysis Methods for Computer Experiments : How to Assess and What Counts?” In: *Statistical Science* 31.1 (2016), pp. 40–60.
- [179] David Ginsbourger, Rodolphe Le Riche, and Laurent Carraro. “Kriging Is Well-Suited to Parallelize Optimization.” In: *Computational Intelligence in Expensive Optimization Problems*. Springer Berlin Heidelberg, 2010, pp. 131–162. DOI: 10.1007/978-3-642-10701-6_6.
- [180] Ruichen Jin, Wei Chen, and Agus Sudjianto. “An Efficient Algorithm for Constructing Optimal Design of Computer Experiments.” In: *Proceedings of DETC '03 ASME 2003 Design Engineering Technical Conference and Computers and Information in Engineering Conference*. (Sept. 2–6, 2003). Chicago, IL, 2003.
- [181] G. Damblin, M. Couplet, and Bertrand Iooss. “Numerical Studies of Space-Filling Designs: Optimization of Latin Hypercube Samples and Subprojection Properties.” In: *Journal of Simulation* 7, 276–289 (2013).

- [182] Victoria C.P. Chen, Kwok-Leung Tsui, Russell R. Barton, and Martin Meckesheimer. "A review on design, modeling and applications of computer experiments." In: *IIE Transactions* 38.4 (2006), pp. 273–291. DOI: 10.1080/07408170500232495.
- [183] M. D. McKay, R. J. Beckman, and W. J. Conover. "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code." In: *Technometrics* 21.2 (1979), p. 239. DOI: 10.2307/1268522.
- [184] Felipe A. C. Viana. "A Tutorial on Latin Hypercube Design of Experiments." In: *Quality and Reliability Engineering International* 32 (2016), pp. 1975–1985. DOI: 10.1002/qre.1924.
- [185] Russel E. Caflisch. "Monte Carlo and quasi-Monte Carlo Methods." In: *Acta Numerica* 7 (1998). DOI: 10.1017/s0962492900002804.
- [186] Longjun Liu. "Could Enough Samples be more Important than Better Designs for Computer Experiments?" In: *Proceedings of the 38th Annual Simulation Symposium (ANSS'05)*. (Apr. 4–6, 2005). 2005. DOI: 10.1109/anss.2005.17.
- [187] Jason L. Loepky, Jerome Sacks, and William J. Welch. "Choosing the Sample Size of a Computer Experiment: A Practical Guide." In: *Technometrics* 51.4 (2009), pp. 366–376. DOI: 10.1198/tech.2009.08040. URL: <http://dx.doi.org/10.1198/TECH.2009.08040>.
- [188] Jack P. C. Kleijnen. "Kriging Metamodeling in Simulation: A Review." In: *SSRN Electronic Journal* 192.3 (2007), pp. 707–716. DOI: 10.2139/ssrn.980063.
- [189] K. Crombecq, E. Laermans, and T. Dhaene. "Efficient space-filling and non-collapsing sequential design strategies for simulation-based modeling." In: *European Journal of Operational Research* 214.3 (2011), pp. 683–696. DOI: 10.1016/j.ejor.2011.05.032.
- [190] F. Xiong, Y. Xiong, W. Chen, and S. Yang. "Optimizing Latin hypercube design for sequential sampling of computer experiments." In: *Engineering Optimization* 41.8 (2009), pp. 793–810. DOI: 10.1080/03052150902852999.
- [191] Tadeusz J. Ulrych, Mauricio D. Sacchi, and Alan Woodbury. "A Bayes tour of inversion: a tutorial." In: *Geophysics* 66.1 (2001), pp. 55–69. DOI: 10.1190/1.1444923.
- [192] Stephen R. Cole, Haitao Chu, and Sander Greenland. "Maximum Likelihood, Profile Likelihood, and Penalized Likelihood: A Primer." In: *American Journal of Epidemiology* 179.2 (2013), pp. 252–260. DOI: 10.1093/aje/kwt245.
- [193] Clemens Kreutz, Andreas Raue, Daniel Kaschek, and Jens Timmer. "Profile likelihood in systems biology." In: *the FEBS journal* 280 (2013), pp. 1564–2571. DOI: 10.1111/febs.12276.

