Improved Methodology for the Determination of Model Uncertainties using the Example of ATHLET

GRS - 785

Gesellschaft für Anlagen-
und Reaktorsicherheit
(GRS) gGmbH

Improved Methodology for the Determination of Model Uncertainties using the Example of ATHLET

Verbesserte Methodik für die Bestimmung von Modellunsicherheiten am Beispiel von ATHLET

Thorsten Hollands
Livia Tiborcz
Tomasz Skorek
Markus Junk
Andreas Wielenberg

August 2024

Remark:

This report refers to the in-house research project RS1597 which has been funded by the German Federal Ministry for the Environment, Nature Conservation, Nuclear Safety and Consumer Protection (BMUV).

The work was conducted by the Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) gGmbH.

The authors are responsible for the content of the report.

GRS - 785
ISBN 978-3-910548-78-7

Keywords
ATHLET, Inverse Uncertainty Quantification, Methodology Development, Validation

Abstract

In the framework of the project RS1597 several aspects of carrying out an inverse uncertainty analysis have been addressed. As the question of parameter choice of interest (parameters related to the reflooding phenomena) has been set at the beginning of the project, the first step of the SAPIUM guidelines have been already fulfilled. As this phenomenon served as the basis for several international projects previously, the set of experiments have been already tested and have been deemed adequate and appropriate for the quantification process.

The approach chosen and implemented for our purposes is the so-called ABC (Approximate Bayesian Computation) method. This method circumvents the derivation of the likelihood function by introducing a rejection scheme based on an appropriately chosen distance metric comparing the simulated values with the observational data, or their properties (mean, skewness, etc.). Several implementation approaches have been looked into, and in the end, Python has been selected due to its flexibility, adaptability, extensive list of available libraries, and for its available easy connection to other GRS tools (MCDET, SUSA) addressing aspects of (forward) uncertainty quantification. The method had been implemented and tested on small examples before moving to adapt it to ATHLET simulations. The last steps of the approach (verification and validation) involve a forward uncertainty analysis applying the derived distributions of the uncertain input parameters.

For the validation of AC²/ATHLET using SET and CET, in the frame of the current project reflooding tests of the test series FEBA, FLECHT and PERICLES were investigated.

Additionally, the ATHLET validation manual has been extended with additional guidance on the application of Wilks' formula, the determination of sample sizes for uncertainty analysis and the treatment of code crashes, first for ATHLET 3.3 and then extended again for ATHLET 3.4. In the frame of international activities GRS participates mainly in two projects resp. networks, OECD/NEA ATRIUM and FONESYS.

Based on the current status of modelling for the inverse uncertainty quantification used for AC²/ATHLET applications, some topics for improvement could be identified: investigating different ABC metrics, expanding the experimental data base, saving chain-information mid analysis, including modelling bias, better resource allocation, the question of the likelihood function as well as surrogate modelling.

Kurzfassung

Das vorliegende Projekt UMRS1597 hatte das Ziel der Entwicklung einer Methodik zur inversen Unsicherheitsquantifizierung für AC²/ATHLET. Neben der Entwicklung der Methodik wurden geeignete Separate Effect Tests (SET) und Combined Effect Tests (CET) für diese Aufgabe analysiert. Das Projekt umfasste auch übergreifende Themen wie die Aktualisierung der ATHLET-Dokumentation für die GRS-Methode sowie die Beteiligung an internationalen Aktivitäten zur (inversen) Unsicherheitsquantifizierung.

Im Rahmen des Projekts wurden mehrere Aspekte der Durchführung einer inversen Unsicherheitsanalyse behandelt. Im OECD/NEA-Vorhaben SAPIUM wurden unter Beteiligung der GRS-Empfehlungen für ein systematisches Vorgehen bei der Bestimmung von Modellunsicherheiten aufgestellt. Diese wurden im vorliegenden Vorhaben bei der Wahl geeigneter Parameter (Parameter im Zusammenhang mit dem Phänomen des Wiederflutens) berücksichtigt. Da dieses Phänomen bereits als Grundlage für mehrere internationale Projekte diente, sind eine Reihe von Experimenten verfügbar, die als angemessen und geeignet für den Quantifizierungsprozess erachtet wurden. Eine geeignete Auswahl dieser Experimente wurde im Wesentlichen während der Vorbereitungsphase festgelegt, um unsere Ressourcen auf den praktischen statistischen Schritt der inversen Unsicherheitsquantifizierung konzentrieren zu können: Ableitung der Wahrscheinlichkeitsverteilungsfunktion (in geschlossener oder numerischer/empirischer Form) für mehrere unsichere Eingangsparameter auf der Grundlage von Integral- und Combined Effect Tests. Zu diesem Zweck wurden verschiedene Datenanalysemethoden im Rahmen der Bayes'schen Methode untersucht. Es ist schwierig, eine Likelihood-Funktion in geschlossener Form zu bestimmen, weil dies entweder mathematisch anspruchsvoll bis unmöglich ist oder man mehrere Einschränkungen bzw. starke Annahmen über das Systemverhalten in Abhängigkeit der unsicheren Parameter einführen muss. Dies gilt sowohl mathematisch als auch auf der Grundlage physikalischer/Modellierungsinformationen, die ein Expertenurteil erfordern würden. Daher wurde beschlossen, zu einer so genannten Likelihood-freien Methode überzugehen. Bei diesen Methoden wird die Likelihood-Funktion weiterhin für die Inferenz verwendet, aber anstatt sie auf irgendeine Weise (analytisch/numerisch) zu bestimmen, wird sie angenähert. Der für unsere Zwecke gewählte und implementierte Ansatz ist die sogenannte ABC-Methode (Approximate Bayesian Computation). Bei dieser Methode wird die Ableitung der Likelihood-Funktion umgangen, indem ein Ablehnungsschema eingeführt wird, das auf einer angemessen gewählten Abstandsmetrik basiert, die die simulierten Werte mit den

Beobachtungsdaten oder deren Eigenschaften (Mittelwert, Abweichung usw.) vergleicht. Das Ziel darin bestand, eine Methode zu entwickeln, die sich leicht anwenden und auf andere Probleme übertragen lässt, so dass Expertenurteile so weit wie möglich vermieden werden sollten. Dies führte nicht nur zur Anwendung der ABC-Methode, sondern auch zu der Entscheidung, keine Metamodelle, sondern den eigentlichen Code (ATHLET) selbst zu verwenden. Dies bedeutet nicht, dass die abgeleitete Methode in Zukunft nicht auf Metamodelle angewandt werden könnte, um Rechenzeit zu sparen, sondern stellt sicher, dass wir in unserem Ansatz keine Annahmen treffen, die nur für das gegebene Problem gelten und nur schwer bis unmöglich auf ein anderes Problem übertragen werden können.

Es wurden mehrere Implementierungsansätze geprüft, und schließlich wurde Python aufgrund seiner Flexibilität, Anpassungsfähigkeit, der umfangreichen verfügbaren Bibliotheken und der möglichen einfachen Verbindung zu anderen GRS-Tools (MCDET, SUSA) ausgewählt, um die Aspekte der (Vorwärts-)Unsicherheitsquantifizierung zu behandeln. Die Methode wurde implementiert und an kleinen Beispielen getestet, bevor sie an die ATHLET-Simulationen angepasst wurde. Hier wurde ein schrittweiser Ansatz verfolgt, bei dem die Beobachtungsmatrix und die entsprechende Simulation schrittweise erweitert wurden, um jedes auftretende Problem rechtzeitig angehen zu können. Leider dauert die eigentliche Simulationsphase aufgrund von Schwierigkeiten bei der Verfügbarkeit von Ressourcen länger als erwartet.

Die letzten Schritte des Ansatzes (Verifizierung und Validierung) beinhalten eine Vorwärts-Unsicherheitsanalyse unter Anwendung der abgeleiteten Verteilungen der unsicheren Eingangsparameter. Als erste Prüfung wurde eine einfache Hüllkurvenprüfung durchgeführt. Diese kann sowohl visuell als auch mathematisch durchgeführt werden. Letzterer Ansatz verwendet eine ist eine mehrdimensionale Abstandsfunktion, welche mit einem angemessen gewählten Grenzwert für die Akzeptanz verglichen wird.

Für die Validierung von AC²/ATHLET SET und CET konnten im Rahmen des aktuellen Projekts Wiederflutungsversuche der Testreihen FEBA, FLECHT und PERICLES untersucht werden. Die Ergebnisse eines dieser Versuche wurden auch für die inverse Unsicherheitsquantifizierung im Rahmen der Methodikentwicklung verwendet. Die Ergebnisse der genannten Testreihen zeigen, dass das qualitative Verhalten für alle drei Testfälle durch ATHLET vorhergesagt werden kann, allerdings teilweise mit zeitlichen Abweichungen während der Wiederflutung.

Der ATHLET-Validierungsband wurde um zusätzliche Anleitungen zur Anwendung der Wilks-Formel, zur Bestimmung von Stichprobengrößen für die Unsicherheitsanalyse und zur Behandlung von Code-Abstürzen erweitert, zunächst für ATHLET 3.3 und dann nochmals erweitert für ATHLET 3.4. Die Ergänzungen des Handbuchs gehen auf die diskutierten Themen näher ein und fassen diese für ATHLET-Anwender zusammen. Darüber hinaus wurden eine Tabelle zum Ablesen von Mindeststichprobengrößen sowie Tabellen zur Abschätzung von oberen und unteren Grenzstichproben für verschiedene Perzentile bereitgestellt.

Im Rahmen der internationalen Aktivitäten beteiligte sich die GRS vor allem an zwei Projekten bzw. Netzwerken, OECD/NEA ATRIUM und FONESYS. Beide Projekte befassen sich – zumindest teilweise bei FONESYS - mit der inversen Unsicherheitsquantifizierung. In OECD/NEA ATRIUM ist ein Benchmark für die inverse Unsicherheitsquantifizierung das Hauptziel. Die GRS hat an der ersten Phase teilgenommen, die noch andauert, und der Abschlussbericht ist in Vorbereitung. Im Rahmen von FONESYS beteiligte sich die GRS an einem virtuellen Benchmark für die umgekehrte natürliche Zirkulation auf der Grundlage des Experiments PKL F1.2. Ein Hauptziel war die Vorhersage der Druckverluste für ein solches Szenario. Die Ergebnisse zeigen, dass es keine erkennbare Korrelation zwischen Druckverlust und Massenstrom bei Zweiphasenströmung gibt.

Basierend auf dem aktuellen Stand der Modellierung für die inverse Unsicherheitsquantifizierung, die für AC²/ATHLET-Anwendungen verwendet wird, konnten Themen mit Verbesserungspotential identifiziert werden:

- Untersuchung verschiedener ABC-Methoden für inverse Quantifizierung
- Erweiterung der experimentellen Datenbasis
- Speicherung von Zwischenschritten während der Simulation zur Erhöhung der numerischen Stabilität und besseren Nutzung von Ressourcen
- Berücksichtigung eines Modellbias
- Ersetzen/Verbesserung des Likelihood-Ansatzes
- Verwendung von Surrogate-Modellen

Wie bereits erwähnt, sollte die Basis von SET und CET erweitert werden, um die Methodik auf mehr als ein Experiment anzuwenden und die inverse

Unsicherheitsquantifizierung auf eine breitere Basis zu stellen. Neben Wiederflutungsexperimenten, wie sie in diesem Projekt verwendet werden, sollten weitere Versuchsreihen mit zusätzlichen Phänomenen untersucht werden.

Durch die Teilnahme an Arbeitsgruppen, internationalen Projekten und experimentbegleitenden Netzwerken könnte die GRS die Definition neuer Experimente, die detaillierte Spezifikation von Experimenten, die Auswahl der zu messenden Größen sowie den Umfang und die Detailliertheit der Dokumentation von Experimenten beeinflussen und mitbestimmen. Auf diese Weise könnten Erkenntnisse und offene Fragen aus der Entwicklung des Forschungsprogramms, aus der Validierung von Experimenten auf anderen Skalen und aus der Anwendung der Computerprogramme in konkrete Anforderungen an die Durchführung von Experimenten auch hinsichtlich der inversen Unsicherheitsquantifizierung umgesetzt werden. Die Diskussion der Versuchsergebnisse in den Arbeitsgruppen, internationalen Projekten und Netzwerken und deren sicherheitstechnische Auswertung gaben darüber hinaus einen frühen Überblick über die Bedeutung der Versuche für die AC2-Validierung und die inverse Unsicherheitsquantifizierung. Die Präsentation der eigenen Analysen in den begleitenden Arbeitsgruppen, Projekten oder Netzwerken ermöglicht den Erfahrungsaustausch mit den Experimentatoren und mit Analytikern, die andere fortgeschrittene Rechenprogramme einsetzen.

Table of Contents

1 Introduction

The safety of nuclear facilities needs to be demonstrated through safety assessment. The safety assessment, which should follow a graded approach, needs to take into account uncertainties relevant to facility, e.g., a nuclear power reactor, and show that adequate safety margins are provided in the design to prevent unacceptable consequences. In the safety assessment, either adequately bounding (or conservative) assumptions are made or an investigation with dedicated uncertainty and sensitivity analyses is necessary (see, e.g., IAEA GSR Part 4 (Rev. 1), Requirement 17 /IAEA 16/). Safety assessments are usually supported by simulations with validated computer codes. In addition to any uncertainties related to the facility and its processes and phenomena, also the uncertainties associated with these tools need to be understood and characterized (see, e.g., IAEA GSR Part 4 (Rev. 1), Requirement 18 /IAEA 16/ and IAEA SSG-2 (Rev. 1), paragraphs 5.34 - 5.36 /IAEA 19/). This is particularly important if a state-of-the-art best-estimate plus uncertainty analysis is envisaged for the safety assessment of a nuclear reactor, see also section 6 of IAEA SSG-2 (Rev. 1) /IAEA 19/.

The system code ATHLET is part of the AC² code package by GRS for the use in safety analyses of nuclear reactors and other nuclear facilities /WEY 24/. Consequently, uncertainties associated with ATHLET need to be understood. Because a system code like ATHLET will react to uncertainties determined directly from experimental data in ways that are hard to foresee, which is partly due to the specific models and their assumptions in ATHLET and also partly due to scaling issues and other distortions, a dedicated determination of uncertainties for such a system code is sensible /HOL 23/. Importantly, this will have to be based on inverse uncertainty quantification following a systematic approach to the extent possible /NEA 18/, /SKO 22/.

This project focused on methods to address this task.

1.1 Objective of the project

In deterministic safety analyses with realistic assumptions (best-estimate) with (plus) uncertainty analyses (BEPU analyses), model parameters that are not sufficiently known/characterised and can significantly influence the results are considered uncertain and described with an uncertainty distribution. Sources of uncertainty are both the models used in simulation codes (model uncertainties), as well as modelling assumptions,

simplifications, or estimates (e.g., due to knowledge gaps) in the input data (e.g. nodalisation or specific pressure loss characteristics of the plant as well as other user effects). An appropriate determination of model uncertainties for the AC² programme system should be carried out for current code versions and therefore requires regular review and extension /HOL 23/. Recommendations for important model uncertainties and their distributions should be handed over to the code user as part of the code documentation (see also IAEA SSG-2 (Rev. 1) paragraph 5.41 /IAEA 19/). Furthermore, a quality assurance objective for the code development of AC² is that model uncertainties for new models and their interactions with others are determined and documented as an integral part of the programme development and validation. In addition to an expansion and safeguarding of the validation basis, needs for further model improvements and for additional models can be derived in this way.

1.1.1 Methodology

Requirements for a procedure suitable for the AC² codes for the determination of model uncertainties in the context of development and validation are to be derived and generic requirements for the selection of suitable experiments are to be established. Using ATHLET as an example, a suitably adapted methodology will be developed and implemented to enable consistent determination of several uncertain parameters simultaneously using Combined Effect Tests (CET). Methods of Bayesian statistics, in particular the data assimilation approach, are considered to be particularly promising. These methods also make it possible to update distributions of uncertain parameters when new data is available, without having to carry out the whole determination process again, and to take into account new and previous information, e.g. from the literature. It should be possible to include the relevance or informative value of the respective experiment for the phenomena to be quantified, but also, if necessary, other aspects such as experimental measurement uncertainties or transfer from other working media. An own determination of experimental uncertainties does not take place, but the quantities determined on the part of the experimenters are taken as a basis.

The procedure to be developed shall be suitable for the AC² codes, but will be developed and implemented in a first step using the example of and for ATHLET:

- Derivation of the requirements for the determination of model uncertainties, establishment of criteria for the suitability and relevance of experiments for the derivation

of model uncertainties and consideration of the GRS-internal quality assurance requirements.

- Clear presentation of the methods for Bayesian inference and, in particular, data assimilation approaches. On the basis of the agreed requirements, an assessment is made of the need for adaptation of established methods, which external tools or libraries can be used, which have to be (further) developed and which need to be adapted to ATHLET. On this basis, a work plan is developed.
- Implementation of the work plan by describing a method for the determination of model uncertainties from CETs, and for updating the model uncertainties by taking into account new data and their properties (such as relevance), as well as by developing and qualifying tools or scripts for the implementation of this method for ATHLET model uncertainties.

Improvement of methods and their implementation based on the findings from the exemplary implementation in WP 2.

1.1.2 CET and SET

In this individual objective, selected model uncertainties will be quantified to accompany and support the work on the new statistical method. In view of the assessment of the quality of the uncertain ATHLET model parameters, which are particularly relevant for LOCA analyses, the model parameters for the critical flow as well as those for the reflooding models based on CETs will be quantified as a matter of priority. These have a substantial influence e.g. on the first and second temperature peak in a 2F LOCA. This is complemented by an evaluation of simple Single Effect Tests (SET), mainly to support the method development in the first sub-objective.

- Exemplary evaluation of CET to support model development
- Quantification of further model uncertainties from CET for model verification
- Updating quantified model uncertainties by SET

1.1.3 Documentation

The basis of experimental data available to GRS that can be used for quantification of uncertain parameters shall be expanded. Additionally, suitable SETs or CETs are to be made available for automated regression tests.

An internal GRS recommendation is to be developed on how model uncertainties can be taken into account in code development and validation.

The user documentation of ATHLET is to be updated and supplemented by recommendations on the performance of uncertainty analyses.

- Make experimental data available to validate the critical flow and reflooding models.
- Updating the validation matrix of ATHLET for these phenomena
- Transfer of selected experiment runs into the continuous integration matrix of AC²
- Elaboration of a GRS-internal recommendation on how the determination of model uncertainties can be integrated into the continuous development and validation of the AC² programmes
- Supplementation of the user manual with updated recommendations on the performance of uncertainty analyses

1.1.4 International Activities with Reference to Uncertainty Analyses and Cross-Cutting Activities

Uncertainty analyses are a topic in various working groups of the OECD, the IAEA, and also in other organisations, such as the FONESYS group, a voluntary association of developers of nuclear system codes. In the FONESYS group, an activity is planned in which uncertainties of closure sequences in the system codes are to be quantified and in which GRS will participate. This includes in particular:

- Exchange of experience for the determination of model uncertainties,
- Comparison of methods for the quantification of closure equations in the system codes,
- Benchmark of methods for quantifying model uncertainties on the basis of CETs.

In addition, a follow-up activity for the SAPIUM activities is planned within the framework of the OECD/NEA. New results on uncertainty analyses are regularly presented at relevant nuclear conferences. This is actively pursued, e.g., through participation in specialised working groups, meetings, or conferences. Furthermore, the status of the application of modern statistical methods, e.g., data assimilation approaches, is actively monitored, e.g., by attending specialist conferences.

During the project duration a benchmark about reverse natural circulation based on the PKL F1.2 experiment was launched within in FONESYS. GRS participated in that activity.

1.2 Structure of the report

In section 0 the code ATHLET as part of the code system AC² is described. In section 1 the developed methodology for inverse uncertainty quantification including its verification is described while in section 4 suitable test series for the validation of ATHLET and inverse uncertainty quantification is given. The update of the ATHLET documentation and the currently used GRS method for uncertainty and sensitivity analysis is described in section 5. The participation in international projects and activities with reference to uncertainty analysis is given in section 6. Finally, the conclusions for the development of the inverse uncertainty quantification, validation and international activities with ATHLET, are summarized in section 7, including an outlook on possible follow-up work.