- [194] Gerhard Venter. "Review of Optimization Techniques." In: *Encyclopedia of Aerospace Engineering*. John Wiley & Sons, 2010. DOI: 10.1002/9780470686652.eae495.
- [195] C. Helbert, D. Dupuy, and L. Carraro. "Assessment of uncertainty in computer experiments from Universal to Bayesian Kriging." In: *Applied Stochastic Models in Business and Industry* 25 (2009), pp. 99–113. DOI: 10.1002/asmb.743.
- [196] D. den Hertog, J. P. C. Kleijnen, and A. Y. D. Siem. "The correct Kriging variance estimated by bootstrapping." In: *Journal of the Operational Research Society* 57.4 (2006), pp. 400–409. DOI: 10.2139/ssrn.557862.
- [197] Martin Meckesheimer, Andrew J. Booker, Russell R. Barton, and Timothy W. Simpson. "Computationally Inexpensive Meta-model Assessment Strategies." In: *AIAA Journal* 40.10 (2002), pp. 2053–2060. DOI: 10.2514/2.1538.
- [198] Husam Hamad. "Validation of metamodels in simulation: a new metric." In: *Engineering with Computers* 27.4 (2011), pp. 309–317. DOI: 10.1007/s00366-010-0200-z. URL: <http://dx.doi.org/10.1007/s00366-010-0200-z>.
- [199] Bertrand Iooss, Loïc Boussouf, Vincent Feuillard, and Amandine Marrel. "Numerical studies of the metamodel fitting and validation processes." In: *International Journal of Advances in Systems and Measurements* 3 (2010) 11-21 (Jan. 7, 2010). arXiv: 1001.1049v2.
- [200] Peter G. Challenor. "Experimental design for the validation of kriging metamodels in computer experiments." In: *Journal of Simulations* 7 (2013), pp. 290–296. DOI: 10.1057/jos.2013.17. URL: <http://dx.doi.org/10.1057/jos.2013.17>.
- [201] M. Goulard and M. Voltz. "Linear coregionalization model: Tools for estimation and choice of cross-variogram matrix." In: *Mathematical Geology* 24.3 (1992), pp. 269–286. DOI: 10.1007/bf00893750.
- [202] Rui Paulo, Gonzalo García-Donato, and Jesús Palomo. "Calibration of computer models with multivariate output." In: *Computational Statistics & Data Analysis* 56.12 (2012), pp. 3959–3974. DOI: 10.1016/j.csda.2012.05.023.
- [203] I. T. Jolliffe. *Principal Component Analysis*. Springer-Verlag, 2002. DOI: 10.1007/b98835. New York.
- [204] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer New York, 2009. DOI: 10.1007/978-0-387-84858-7.

- [205] P. Reichert and N. Schuwirth. "Linking statistical bias description to multiobjective model calibration." In: *Water Resources Research* 48.9 (2012). DOI: 10.1029/2011wr011391.
- [206] David Huard and Alain Mailhot. "A Bayesian perspective on input uncertainty in model calibration: Application to hydrological model "abc"." In: *Water Resources Research* 42.7 (2006). DOI: 10.1029/2005wr004661.
- [207] Anthony O'Hagan. "Bayesian inference with misspecified models: Inference about what?" In: *Journal of Statistical Planning and Inference* 143.10 (2013), pp. 1643–1648. DOI: 10.1016/j.jspi.2013.05.016.
- [208] Jenný Brynjarsdóttir and Anthony O'Hagan. "Learning about physical parameters: the importance of model discrepancy." In: *Inverse Problems* 30.11 (2014), p. 114007. DOI: 10.1088/0266-5611/30/11/114007.
- [209] George B. Arhonditsis, Dimitra Papantou, Weitao Zhang, Gurbir Perhar, Evangelia Massos, and Molu Shi. "Bayesian calibration of mechanistic aquatic biogeochemical models and benefits for environmental management." In: *Journal of Marine Systems* 73.1-2 (2008), pp. 8–30. DOI: 10.1016/j.jmarsys.2007.07.004.
- [210] Dave Higdon, Marc Kennedy, James C. Cavendish, John A. Cafo, and Robert D. Ryne. "Combining Field Data and Computer Simulations for Calibration and Prediction." In: *SIAM Journal on Scientific Computing* 26.2 (2004), pp. 448–466. DOI: 10.1137/s1064827503426693.
- [211] Christopher K. Wikle, Ralph F. Milliff, Doug Nychka, and L. Mark Berliner. "Spatiotemporal Hierarchical Bayesian Modeling Tropical Ocean Surface Winds." In: *Journal of the American Statistical Association* 96.454 (2001), pp. 382–397. DOI: 10.1198/016214501753168109.
- [212] Christopher K. Wikle, L. Mark Berliner, and Noel Cressie. "Hierarchical Bayesian space-time models." In: *Environmental and Ecological Statistics* 5.2 (1998), pp. 117–154. DOI: 10.1023/a:1009662704779.
- [213] Kyle Siegrist. *Transformations of Random Variables*. 2017. URL: <http://www.randomservices.org/random/dist/Transformations.html>.
- [214] F. Liu, M.J. Bayarri, J.O. Berger, R. Paulo, and J. Sacks. "A Bayesian analysis of the thermal challenge problem." In: *Computer Methods in Applied Mechanics and Engineering* 197.29-32 (2008), pp. 2457–2466. DOI: 10.1016/j.cma.2007.05.032.