2 Main computer codes in the project

2.1 AC²/ATHLET

The main thermohydraulic computer code used in the project was the GRS code AC² /WIE 19/, /WEY 24/ and specifically its constituent code ATHLET in the release versions ATHLET 3.3 /AUS 22/ and ATHLET 3.4 /SCH 24/ as well as some internal developer versions available throughout the project.

The thermal-hydraulic computer code ATHLET (Analysis of THermal-hydraulics of LEaks and Transients) is being developed by the Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) for the analysis of operational conditions, abnormal transients and all kinds of leaks and breaks in nuclear power plants. The aim of the code development is to cover the whole spectrum of design basis and beyond design basis accidents (without core degradation) for PWRs, BWRs, SMRs and future Gen IV reactors with one single code /HOL 22/ .

The main code features are:

- advanced thermal-hydraulic modelling (compressible fluids, mechanical and thermal non-equilibrium of vapour and liquid phase)
- availability of diverse working fluids: light or heavy water, helium, sodium, lead or lead-bismuth eutectic, supercritical carbon dioxide, molten salts as well as user-provided single-phase (non-boiling) working fluids
- heat generation, heat conduction and heat transfer to single- or two-phase fluid considering structures of different geometry, e.g., rod or pebble bed
- interfaces to specialized numerical models such as 3D neutron kinetic codes or 3D CFD codes for coupled multiphysical or multiscale simulations
- control of ATHLET calculation by call backs to programming language independent user code enabling the coupling of external models
- plug-in technique for user provided code extensions
- modular code architecture
- separation between physical models and numerical methods

- numerous pre- and post-processing tools
- portability
- continuous and comprehensive code validation

A detailed description of the code can be found in /LER 19/, /AUS 21/.

3 Methodology

3.1 Uncertainty Analysis

Safety analyses using best-estimate codes are widely used in safety demonstration of nuclear facilities. The application of these codes can be combined with conservative assumptions about system availability and initial and boundary conditions, as well as with realistic, best estimate values (Table 1 in IAEA SSG-2 (Rev. 1) /IAEA 19/). Historically, the conservative approach dominated for safety analysis, where uncertainties are implicitly accounted for by the user through the selection of appropriate (conservative) parameters that penalize the simulation regarding the relevant acceptance criteria, like, e.g., peak cladding temperature. This is included in several regulatory guidelines (German Safety requirements /BMUV 22/ in Annex 5, paragraph 3.4, and in its Appendix 1; German KTA rules; Appendix K /NRC 19/, etc.).

Alternatively, an analysis with a best-estimate code and realistic assumptions (so-called best estimate) along with an uncertainty analysis can be conducted, see, e.g., the German Safety Requirement, Annex 5, paragraph 3.2 and 3.3 /BMUV 22/ and similarly IAEA SSG-2 (Rev. 1) /IAEA 19/. This is known as best-estimate plus uncertainty (BEPU) and is increasingly being used in safety analyses supporting the safety demonstration of nuclear reactors and their review by regulatory authorities. Additionally, it is being applied in safety analyses in research and verification procedures as well. A quick conceptual overview of relevant terms like acceptance criterion on safety margin relevant to BEPU application for safety demonstration is shown in Fig. 3.1.

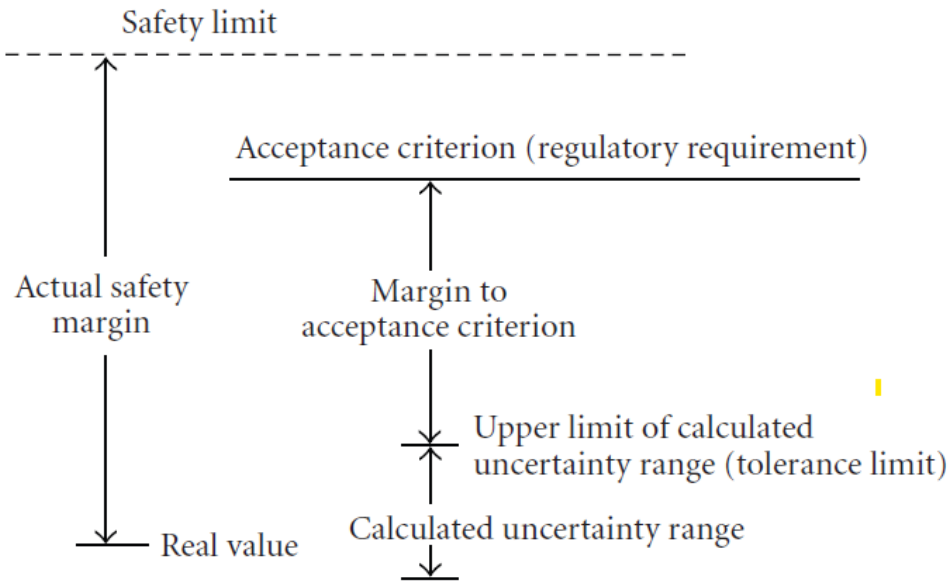

Fig. 3.1 Margin illustration

Uncertainty Quantification (UQ) is a crucial step in validating computational models, as assessing model accuracy requires a measurable and concrete evaluation of uncertainty in model predictions. In the nuclear field, UQ typically refers to forward UQ (FUQ), where uncertainty propagates from model inputs to outputs (or is being extrapolated, see below). The widely applied methods of FUQ require uncertainty distributions describing the uncertain input parameters. Their identification and quantification currently rely heavily on expert judgment and user assessment. While sensitivity analysis or parameter studies aid such estimation procedures, expert opinion is in in most cases unavoidable.

Inverse uncertainty quantification (IUQ) addresses exactly this issue. During an IUQ process, the information from (code) model outputs and from experimental data is the basis for deriving and refining the input uncertainties. It aims to describe the uncertain input parameters with statistical methods so that code results are consistent with observations. However, these processes, even though not less important than FUQ, have only recently gained more attention. A conceptual overview is given in Fig. 3.2.

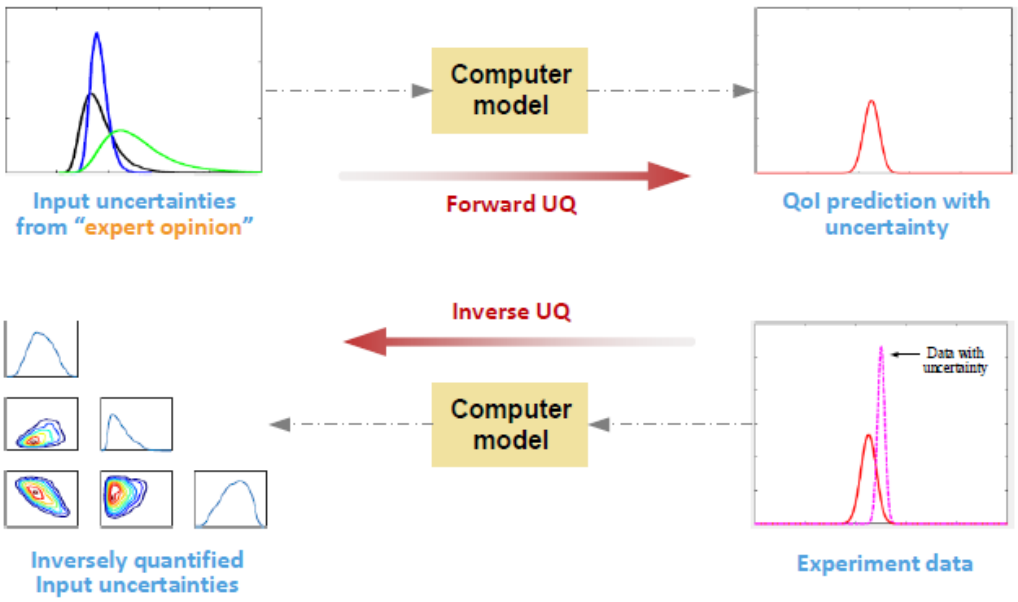

Fig. 3.2 FUQ vs IUQ /WU 21/

Forward Uncertainty Quantification

The two main approaches applied in the nuclear thermo-hydraulic field for carrying out a forward uncertainty analysis are forward propagation of the uncertain input parameters through code simulation, and the extrapolation of the output uncertainties. The latter one is the basis for the Uncertainty Methodology based on Accuracy Extrapolation (UMAE) method /DAU 95/, sometimes also referred to as the Pisa-method. The methods based on the forward propagation of the uncertain input parameters include, e.g., the so-called GRS method /GLA 08/ and also the Code Scaling, Applicability and Uncertainty (CSAU) method /BOY 89/.

The GRS method makes use of the findings of Wilks /WIL 41/, /WIL 42/, for determining the minimum required sample size for a distribution free tolerance limit (i.e. upper/lower limits are derived, but Wilks' formula provides no further information for what happens in between, and an output pdf would have to be approximated separately using the samples obtained in the GRS method). The main advantage of Wilks approach is that it is distribution-free, i.e. we do not have to make any assumptions about the results obtained via the simulation code or the functional relationship between input and output uncertainties, except that code outputs must be a random sample and that they have to be suitable for a rank ordering relation. Furthermore, it has the advantage that the required number of samples N (i.e. code simulations) does not depend on the number of uncertain input parameters but can be solely determined based on the chosen probability content α and

confidence level β (α/β). Simultaneous variations of the uncertain input parameters are necessary according to their respective pdfs, while the tolerance limits are determined by appropriately chosen order statistics. The minimum number of simulations N for a given probability α and confidence level β can be determined based on Wilks` findings using eq. (3.1) and (3.2) for one- and two-sided tolerance limits, respectively. The minimum number of required calculation for some α/β can be found in Tab. 3.1.

$$1 - \alpha^N \geq \beta \tag{3.1}$$

$$1 - \alpha^N - N(1-\alpha)\alpha^{N-1} \geq \beta \tag{3.2}$$

Tab. 3.1 Minimum number of required calculations according to Wilks

One-sided				Two-sided			
ß/α	*0.90*	*0.95*	*0.99*	*ß/α*	*0.90*	*0.95*	*0.99*
0.90	22	45	230	*0.90*	38	77	388
0.95	29	59	299	*0.95*	46	93	473
0.99	44	90	459	*0.99*	64	130	662

The most commonly used tolerance limit in safety demonstration is 95/95 due to its acceptance by regulatory authorities, see, e.g., Annex 5 to the German Safety Requirements, paragraph 3.3 (3) /BMUV 22/, for licensing, therefore making it relevant in the nuclear industry /IAEA 08/. More information on the background and application of the GRS method can be found in e.g. /HOL 23/ and in chapter 5.

In case of the forward propagation methods such as the GRS-method one of the most crucial steps is the identification and quantification of the uncertain input parameters. This is of utmost importance to avoid “rubbish in, rubbish out” situations and to be able to produce a robust safety assessment that meets regulatory expectations. This quantification generally has been done by the code user, which can be overwhelming, since there might be not enough information available about the modelling approaches as well as the implementation of methods within the codes. One solution is extending the validation of the code with uncertainty quantification of model uncertainties by the developers /UNA 11/, /SKO 19/, /SKO 18/. At the same time, code users are still responsible for uncertainties related to the specific modelling choices in the input model for their facility.

While they can also attempt quantifying model input uncertainties, they have a major disadvantage compared to developers, who are in possession of implementation information and coding details.

3.2 Importance of uncertain input parameters

There has been extensive work in the international nuclear community for the evaluation of the BEPU methodologies and to establish a harmonised approach for the determination of the uncertain input parameters and their quantification. The BEMUSE /CRÉ 08/, /PER 10/ PREMIUM /NEA 17/, and SAPIUM /BAC 20/, /BAC 20/, /NEA 18/ benchmarks or the ongoing ATRIUM /NEA 23/ project are important international collaborations under the OECD framework in establishing such approaches.

The BEMUSE (Best Estimate Methods and Sensitivity Evaluation) programme represents an important step towards reliable application of high-quality best-estimate and uncertainty and sensitivity evaluation methods and aimed at evaluating the performance of different BEPU approaches on the basis of two exercises. The first focused on conducting an uncertainty and sensitivity analysis for the LOFT L2-5 test**,** an experimental simulation of a large break loss-of-coolant accident (LB-LOCA). The second step applied the same analysis to a nuclear power plant CL-LB-LOCA scenario. It was concluded that the BEPU methodologies were mature enough to be utilised in licencing applications. At the same time, a clear need for a reliable quantification of the input model uncertainties (IUQ) was identified.

In the PREMIUM (Post BEMUSE Reflood Models Input Uncertainty Methods) Benchmark /CSNI 16/, various IUQ methods were compared based on two reflooding experiments (FEBA, PERICLES). Large discrepancies among the results obtained by different participants were identified. Furthermore, the activity showed difficulties to validate the uncertainties estimated on the quantification database (FEBA) against the data in the validation data base (PERICLES). The lack of common consensus and best-practice guidelines have been identified as some of the main reasons behind these discrepancies. Some of the approaches addressed in the project are briefly described below.

CIRCE (Calcul des Incertitudes Relatives aux Corrélations Élementaires) is a method developed by CEA based on a Maximum Likelihood approach. For IUQ, it is assumed that the uncertainty distributions of the parameters follow a normal distribution and that

the simulation quantities are linearly dependent on the input uncertainties. The calculation of uncertain parameters is based on comparing the difference between measured and simulated values combined with calculating local derivatives, upon which an iterative process determines a local maximum for the description of the measured values through the uncertainty distributions. The strong assumptions mentioned above, combined with the limitation to specific time points, make this approach less suitable for further development. CIRCE is currently being supplemented with methods of Bayesian inference /DAM 18/. The IPREM method from the University of Pisa /KOV 17/ applies a Fast Fourier Transformation to the deviation of experimental measurements from the corresponding simulation quantities. This calculates an error metric in the frequency domain. Uncertainty distributions are then determined using expert estimates that minimize this error metric. The IRSN method DIPE (Determination of Input Parameters Empirical Properties) /JOU 08/ compares how many measured data points exceed the simulation for uncertain parameters through sequential variation and derives a cumulative distribution function. This is averaged with results from other experiments to estimate model uncertainties. This leads to very conservative distributions of model uncertainties in multivariate evaluations and was therefore not selected as a basis for this project.

TRACTEBEL developed an iterative method where several prior distributions are propagated through the model using DAKOTA-Statistical Software /ADA 10/, and the results are compared with measured data. It is checked whether sufficient data is included between the upper and lower quantiles of the result distributions. If not, the prior distributions are adjusted through expert estimation, and the process is then repeated /ZHA 19/. GRS has applied a similar method, where the number of iterations was optimized by applying higher-order samples and considering distributions already determined from SET /SKO 17/. However, this approach has only been tested for the calculation of one parameter. In theory, a minimization of an appropriately defined error metric through generally nonlinear, multivariate optimization methods would also be possible, but this has not been realized so far.

MCDA (Model Calibration through Data Assimilation) is a development by KAERI and has been implemented in the PAPIRUS tool /HEO 15/. This method is based on a Data Assimilation approach for a Bayesian update of prior distributions. For linear dependencies and normal prior distributions, algebraic formulas are evaluated to calculate the posterior distributions. For non-linear dependencies and non-normal prior or posterior

distributions, numerical methods, such as Markov Chain Monte Carlo, are required to evaluate the Bayesian update.

The main recommendation from the PREMIUM benchmark was that a systematic approach devoted to model input uncertainty quantification and validation should be developed to improve the reliability of the analysis and to ensure the extrapolation of its results to the nuclear power plant case. This has been realised in the SAPIUM (Systematic Approach for Input Uncertainty Quantification Methodology) project /BAC 20/, /BAC 20/, /NEA 18/. Within this framework a stepwise approach based on an experimental-based quantification process has been established (Fig. 3.3). This included several recommendations and best-practice guidelines to perform IUQ. At the end of the project the need to test the applicability of the established guidelines was highlighted.

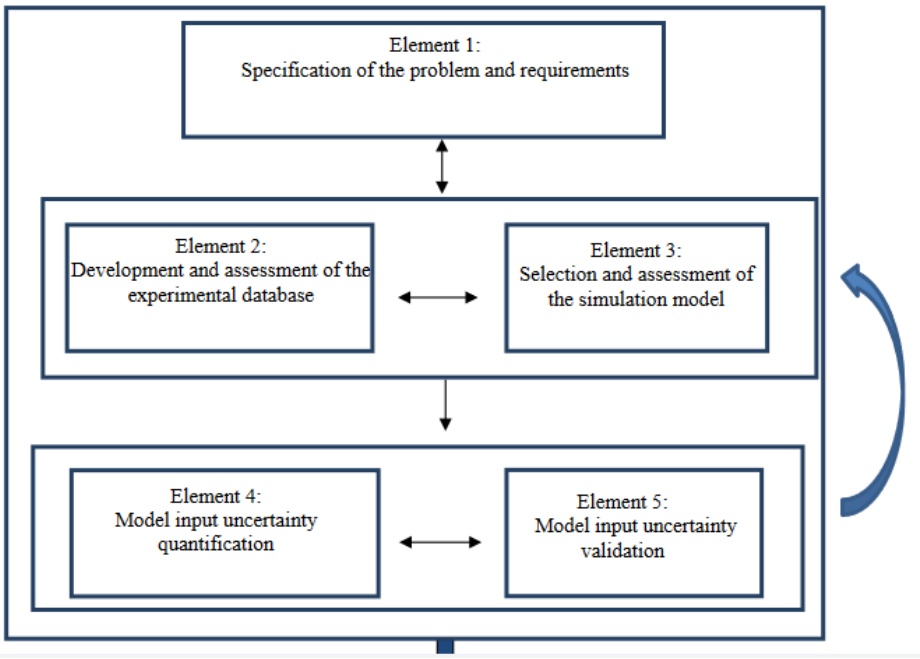

Fig. 3.3 The five key elements of IUQ as established by the SAPIUM project /BAC 20/

The guideline established within SAPIUM has been/is being tested in the ATRIUM (Application Tests for Realisation for Inverse Uncertainty quantification and validation methodologies in thermal hydraulics) project /NEA 23/ still ongoing in 2025. Its main objectives are to verify the applicability of the best practices, identify possible issues, and if necessary, update the recommendations based on the outcomes of the project. The project stretches over three phases: the first has been dedicated to the critical mass flow rate (see section 6.1), the second one focuses on the post-CHF, while the third one

applies the finding on a suitable IET, to validate the derived uncertainties and to possible justify their extrapolation to reactor scale.

3.3 Inverse uncertainty quantification

The quantification of uncertain input parameters currently relies heavily on engineering judgment. As highlighted in the SAPIUM guidelines there are some steps that continue to rely on this judgement, but the goal to reduce this reliance is one that is worth putting effort in. In this study the elements 1 and 2 of Fig. 3.3 have been mostly addressed in the preparation phase of the project. The main element addressed in the present study is element 4: Model input uncertainty quantification. This is basically the focus of work package 1. As the method is developed and tested based on the reflooding modelling that defines the problem (element 1) and the pre-selection of the available data base (element 2). The selected experimental set-ups (FEBA, FLECHT, PERICLES) have been extensively used previously to validate and evaluate the reflooding models in ATHLET. At the same time, not the whole experimental matrix should be involved in the validation. Therefore, these experiments have been selected already in the preparation phase and deemed as a suitable basis for the development of a statistical method for the actual IUQ procedure. Element 3 has been partially addressed in the framework of this project (see chapter 3.8.2). The models applied are extensively studied in previous projects, see, e.g., /SKO 16/, /TIB 15/, therefore no further assessment has been deemed necessary. The choice of the actual model parameters is described in chapter 3.8.2.

Generally, IUQ methods can be categorised into three main groups: deterministic (frequentist), probabilistic (Bayesian, Data Assimilation, etc), and empirical (design-of-experiments, case-to-case approach). An overview of IUQ method developed in the last years is summarised in Tab. 3.2.