- [215] Luke Tierney. "Markov Chains for Exploring Posterior Distributions." In: *The Annals of Statistics* 22.4 (1994), pp. 1701–1728. DOI: 10.1214/aos/1176325750.
- [216] Kenneth Lange. "Generating Random Variates." In: *Numerical Analysis for Statisticians*. 2010. Chap. 22. DOI: 10.1007/978-1-4419-5945-4.
- [217] N. Balakrishnan. *Continuous Multivariate Distributions*. In: *Wiley StatsRef: Statistics Reference Online*. 2014. DOI: 10.1002/0471722065.
- [218] S. P. Brooks. "Quantitative convergence assessment for Markov chain Monte Carlo via cusums." In: *Statistics and Computing* 8.3 (1998), pp. 267–274. DOI: 10.1023/A:1008965613031.
- [219] A. Sokal. "Monte Carlo Methods in Statistical Mechanics: Foundations and New Algorithms." In: *Functional Integration*. Springer US, 1997, pp. 131–192. DOI: 10.1007/978-1-4899-0319-8_6.
- [220] John Stachurski. *Economic Dynamics: Theory and Computation*. Cambridge: The MIT Press, 2009. Massachusetts.
- [221] Thomas J. Sargent and John Stachurski. *Lectures in Quantitative Economics. Finite Markov Chain*. QuantEcon. 2017. URL: https://lectures.quantecon.org/py/finite_markov.html (visited on 09/27/2017).
- [222] Christian P. Robert and George Casella. *Monte Carlo Statistical Methods*. Springer New York, 2004. DOI: 10.1007/978-1-4757-4145-2.
- [223] Art B. Owen. *Monte Carlo theory, methods and examples*. 2013.
- [224] Christian P. Robert and George Casella. *Introducing Monte Carlo Methods with R*. Springer New York, 2010. DOI: 10.1007/978-1-4419-1576-4.
- [225] J. E. Gubernatis. "Marshall Rosenbluth and the Metropolis algorithm." In: *Physics of Plasmas* 12.5 (2005), p. 057303. DOI: 10.1063/1.1887186.
- [226] Siddhartha Chib and Edward Greenberg. "Understanding the Metropolis-Hastings Algorithm." In: *The American Statistician* 49.4 (1995), pp. 327–335. DOI: 10.1080/00031305.1995.10476177.
- [227] Heikki Haario, Eero Saksman, and Johanna Tamminen. "An Adaptive Metropolis Algorithm." In: *Bernoulli* 7.2 (2001), p. 223. DOI: 10.2307/3318737.
- [228] Heikki Haario, Marko Laine, Antonietta Mira, and Eero Saksman. "DRAM: Efficient adaptive MCMC." In: *Statistics and Computing* 16.4 (2006), pp. 339–354. DOI: 10.1007/s11222-006-9438-0.