Tab. 3.2 Overview of available IUQ methods (* applied in PREMIUM benchmark)

IUQ method	Category	References
*CET-based Sample Adjusting (GRS, Germany)	Empirical	/SKO 17/
*CIRCE (CEA, France)	Frequentist	/CRÉ 01/, /CRÉ 04/, /CRÉ 12/
*DIPE (IRSN, France)	Empirical	/JOU 08/

IUQ method	Category	References
*IPREM (UNIPI, Italy)	Empirical	/PET 11/, /KOV 14/, /KOV 17/
*Sampling-based IUQ (Tractebel, Belgium)	Empirical	/ZHA 19/
*MCDA (KAERI, South Korea)	Frequentist /Bayesian	/HEO 14/
Non-parametric Clustering (PSI, Switzerland)	Empirical	/VIN 07/
Data Adjustment and Assimilation (KIT, Germany)	Frequentist	/CAC 10/, /CAC 10/, /PET 10/
CASUALIDAD (NINE, Italy)	Frequentist	/PET 08/, /PET 14/, /PET 19/
MLE and MAP (UIUC, USA)	Frequentist /Bayesian	/SHR 16/, /HU 16/, /ABU 19/, /ABU 19/
Modular Bayesian Approach (UIUC, USA)	Bayesian	/WU 18/, /WU 18/, /WU 18/, /WU 19/
Bayesian CIRCE (CEA Saclay, France)	Bayesian	/DAM 20/, /DAM 18/

The main goal of the project has been to establish a statistics-based IUQ method combined with experimental data, which can reduce the reliance on expert judgment and quantify multiple uncertain input parameters simultaneously while considering IETs and CETs. Therefore, our goal has been to establish a probabilistic approach that can fit our needs. The method was envisaged to exhibit, among others, the following important characteristics:

- flexible and adaptable to other quantification processes,
- robust,
- being able to quantify several parameters at once,
- being able to use IET and CET as basis for the quantification,

- easy incorporation of new information (without having to redo the whole quantification process),
- feasibility should be kept in mind (CPU intensity, etc.),
- easy to use (once in a closed software-like format).

While some expert judgment is still necessary, e.g., for selecting the relevant uncertain input parameters and the appropriate experimental data[1], the actual quantification process should be shifted to a statistically sound basis.

3.4 Bayesian inference

Bayesian inference is a powerful statistical method rooted in probability theory, allowing for the updating of beliefs or knowledge about uncertain events as new information becomes available. Since its first formulation in the 18th century, Bayesian inference has become one of the most widely applied techniques in various fields, including science, engineering, economics, medicine, and machine learning. At its core, Bayesian inference provides a formal framework for combining prior knowledge with observed data to make informed decisions and predictions.

Bayes' Theorem defines how to incorporate new information into our prior knowledge and come to a more informed state. /BAY 63/ (3.3)

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)} \tag{3.3}$$

where:

- A, B are two events
- P(A|B) is the posterior probability of A given B
- P(A) is the prior probability of observing A (without regard to B)

[1] Efforts in the international community, especially within the framework of the ATRIUM project, are ongoing to establish guidelines in order to reduce the weight of expert opinion within the whole IUQ process. See e.g. /BAC 24/, and section 6.1.

- P(B) the probability of observing B (evidence/marginal likelihood)
- P(B|A) – likelihood, the probability of observing B given that A is true

Prior Distribution: The prior distribution represents the initial belief or assumptions about the uncertainty in a parameter before any data is observed. In Bayesian analysis, it is essential to choose a prior that appropriately reflects the problem at hand. Priors can be either: informative priors, which are based on existing knowledge or expert judgment about the parameter, or non-informative priors, which reflect a lack of prior knowledge, typically assuming equal probability across all possibilities. One of the main concerns is the selection of the prior distribution since different priors might lead to different conclusions. In a perfect world, however, (availability of data, computational resources) with enough iterations the posterior always converges to a specific distribution (in a well-defined system to the true value). In practical approaches convergence and quality of the results must be tested.

Likelihood: The likelihood function expresses how likely it is to observe the given data under different parameter values. It quantifies how well a particular hypothesis explains the observed data and is central to the process of updating the prior distribution. It is also one of the hardest steps to determine its value. In practice, a log-likelihood function is often applied in order to simplify the inference processes. (With log-likelihood the products of the probabilities turn into sums, making complex probability models more manageable to compute and differentiate).

Deriving the likelihood function can be challenging, especially for complex models and high-dimensional data, as it often requires intricate mathematical formulations and assumptions that may not easily align with real-world data. In Bayesian inference, the likelihood function represents the probability of observed data given specific model parameters, forming a crucial component in calculating the posterior distribution. However, the function must capture all relevant dependencies within the data-generating process, which can be particularly demanding for non-linear, hierarchical, or latent-variable models. When models contain numerous interdependent parameters, or when there is latent or missing data, deriving a closed-form likelihood function becomes analytically intractable. Additionally, high-dimensional data amplifies these difficulties, as the likelihood function's complexity grows with the dimensionality, making it computationally expensive and, at times, infeasible to compute.

Posterior Distribution: The posterior distribution combines the information from the prior distribution and the likelihood. It reflects the updated beliefs about the parameters after considering the new data. For complex models, analytical derivation of the posterior distribution is often not feasible and instead numerical approximations are used (e.g. MCMC)

Evidence (Marginal Likelihood): The evidence term normalizes the posterior distribution to ensure that the probabilities sum to one. It represents the total probability of the observed data, accounting for all possible values of the parameters. While this term plays a crucial role in Bayesian model comparison, it is often challenging to compute directly in practice, especially for complex models.

The main differences between Bayesian and frequentist approaches lie in how model parameters are treated. In frequentist statistics, parameters are considered fixed but unknown quantities, and estimation focuses on finding a single best estimate value. Uncertainty is expressed via the sampling distribution, relying on the concept of infinite repeated sampling to make inferences about the parameter.

In contrast, Bayesian statistics treats model parameters as random variables with probability distributions, representing uncertainty based on observed data and prior knowledge. These (posterior) distributions capture the uncertainty surrounding the parameters, with some values being more probable than others based on the observed data and prior knowledge. This approach makes it promising for Inverse Uncertainty Quantification (IUQ) applications since, rather than calibrating a single "best estimate" of model input variables, it allows treating them as uncertain random parameters with probability distribution functions - either in closed analytical form or numerically. Bayesian inference provides a way to update our prior belief (initial understanding of the hypothesis) by combining it with new data (simulation of experimental results), resulting in the posterior. It is basically the reallocation of the credibilities across possibilities (i.e., probabilities across possible parameter values).

Advantages of Bayesian inference

- Incorporation of prior knowledge: One of the primary advantages of Bayesian inference is its ability to systematically incorporate prior knowledge or expert judgment into the analysis. This is particularly useful in situations where data is limited or expensive to obtain, allowing for more informed predictions.

→ This allows to combine researchers' subjective beliefs and evidence from data. This is crucial in our case.

- Quantification of uncertainty: Bayesian methods provide a natural way to quantify uncertainty in parameter estimates. The posterior distribution provides a complete picture of the uncertainty by offering a probabilistic description of the parameters, rather than a single point estimate.
- Flexibility: Bayesian inference is highly flexible and can be applied to a wide range of models and data types. It is particularly useful in complex models where classical methods may struggle, such as hierarchical models, nonlinear models, or models with missing data.
- Learning over time: Bayesian inference is dynamic, meaning it can update beliefs iteratively as new data becomes available. This makes it suitable for real-time applications where continuous learning and adaptation are required.
- Bayesian methods provide a formal framework to support decision making under uncertainty. The ability to quantify and incorporate uncertainty into the decision-making process leads to more robust and informed decisions.
- Computation became easier in recent years: several libraries (in R, Python, etc.) being available, allowing a smoother integration of a Bayesian approach.

Challenges of Bayesian inference

- Computational complexity: Bayesian methods can be computationally demanding, especially for models with a large number of parameters or complex likelihood functions. Techniques like Markov Chain Monte Carlo (MCMC) are often required to approximate the posterior distribution, which can be time-consuming.
- Incorporating subjective beliefs to data-analytic models requires careful and often time-consuming sensitivity analysis (this point might not always be relevant).
- The selection of a prior distribution is often subjective and can influence the results of the analysis. While informative priors can improve the quality of inferences, they may also introduce bias if not chosen carefully. Non-informative priors, on the other hand, can lead to less precise estimates, especially with small sample sizes. Proper selection requires careful and often time-consuming sensitivity analysis or substantial experience (and thus can introduce user-effects, which over enough iteration diminishes).

- Likelihood functions: it is necessary to have some form, be it in a closed analytical form, or computationally derived with only the distribution family being available. This can be rather challenging in case of complex models (system codes typically will fall into this latter category)
- Computational costs: though several sampling and computational methods are readily available, they do require some extra steps in data analysis, such as model convergence checks.

3.5 Model bias, experimental error, etc.

Identifiability is a difficult question in inverse uncertainty analysis. How to allocate the uncertainties observed to different parts of the various models and implementation approximations (a model is per definition always an approximation of the reality with a specific purpose in mind, for which it is supposed to be optimal – or at least adequate), or the observations themselves (experimental error, systematic errors, instrumental tolerance, etc.). When quantifying several uncertain input parameters at the same time it is difficult to answer this question. It becomes a multidimensional problem and can (and generally does) have several optimum points that would provide the experienced variability of the simulation results sufficiently explaining and covering the experimental values. How to choose between these manifestations, and which one is the “true” one, is a difficult question (if it can be answered at all). Furthermore, the question, whether parameters quantified together can be applied outside their original set, combined with other u.i.p., is also one that should be investigated. A simple representation of the connection between reality, experiments and simulation is shown in Fig. 3.4.

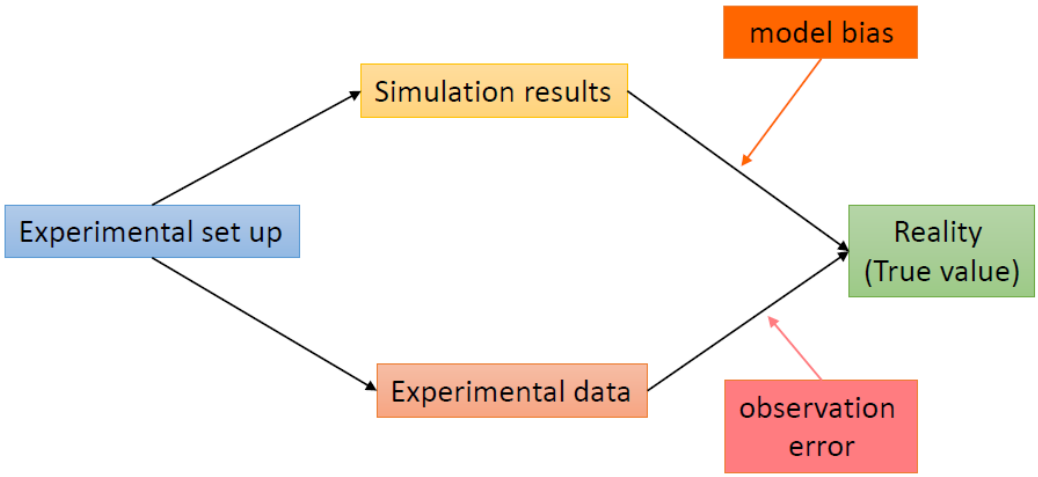

Fig. 3.4 Connection between computer code, experiment and reality

This can be expressed by the following system of equation (3.4) and (3.5).

$$y_E(x_c, \lambda) = y_T(x_c, \lambda) + \epsilon(\lambda) \tag{3.4}$$

$$y_T(x_c, \lambda) = y_M(x_c, x_m, \lambda) + \delta(x_c, \lambda) \tag{3.5}$$

where:

- y_E experimental data (observation)
- y_M model result
- y_T true value
- ϵ observation error
- δ model bias
- x_c controllable inputs (BC, etc.)
- x_m modelling parameters (u.i.p.)
- λ element f the observation layout Λ (e.g. (T, z_1, r_1); (T, z_2, r_1), …)

In case of a perfect model, the model bias is zero (δ = 0). Assuming such a case would inevitably lead to overfitting of the parameter set. Bias could be expressed as a sum of a mean value (i.e. a systematic bias), and a covariance representing the random variation. As ATHLET is extensively validated /HOL 23/, we assume as a first approximation to develop our IUQ method that no systematic bias is present in the model we investigate.

Experimental uncertainty can be expressed as on offset value, i.e. systematic error, and a random variation. The assumption in our study is that the experimental error can be represented as a normal distribution $N(0,\sigma)$ around the experimental value. There is no systematic error (we assume the experimental set up has been sufficiently calibrated and if such a systematic error would have been present the experimentalists would document and compensate for it). To cover most of the variability, we have decided to take 3σ interval covering 99.7%.

Nevertheless, to identify which part of the discrepancy found between observation and simulation is coming from which source is a quite difficult task. This multidimensional problem most of the times has several optimum points yielding similar results and to choose one as the “true” solution is not only difficult but requires engineering judgement. One big advantage in applying Bayesian inference for such analysis is that, regardless of our choice, when new information is obtained it can be easily incorporated into our previous knowledge. Therefore, making it a state-of-the-art prediction, which is not only able but designed to update our beliefs as new information is available.

3.6 Applying MCMC

A Markov chain is a discrete time stochastic process that satisfies the Markov property. It is a sequence of events or states where the probability of moving to the next state depends solely on the current state, not on any previous states. This is known as the Markov property and implies "memory-lessness" in the process, meaning that past states are irrelevant in predicting the future, once the present state is known. Markov chains are typically represented by a transition matrix (3.6) that contains the probabilities of moving from one state to another.

$$\mathrm{T} = \begin{pmatrix} T(S_1|S_1) & T(S_2|S_1) & T(S_3|S_1) \\ T(S_1|S_2) & T(S_2|S_2) & T(S_3|S_2) \\ T(S_1|S_3) & T(S_2|S_3) & T(S_3|S_3) \end{pmatrix} \tag{3.6}$$

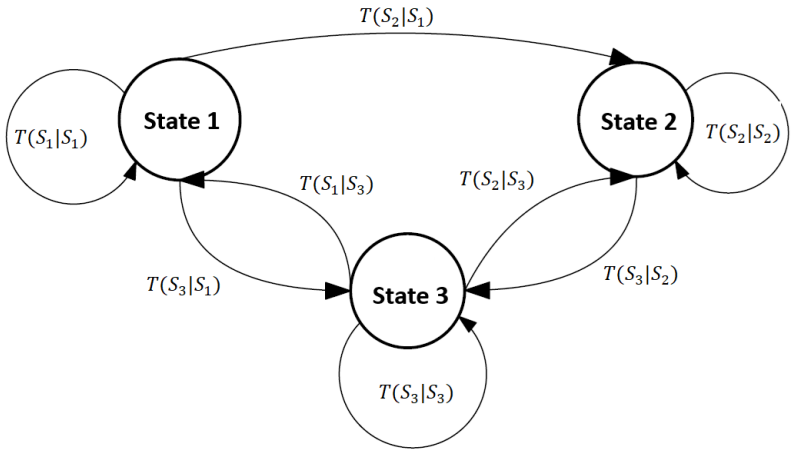

Fig. 3.5 Illustration of a 3-state Markov Chain

Markov Chain Monte Carlo (MCMC) sampling is a computational technique used in Bayesian inference to approximate complex posterior distributions. MCMC generates samples from a target distribution by constructing a Markov chain that has the desired posterior as its equilibrium distribution. There are no restrictions of the target distribution (it does not necessarily have to have a nice analytical closed format), which makes it so useful in Bayesian inference especially with complex models, where other derivations of the posterior would be borderline impossible, or actually mathematically impossible. Common MCMC algorithms include Metropolis-Hastings /HAS 70/ and Gibbs sampling /GEM 84/, each designed to explore the parameter space and converge towards high-probability regions. (See more in chapter 3.7.)

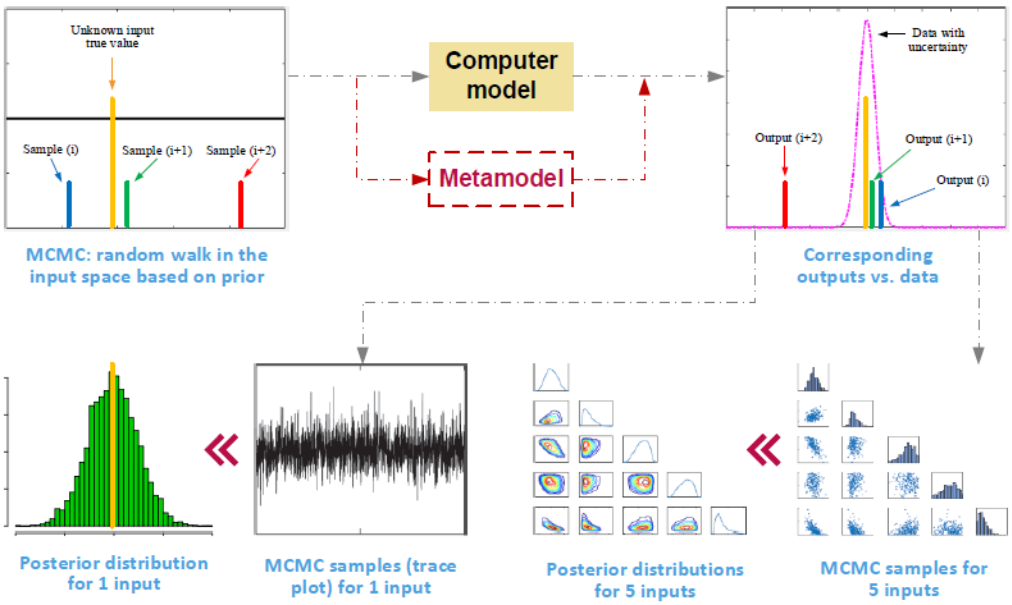

Fig. 3.6 Illustration of MCMC sampling /WU 21/

Applying MCMC (Markov Chain Monte Carlo) is common in Bayesian application, particularly where the derivation of a closed form posterior (or the likelihood) is not possible. For thermal hydraulic codes such as ATHLET, this generally is the case.

At the same time, MCMC does need some extra steps, e.g., its convergence must be checked. A chain is deemed converged, when it provides a reasonably close or representative approximation of the target. The length (or size) of the chain should be long enough to provide a stable and accurate estimator (there should be only small variations in posterior summaries across repeated runs). Furthermore, efficiency is key in order to have a practically applicable method: resulting values should reach a reasonable level of proximity to the target as quickly as possible. Convergence must be reached for all model parameters. There are several features that provide information about this and that can (and should) be checked.

- Mixing: The MCMC chain should thoroughly explore all regions of the parameter space, ensuring that every area is “visited” adequately to represent the posterior distribution fully.
- Autocorrelation: Consecutive samples from the posterior should ideally be independent. While some degree of autocorrelation is common, excessive autocorrelation can slow convergence by requiring higher sample size.

- Sample Size: Sufficiently large sample sizes are essential for achieving reliable posterior estimates and improving convergence. (effective sample size = sample size adjusted for autocorrelation, /KRU 15/.
- Stationarity: Local estimates from different parts of the chain should be consistent, meaning that the chain does not exhibit significant shifts in values as it progresses. (Geweke test /GEW 92/, Heidelberg and Welch /HEI 81/, /HEI 83/, Raftery /RAF 95/, /RAF 92/, running mean).
- Accuracy: percentiles of the posterior should be estimated with sufficient precision.
- Independence from starting values (cross-chain convergence): The final posterior summary should not depend on the initial values, indicating that wherever the chain starts, it will converge to the same posterior distribution.