- [229] Jim E. Griffin and Stephen G. Walker. “On adaptive Metropolis–Hastings methods.” In: *Statistics and Computing* 23.1 (2011), pp. 123–134. DOI: 10.1007/s11222-011-9296-2.
- [230] David Huijser, Jesse Goodman, and Brendon J. Brewer. “Properties of the Affine Invariant Ensemble Sampler in high dimensions.” In: (Sept. 7, 2015). arXiv: 1509.02230v2 [stat.CO].
- [231] Fengji Hou, Jonathan Goodman, David W. Hogg, Jonathan Weare, and Christian Schwab. “An Affine-Invariant Sampler for Exoplanet Fitting and Discovery in Radial Velocity Data.” In: *The Astrophysical Journal* 745.2 (2012), p. 198. DOI: 10.1088/0004-637x/745/2/198.
- [232] Joël Akeret, Sebastian Seehars, Adam Amara, Alexandre Refregier, and André Csillaghy. “CosmoHammer: Cosmological parameter estimation with the MCMC Hammer.” In: *Astronomy and Computing* 2 (2013), pp. 27–39. DOI: 10.1016/j.ascom.2013.06.003.
- [233] Bob Carpenter. *Ensemble Methods are Doomed to Fail in High Dimensions*. 2017. URL: <http://andrewgelman.com/2017/03/15/ensemble-methods-doomed-fail-high-dimensions/>.
- [234] Jonathan Goodman. *ACOR*. <https://www.math.nyu.edu/faculty/goodman/software/acor/index.html>. 2009.
- [235] Dan Foreman-Mackey. *ACOR*. <https://github.com/dfm/acor>. 2014.
- [236] Charles J. Geyer. “Practical Markov Chain Monte Carlo.” In: *Statistical Science* 7.4 (1992), pp. 473–483. DOI: 10.1214/ss/1177011137.
- [237] William A. Link and Mitchell J. Eaton. “On thinning of chains in MCMC.” In: *Methods in Ecology and Evolution* 3.1 (2011), pp. 112–115. DOI: 10.1111/j.2041-210x.2011.00131.x.
- [238] F. Liu, M. J. Bayarri, and J. O. Berger. “Modularization in Bayesian analysis, with emphasis on analysis of computer models.” In: *Bayesian Analysis* 4.1 (2009), pp. 119–150. DOI: 10.1214/09-ba404.
- [239] Adam Mantz. *rgw: Goodman-Weare Affine-Invariant Sampling*. R package version 0.1.0. 2016. URL: <https://CRAN.R-project.org/package=rgw>.
- [240] David J. Lunn, Andrew Thomas, Nicky Best, and David Spiegelhalter. “WinBUGS — a Bayesian modelling framework: concepts, structure, and extensibility.” In: *Statistics and Computing* 10.4 (2000), pp. 325–337. DOI: 10.1023/a:1008929526011.
- [241] Martyn Plummer. *JAGS: A program for analysis of Bayesian graphical models using Gibbs sampling*. 2003. URL: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.13.3406>.

- [242] Bob Carpenter, Andrew Gelman, Matthew D. Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. "Stan: A Probabilistic Programming Language." In: *Journal of Statistical Software* 76.1 (2017). DOI: 10.18637/jss.v076.i01.
- [243] Ramesh Rebba and Sankaran Mahadevan. "Validation of models with multivariate output." In: *Reliability Engineering & System Safety* 91.8 (2006), pp. 861–871. DOI: 10.1016/j.res.s.2005.09.004.
- [244] Ramesh Rebba, Shuping Huang, Yongming Liu, and Sankaran Mahadevan. "Statistical validation of simulation models." In: *International Journal of Materials and Product Technology* 25.1/2/3 (2006), p. 164. DOI: 10.1504/ijmpt.2006.008280.
- [245] Xiaomo Jiang and Sankaran Mahadevan. "Bayesian validation assessment of multivariate computational models." In: *Journal of Applied Statistics* 35.1 (2008), pp. 49–65. DOI: 10.1080/02664760701683577.
- [246] Jean Baccou and Eric Chojnacki. "A practical methodology for information fusion in presence of uncertainty: application to the analysis of a nuclear benchmark." In: *Environment Systems and Decisions* 34.2 (2014), pp. 237–248. DOI: 10.1007/s10669-014-9496-3.
- [247] Daniel Foreman-Mackey. "corner.py: Scatterplot matrices in Python." In: *The Journal of Open Source Software* 24 (2016). DOI: 10.21105/joss.00024. URL: <http://dx.doi.org/10.5281/zenodo.45906>.
- [248] Dan Carr, ported by Nicholas Lewin-Koh, Martin Maechler, and contains copies of lattice functions written by Deepayan Sarkar. *hexbin: Hexagonal Binning Routines*. R package version 1.27.1. 2015. URL: <https://CRAN.R-project.org/package=hexbin>.
- [249] Xu Wu, Tomasz Kozłowski, and Hadi Meidani. "Kriging-based inverse uncertainty quantification of nuclear fuel performance code BISON fission gas release model using time series measurement data." In: *Reliability Engineering & System Safety* 169 (2018), pp. 422–436. DOI: 10.1016/j.res.s.2017.09.029.
- [250] Keegan E. Hines, Thomas R. Middelndorf, and Richard W. Aldrich. "Determination of parameter identifiability in nonlinear biophysical models: A Bayesian approach." In: *The Journal of General Physiology* 143.3 (2014), pp. 401–416. DOI: 10.1085/jgp.201311116.