Convergence is assessed using a combination of visual diagnostics (e.g., trace plots, autocorrelation plots) and formal statistical tests. If signs of non-convergence are detected, the following steps can help improve the study:

- Increased sample size: Adding more samples to the chain is the primary corrective measure to improve convergence.
- Discard initial iterations: To mitigate the effect of the starting-point choice, it is often useful to discard the initial iterations, also known as the burn-in period, from the final inferences (“burn in”).
- Run multiple chains: Running at least three or four chains from different starting points can provide greater confidence in convergence by verifying that each chain reaches the same posterior distribution.
- Choose starting values wisely: Selecting reasonable starting values, such as frequentist estimates, can enhance convergence.
- Thinning the chain: To reduce autocorrelation, it may help to save only every k-th iteration, where k is the lag, instead of every iteration.

It is important to note that convergence in itself does not say anything about the quality of the result. For Bayesian inference, achieving convergence is a necessary, but not sufficient condition. Convergence indicates that the sampling algorithm has reached a stable distribution, but additional assessments are required to verify that the estimates accurately reflect the true posterior distribution.

3.7 Choice of tools

The increasing complexity of scientific models, especially in fields such as physics, engineering, and environmental and social sciences, has led to the development of specialized computational tools designed to handle the computational demands of Bayesian inference. In the framework of this project several of these tools have been reviewed. Our aim was to select a tool that is adaptable for our purposes, while keeping in mind further applications, that should be possible with minimum adjustments in order to avoid a case-to-case necessity of starting practically from square one.

Probabilistic Programming Frameworks allow users to define probabilistic models and perform Bayesian inference on them using various algorithms (e.g., Markov Chain Monte Carlo, Variational Inference). We briefly characterize those we looked into.

PyMC (PyMC3, PyMC4) /SAL 16/ is a probabilistic programming library for Python that enables users to build Bayesian models using a straightforward Python API and fit them with Markov Chain Monte Carlo (MCMC) methods. It also leverages PyTensor (a fork of the Theano project) to boost performance.

Stan /STA 24/ is a high-performance probabilistic programming language with support for Bayesian inference. Stan was developed to support full Bayesian inference, where modelling involves defining the posterior probability function up to a constant proportion. According to Bayes' rule, this is equivalent to modelling the product of the likelihood function and the prior distribution. It provides Hamiltonian Monte Carlo (HMC) and No-U-Turn Sampler (NUTS) algorithms for efficient sampling. It is available in multiple languages like R (RStan), Python (PyStan), and CmdStan.

TensorFlow Probability (TFP) /TEN 24/ is a library built on top of TensorFlow, that makes it easy to combine probabilistic models and deep learning on modern hardware (TPU, GPU). It enables scalable Bayesian analysis and probabilistic models and supports Variational Inference and MCMC methods, focusing on deep probabilistic models.

JAGS (Just Another Gibbs Sampler) /PLU 23/ is designed for the analysis of Bayesian hierarchical models using Gibbs Sampling. Typically used via R through rjags.

Markov Chain Monte Carlo (MCMC) Tools algorithms are crucial for approximating the posterior distribution by generating samples from it. A Markov chain is a process that

obeys the Markov property: having no memory (see above). Conditional probability distribution of future states depends only on the present one, but not on earlier ones. In theory it allows the approximation of posteriors of any complexity. For reliable estimates an MCMC algorithm must converge to the target distribution (for convergence checks see section 3.6). Some common algorithms are:

Metropolis-Hastings Algorithm /HAS 70/ is one of the simplest MCMC methods for obtaining a sequence of random samples from a probability distribution from which direct sampling is difficult. It proposes new parameter values and accepts or rejects them depending on the value of the probability distribution at that point. The resulting sequence can be used to approximate the distribution.

Hamiltonian Monte Carlo (HMC) /BRO 11/ is a more sophisticated method that reduces random walk behaviour (reducing the correlation between successive sampled states) by proposing moves to distant states which maintain a high probability of acceptance. Tools like Stan and PyMC use HMC for efficient sampling.

No-U-Turn Sampler (NUTS) /HOF 11/ is an extension of HMC, automatically tuning the trajectory length of the Hamiltonian system. It is used by Stan and PyMC as a default algorithm due to its efficiency.

Variational Inference (VI) Tools /BLE 17/ Variational inference has gained popularity as a method for approximating posteriors in complex latent variable models. However, developing a variational inference algorithm typically demands extensive model-specific analysis. This process can slow down and limit our ability to rapidly develop and explore different models for a given problem. It is a deterministic approach which can be faster than MCMC but not so reliable. Approaches like *Autoencoding Variational Bayes (AEVB)* /KIN 13/ or *Black Box Variational Inference* /RAJ 14/ are available, and libraries like PyMC and TensorFlow Probability provide built-in support for VI techniques.

Sequential Monte Carlo (SMC) Tools /DOU 01/ SMC methods (also known as particle filters) are powerful tools for Bayesian inference but require a large number of particles to achieve accurate estimates, resulting in high computational costs. The essential idea is to represent the filtered density $p(x1|y1)$ as an empirical distribution represented using a finite weighted sum of point-mass distributions (Dirac delta functions). They update a population of parameter estimates as more data arrives. PyMC has built-in support for

SMC, which is particularly useful when the posterior distribution is multimodal, or the prior is highly informative.

Bayesian Optimization Tools A sequential design strategy for the global optimization of black-box functions that makes no assumptions about their functional form. It is commonly used to optimize functions that are costly to evaluate (e.g. /JON 98/).

Statistical program choice

R /RCO 23/ is a programming language and environment specifically designed for statistical computing, data analysis, and graphical representation. Originating as an open-source alternative to the S language developed at Bell Laboratories, R has evolved to become one of the most widely used tools for data analysis across a variety of fields, including statistics, data science, bioinformatics, and social sciences. Its popularity stems from its powerful computational capabilities, extensive package ecosystem, and adaptability in both academic and industry settings.

While powerful, R has limitations. It can be less efficient with very large datasets compared to some newer data processing languages, though integration with big data frameworks (e.g., Spark via `sparklyr`) is available. Additionally, R's memory-intensive nature and reliance on single-thread processing in some functions can make it slower for specific large-scale or high-performance tasks.

Available packages include: Rstan, BUGS, JAGS, INLA (Integrated Nested Laplace Approximation), MCMpack, etc.

Licensing: depending on the library applied, etc. ranging from BSD to GPL, GPL-3. can get quite complicated.

MATLAB /THE 22/ offers a powerful environment for Bayesian analysis, integrating probabilistic modelling, parameter estimation, and inference with its computational and visualization capabilities. With dedicated toolboxes and functions, MATLAB supports various Bayesian methods, including Bayesian optimization, MCMC methods, etc.

At the same time, MATLAB is not a free tool with quite substantial yearly license fees. On the other hand, code programmed in MATLAB is not inherently under any specific open-source license upon creation. The license can be freely chosen for distribution

(e.g., BSD, MIT, GPL), unless specific libraries, MathWorks-specific channels or products like the MATLAB Compiler is used, in which case their licence requirements must be met.

Python /VAN 95/ is a versatile and popular programming language widely used in data science, statistics, and machine learning, including Bayesian analysis. Its extensive libraries make it particularly suited for building, analysing, and visualizing Bayesian models. Some popular libraries for Bayesian analysis include:

- **PyMC** Facilitates the construction and inference of Bayesian models with a straightforward Python API, supporting MCMC and variational inference methods for estimating complex posterior distributions. /SAL 16/
- **Stan (via PyStan or CmdStanPy)** Allows users to specify Bayesian models using the Stan language and perform Bayesian inference, often used for hierarchical models and complex statistical modelling. /STA 24/
- **TensorFlow Probability (TFP)** A library for probabilistic reasoning that integrates Bayesian models with deep learning workflows, allowing for flexible and scalable Bayesian analysis. /TEN 24/

Python is applied to a range of Bayesian analyses in research and industry. Common applications include probabilistic forecasting, parameter estimation, Bayesian optimization, and A/B testing. By using Python's visualization libraries (e.g., Matplotlib, Seaborn), researchers can assess model diagnostics, convergence, and posterior distributions effectively. These tools collectively make Python a go-to language for implementing Bayesian analysis, from simple models to highly complex hierarchical structures.

Python itself is distributed under the Python Software Foundation (PSF) License**,** which is an open-source, permissive license similar to the MIT license. This allows Python to be freely used, modified, and distributed, making it very accessible for both open-source and commercial applications. PyMC is released under the Apache License 2.0. This license is permissive, allowing users to freely use, modify, and distribute the software, even for commercial purposes, as long as proper attribution is included, and liability of the authors is excluded.

In Tab. 3.3 an overview of the three options we considered can be found. Each language has strengths based on context: R is highly suitable for academic and statistical

research. Python is ideal for scalable, machine learning-integrated Bayesian analysis. MATLAB is preferred in engineering and industrial applications, where computational speed and deployment are crucial.

Ultimately, Python was chosen for the project due to its flexibility, extensive available libraries, and suitable licensing options. Additionally, considering future projects, potential extensions and other projects at GRS related to uncertainties, e.g., SUSA development, Python seemed practical enough for long-term scalability and adaptability across various problems. The main package utilised for the study to avoid having to code every single routine, especially if readily available, is PyMC. (more information is provided in section 3.7.1)

Tab. 3.3 Overview of the available tools

	Pros	Cons
R	Extensive statistical packages, such as *rstan* and *brms*, designed for Bayesian modelling. Excellent visualization capabilities with *ggplot2* for posterior diagnostics and model evaluation. Strong support in the statistics community, particularly for academic and research applications. *Stan* integration is highly developed, making it easier to implement complex Bayesian models.	Slower computational performance for large datasets compared to Python and MATLAB. Limited deep learning and large-scale data support for Bayesian workflows. Requires familiarity with R-specific syntax and statistical concepts for efficient use. Coupling with ATHLET is possible, but more complicated than with Python
Python	Rich ecosystem for Bayesian analysis through *PyMC*, *TensorFlow Probability*, and *Stan* (via *CmdStanPy*). Excellent scalability and integration with machine learning libraries for complex or high-dimensional Bayesian models. Strong visualization tools (e.g., Matplotlib, Seaborn) and extensive support for reproducible workflows. Large community and resources for both Bayesian and machine learning applications. Coupling with ATHLET via Python scripting is possible (and reusable for different problems) Used in GRS also for SUSA and MCDET development	Slower than MATLAB for certain matrix-heavy computations, though libraries like *NumPy* and JAX mitigate this. Requires more setup for certain types of models (e.g., hierarchical models) compared to R. Diagnostic and posterior checking tools are not as intuitive as in R.
MATLAB	High performance in numerical computing and efficient handling of matrix operations, beneficial for Bayesian MCMC. Good support for Bayesian optimization and parameter estimation in engineering and applied fields. Powerful visualization tools and diagnostics for model convergence and posterior analysis.	Limited Bayesian-specific libraries; more complex models require additional coding or integration with external tools like *Stan*. Expensive licensing Smaller scientific community for advanced Bayesian analysis compared to R and Python.