- [251] Roland Brun, Martin Kühni, Hansruedi Siegrist, Willi Gujer, and Peter Reichert. "Practical identifiability of ASM2d parameters—systematic selection and tuning of parameter subsets." In: *Water Research* 36.16 (2002), pp. 4113–4127. DOI: 10.1016/S0043-1354(02)00104-5.
- [252] Ryan N. Gutenkunst, Joshua J. Waterfall, Fergal P. Casey, Kevin S. Brown, Christopher R. Myers, and James P. Sethna. "Universally Sloppy Parameter Sensitivities in Systems Biology Models." In: *PLoS Computational Biology* 3.10 (2007), e189. DOI: 10.1371/journal.pcbi.0030189.
- [253] Damar Wicaksono, Omar Zerkak, and Andreas Pautz. "Global Sensitivity Analysis of Transient Code Output Applied to a Reflood Experiment Model Using the TRACE Code." In: *Nuclear Science and Engineering* 184.3 (2016), pp. 400–429. DOI: 10.13182/nse16-37.
- [254] Stefano Marelli and Bruno Sudret. "UQLab: A Framework for Uncertainty Quantification in Matlab." In: *Vulnerability, Uncertainty, and Risk*. American Society of Civil Engineers, 2014. DOI: 10.1061/9780784413609.257.
- [255] Brian M. Adams et al. *Dakota , A Multilevel Parallel Object-Oriented Framework for Design Optimization , Parameter Estimation , Uncertainty Quantification , and Sensitivity Analysis : Version 6 .7 User's Manual*. SAND2014-4633. Sandia National Laboratory. 2017.
- [256] Michaël Baudin, Anne Dutfoy, Bertrand Iooss, and Anne-Laure Popelin. "OpenTURNS: An Industrial Software for Uncertainty Quantification in Simulation." In: *Handbook of Uncertainty Quantification*. Springer International Publishing, 2017, pp. 2001–2038. DOI: 10.1007/978-3-319-12385-1_64.
- [257] Fabrice Gaudier. "URANIE: The CEA/DEN Uncertainty and Sensitivity platform." In: *Procedia - Social and Behavioral Sciences* 2.6 (2010), pp. 7660–7661. DOI: 10.1016/j.sbspro.2010.05.166.
- [258] Daoqiang Zhang and Zhi-Hua Zhou. "Two-directional two-dimensional PCA for efficient face representation and recognition." In: *Neurocomputing* 69.1-3 (2005), pp. 224–231. DOI: 10.1016/j.neucom.2005.06.004.
- [259] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. "A Global Geometric Framework for Nonlinear Dimensionality Reduction." In: *Science* 290.5500 (2000), pp. 2319–2323. DOI: 10.1126/science.290.5500.2319.
- [260] Sam T. Roweis and Lawrence K. Saul. "Nonlinear Dimensionality Reduction by Locally Linear Embedding." In: *Science* 290.5500 (2000), pp. 2323–2326. DOI: 10.1126/science.290.5500.2323.

- [261] A. Graps. "An introduction to wavelets." In: *IEEE Computational Science and Engineering* 2.2 (1995), pp. 50–61. DOI: 10.1109/99.388960.
- [262] Sander Greenland and David Draper. *Exchangeability*. In: *Encyclopedia of Biostatistics*. 2nd. John Wiley & Sons, Inc., 2005. DOI: 10.1002/0470011815.b2a15037. URL: <http://onlinelibrary.wiley.com/book/10.1002/0470011815>.
- [263] D. R. Jensen. *Multivariate Normal Distribution: Overview*. In: *Wiley StatsRef: Statistics Reference Online*. 2014. DOI: 10.1002/9781118445112.stat05654. URL: <http://onlinelibrary.wiley.com/doi/10.1002/9781118445112.stat05654/full>.
- [264] J. Tiago De Oliveira. *Gumbel Distribution*. In: *Wiley StatsRef: Statistics Reference Online*. 2014.
- [265] James O. Ramsay, Giles Hooker, and Spencer Graves. *Functional Data Analysis with R and MATLAB*. Springer, 2009. DOI: 10.1007/978-0-387-98185-7. New York.
- [266] Limin Wang. "Karhunen-Loeve Expansions and their Applications." PhD thesis. The London School of Economics and Political Science, 2008.
- [267] Frank P. Kelly. *Reversibility and Stochastic Networks*. New York: Cambridge University Press, 1978. ISBN: 1107401151.

ACRONYMS AND ABBREVIATIONS

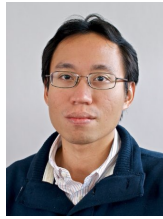
AIES	affine-invariant ensemble sampler.
BEMUSE	Best-Estimate Methods – Uncertainty and Sensitivity Evaluation.
BWR	boiling water reactor.
CHF	critical heat flux.
CI	confidence interval.
DFFB	dispersed flow film boiling.
FDA	functional data analysis.
FEBA	Flooding Experiments with Blocked Arrays.
FFTBM	Fast Fourier Transform-Based method.
FPC	functional principal component.
FPCA	functional principal component analysis.
GP	Gaussian process.
GSA	global sensitivity analysis.
HPDI	highest posterior density interval.
HT	heat transfer.
HTC	heat transfer coefficient.
IAFB	inverted annular film boiling.
IID	independent and identically distributed.
LBLOCA	large break loss-of-coolant accident.
LHS	latin hypercube sampling.
LMC	linear model of core regionalization.
LOCA	loss-of-coolant accident.
LRs	Laboratory for Reactor Physics and Systems Behavior.
LWR	light water reactor.

MC	Monte Carlo.
MCMC	Markov Chain Monte Carlo.
MH	Metropolis-Hastings.
ML	Maximum Likelihood.
MLE	Maximum Likelihood Estimation.
MVN	Multivariate Normal.
NEA	Nuclear Energy Agency.
NPP	nuclear power plant.
OAT	one-at-a-time.
OECD	Organization for Economic Cooperation and Development.
PC	principal component.
PCA	principal component analysis.
PCE	polynomial chaos expansion.
PCT	peak clad temperature.
PDF	probability density function.
POST-CHF	post-Critical-Heat-Flux.
PREMIUM	Post-BEMUSE Reflood Models Input Uncertainty Methods.
PSI	Paul Scherrer Institut.
PWR	pressurized water reactor.
QOI	quantity of interest.
RMSE	root-mean-square-error.
RPV	reactor pressure vessel.
SA	sensitivity analysis.
SE	standard error.
SETF	separate effect test facility.
SRS	simple random sampling.
STARS	Steady-state and Transient Analysis Research for Swiss Reactors.
SVD	singular value decomposition.

TH	thermal-hydraulics.
TRACE	TRAC/RELAP Computational Engine.
UQ	uncertainty quantification.
USNRC	the United States Nuclear Regulatory Commission.
v&v	verification and validation.
WGAMA	Working Group on the Analysis and Management of Accidents.

DAMAR CANGGIH WICAKSONO

PERSONAL INFORMATION



Born in Jakarta, 15 May 1986

damar.wicaksono@gmail.com

<https://github.com/damar-wicaksono>

(M) +41 (0) 78 798 5785

EDUCATION

<i>Doctor of Science</i>	2013-2018	EPF Lausanne, Switzerland
		<i>Nuclear Engineering</i> Thesis: <i>Bayesian Uncertainty Quantification of Physical Models in Thermal-Hydraulics System Codes</i> Advisors: Prof. Andreas PAUTZ, Mr. Omar ZERKAK, & Dr. Gregory PERRET
<i>Master of Science</i>	2010-2012	EPF Lausanne – ETH Zürich, Switzerland
		<i>Nuclear Engineering</i> · GPA: 5.52/6.00 Thesis: <i>Development and Assessment of an Improved Temporal Coupling for TRACE/S3K Analysis</i> Advisors: Prof. Rakesh CHAWLA & Mr. Omar ZERKAK
<i>Bachelor of Engineering</i>	2004-2009	Universitas Gadjah Mada, Indonesia
		<i>Nuclear Engineering</i> · GPA: 3.92/4.00 Thesis: <i>Multiobjective Optimization of PWR Fuel Loading Pattern using Simulated Annealing Algorithm</i> Advisor: Dr. Alexander Agung

WORK EXPERIENCE

<i>Doctoral Assistant</i>	2013-2018	Paul Scherrer Institut / EPF Lausanne
		<ul style="list-style-type: none">• Developed and validated novel methodology for inverse uncertainty quantification of nuclear safety analysis code using Bayesian statistics and techniques.• Applied Gaussian process regression technique for metamodeling a computationally expensive simulation code.• Gain skills in Python and R programming.• Project embedded within the STARS program, a Swiss technical safety organization supporting the Swiss Federal Nuclear Safety Inspectorate (ENSI).• Frequent technical reporting in an independent working environment.
<i>Intern</i>	Aug-Nov 2011	Paul Scherrer Institut
		Tested, analyzed, and validated different Monte Carlo-based simulation codes for in-core nuclear fuel utilization.
<i>Intern</i>	Jul-Oct 2011	Kernkraftwerk Leibstadt AG
		Industrial internship in the Safety Analysis Group, developing computer model of nuclear power plant for deterministic safety analysis purpose. Gained experience in writing technical report.

PUBLICATIONS AND CONFERENCE CONTRIBUTIONS

D. Wicaksono, O. Zerkak, and A. Pautz, "Global Sensitivity Analysis of Transient Code Output applied to a Reflood Experiment Model using TRACE Code," *Nuclear Science and Engineering*, vol. 184, no. 6, 2016.

D. Wicaksono, O. Zerkak, and A. Pautz, "Bayesian Calibration of Thermal-Hydraulics Model with Time-Dependent Output," in the *11th International Topical Meeting on Nuclear Thermal-Hydraulics, Operation and Safety (NUTHOS-11)*, Gyeongju, South Korea, Oct. 9–13, 2016.

D. Wicaksono, O. Zerkak, and A. Pautz, "A Methodology for Global Sensitivity Analysis of Transient Code Output applied to a Reflood Experiment Model using TRACE," in the *16th International Topical Meeting on Nuclear Reactor Thermal-Hydraulics*, Chicago, Illinois, Aug. 30 – Sept. 4, 2015.

D. Wicaksono, O. Zerkak, and A. Pautz, "Sensitivity Analysis of a Bottom Reflood Simulation using the Morris Screening Method," in the *10th International Topical Meeting on Nuclear Thermal-Hydraulics, Operation and Safety (NUTHOS-10)*, Okinawa, Japan, Dec. 14 – 18, 2014.

D. Wicaksono, O. Zerkak, and A. Pautz, "Exploring Variability in Reflood Simulation Results: an Application of Functional Data Analysis," in the *10th International Topical Meeting on Nuclear Thermal-Hydraulics, Operation and Safety (NUTHOS-10)*, Okinawa, Japan, Dec. 14 – 18, 2014.

COMPUTER SKILLS

<i>Basic</i>	C++, Adobe Illustrator
<i>Intermediate</i>	Shell scripting, Matlab, FORTRAN77/90, L ^A T _E X, Microsoft Office
<i>Advanced</i>	PYTHON, R

AWARDS AND ACCOLADES

2015 · Best Student Paper · NURETH-16, American Nuclear Society
2014 · Best Student Paper · NUTHOS-11, Japanese Nuclear Society
2014 · Best 1st Year Graduate Student · NES PhD Day, PSI
2010-12 · Excellence Scholarship · Federal Commission for Scholarship, CH
2009 · Cum Laude Graduate · Universitas Gadjah Mada, Indonesia

<i>Languages</i>	INDONESIAN · Mother tongue
	ENGLISH · Professional fluency
	FRENCH · Intermediate (B1)
	GERMAN · Basic (A1.2)

February 7, 2018

COLOPHON

This document was typeset using the typographical look-and-feel `classicthesis` developed by André Miede. The style was inspired by Robert Bringhurst's seminal book on typography "*The Elements of Typographic Style*". `classicthesis` is available for both \LaTeX and \LyX :

<https://bitbucket.org/amiede/classicthesis/>

Happy users of `classicthesis` usually send a real postcard to the author, a collection of postcards received so far is featured here:

<http://postcards.miede.de/>

Final Version as of February 8, 2018 (`classicthesis` version 4.2).