	Pros	Cons
	MATLAB Compiler allows deployment of compiled Bayesian models without needing MATLAB licenses for end-users.	

3.7.1 Useful python libraries

PyMC /SAL 16/ is a robust Python package for Bayesian modelling and probabilistic machine learning, offering tools to specify, estimate, and analyse complex Bayesian statistical models. It supports both Markov Chain Monte Carlo (MCMC) and variational inference, allowing for the flexible estimation of posterior distributions in a wide variety of models. PyMC's intuitive syntax and extensive documentation make it accessible for researchers needing advanced Bayesian analyses, while its interoperability with libraries like ArviZ and NumPy further enhances its utility.

NumPy /HAR 20/ is a foundational library for numerical computation in Python, supporting powerful operations on large, multi-dimensional arrays and matrices, which are essential for complex scientific calculations. NumPy includes a wide range of mathematical functions that enable efficient and optimized operations, such as linear algebra, Fourier transforms, and random number generation. As the core library for array computing in Python, NumPy is essential for data manipulation and serves as a building block for many other scientific libraries, such as SciPy, Pandas, and Scikit-Learn. Its efficient handling of array operations has made it indispensable in fields like physics, engineering, data science, and machine learning.

Pandas /MCK 10/ is an open-source Python library that provides high-performance data manipulation and analysis tools. Designed primarily for working with structured data, it offers data structures like DataFrames (2D labelled tables) and Series (1D labelled arrays), which allow for easy data manipulation, including filtering, aggregation, and reshaping. Pandas is widely used for tasks like data cleaning, exploration, and preprocessing in both small and large datasets, and it integrates well with other data science libraries like NumPy, Matplotlib, and SciPy.

Matplotlib /HUN 07/ is a versatile and widely used plotting library in Python, essential for creating a broad range of visualizations, from simple line plots to complex multi-panel figures. It offers both static and dynamic visualization capabilities, making it suitable for both data exploration and publication-quality graphics. Matplotlib's flexibility allows for extensive customization, enabling precise control over plot elements to meet specific scientific requirements. It is a cornerstone of data visualization in Python, commonly used in fields such as bioinformatics, engineering, and social sciences to effectively communicate data insights and research findings.

Seaborn /WAS 21/ is a high-level data visualization library built on top of Matplotlib, specifically designed to make statistical graphics more accessible and informative. By offering a refined set of themes and colour palettes, Seaborn enables the creation of aesthetically appealing plots that facilitate data exploration and interpretation. Its API simplifies complex visualizations, such as categorical plots, violin plots, and pair plots, which are commonly used in statistical data analysis. Seaborn is particularly useful for exploratory data analysis, where visual insights can guide hypothesis formation and further model development.

ArviZ /KUM 19/ is a comprehensive Python library for exploratory and diagnostic analysis of Bayesian models. Specifically designed to work with posterior distributions from probabilistic programming libraries, ArviZ enables researchers to visualize, diagnose, and summarize outputs from Bayesian inference processes. It provides robust tools for posterior predictive checks, trace plots, rank plots, and other visualization techniques that help assess convergence and model fit. By simplifying the analysis of posterior distributions, ArviZ facilitates the interpretation of Bayesian models and improves reproducibility in probabilistic research, making it invaluable for advanced statistical and scientific studies.

PyTensor /PYT 23/ is a Python library dedicated to symbolic math and tensor computation, often used as a backend for probabilistic programming and deep learning frameworks. By supporting automatic differentiation and symbolic graph computation, PyTensor enables complex gradient-based optimization tasks required in machine learning and Bayesian inference. Its efficient computation of mathematical expressions with tensor data structures makes it especially useful for high-dimensional probabilistic models. PyTensor enhances the ability of researchers to construct and optimize models with intricate dependencies and large-scale data, making it valuable for applications that demand computational efficiency and scalability.

OpenPyXL /ERI 10/ is a Python library for reading and writing MS Excel® (xlsx) files, often used for integrating spreadsheet data with Python-based data processing workflows (e.g. experimental data base). It enables programmatic access to cells, formulas, and worksheets, allowing researchers and analysts to automate data entry, extraction, and manipulation tasks in Excel® files.

3.8 Modelling choices

3.8.1 Meta-model vs. full scale ATHLET

Meta-models, also known as surrogate models (e.g., Polynomial Regression Models, Polynomial chaos expansion, Gaussian Process Models, Neural Networks, Kriging, etc.), are simplified representations of complex systems. They approximate the behaviour of a more computationally expensive model (in our case ATHLET) to allow for faster analysis, optimization, and uncertainty quantification. Meta-models are widely used when direct simulations are costly, such as in engineering simulations, climate modelling, and other fields involving large datasets or high computational loads /SIM 01/, /FOR 09/, /RAZ 12/

On the other hand, they have some shortcomings. Meta-models often lack the full accuracy of the original model. This can lead to misleading results, particularly for highly nonlinear or complex systems that meta-models may not capture accurately. Poorly tuned meta-models can either overfit to the training data, failing to generalize to new conditions, or underfit, missing key system behaviours. Meta-models perform best within the range of training data and are unreliable for predictions outside this range (extrapolation), making them less useful for exploring new conditions (case-to-case). Meta-model accuracy often depends heavily on the choice of input parameters and can be sensitive to noise in the data, requiring careful data preprocessing and parameter selection. While meta-models are faster to run, they can require significant initial setup, including generating sufficient training data from the original model and selecting an appropriate meta-model type. Furthermore, it must be set up on a case-to-case basis making it difficult (or impossible) to apply them to another problem. Some complex meta-models, like neural networks, lack transparency, making it difficult to interpret how input variables influence the output.

The ATHLET inputs to be used are quite fast running (reflooding modelling for FEBA, FLECHT, PERICLES), nevertheless it would most certainly benefit from a meta-model in light of the thousands of simulations necessary for Bayesian inference. Furthermore, meta-models require updates to stay accurate if the underlying system changes, which can be time-consuming and similarly computationally expensive. Obviously, ATHLET being under continuous development, the source code of ATHLET is changing regularly, which would then require a (potentially targeted) update of the meta-model, if it is affected (the latter point might not be easy to decide as well). In order to be able to easily apply the method to other problems, without having to make engineering calls and be as

familiar with the modelling approach and its implementation in the code as the developers, it has been decided to not use meta models for this project. At the same time this decision does not reduce the applicability of the developed method for meta-models that might be investigated in the future. It only makes it more robust avoiding the implementation of specific simplifications, that would only be applicable to the problem in question, but not to others.

3.8.2 Priors

One of the main advantages of the Bayesian approach is that with any prior it should (if set up appropriately and enough data is available) converge to the true solution (which always must be checked). Nevertheless, the more informative a prior the quicker it should converge, reducing the computational cost. In setting up the prior distributions any available knowledge can be used, be it based on engineering judgement, sensitivity analysis, frequentist IUQ, etc.

The method development aimed to not use any physical specification of the models being quantified. This approach has been adopted in order to be able to develop an inverse uncertainty quantification method that can be applied flexibly to several problems without restrictions in modelling or its implementation approach. Nevertheless, the inverse uncertainty quantification method not only has to be tested, but during the development process it must be aided with actual examples. In our project, for this purpose the reflooding modelling within ATHLET (see also section 4.1) has been chosen. For the purpose of this development work and study the FEBA /SKO 12/, FLECHT /LOF 80/ and PERICLES tests have been selected.

The selection process of the model parameters, which should be quantified with the newly developed method, is based on sensitivity analysis /SKO 17/ and a respective parameter study /TIB 15/, and engineering judgement. The reflooding model within ATHLET has been intensively investigated in the past in the framework of OECD projects /NEA 17/, /CRÉ 08/ as well as in GRS projects /SKO 17/, /SKO 16/, /TIB 15/.

The following model parameters have been chosen.

Parameters quantified on the basis of the separate effect tests only (better accuracy of quantification cannot be obtained using ‘‘intermediate” tests):

- HTC for forced convection to steam – multiplier
- HTC for film boiling – multiplier

Parameters for which singular phenomena (related to only one measurement) depending only on the selected (quantified) model could be identified:

- Relative velocity in the bundle geometry – pressure drop below swell level
- Water entrained fraction – pressure drop above swell level
- Heat transfer coefficient (HTC) at the quench front – however related to several measurements but it is the only parameter responsible for characteristic cladding temperature bend at the beginning of rapid temperature decrease before quenching

Parameters related to several phenomena:

- number of droplets in the evaporation model (but strong influence on the peak cladding temperature could be observed)

The applied priors are summarized in Tab. 3.4. These have been based on extensive previous work within GRS /SKO 17/, /TIB 15/, /SKO 16/. Since the correction factors for film boiling (OHWFB) and steam forced convection (OHVCF) were previously quantified based on single effect tests, these have been assumed to have a fixed uncertainty distribution, while the (updated) quantification of the others has been the target of the present study. Both fixed u.i.p. have been included in the quantification process, but solely as boundary conditions. This has been decided to avoid associating uncertainties with the parameters in question even though based on the SET quantification their source is known.

The previous quantification of the other parameters has been used as their prior distributions. Furthermore, no correlation between the parameters have been assumed or found. In cases that include influential uncertain input parameters that are strongly correlated ignoring their correlation in the IUQ process would inflate the uncertainty band derived with their help in the FUQ. This, while being more “conservative” in the sense of a potential licensing application, it could cause the results of the FUQ to be not informative enough for applications to large-scale power reactors, considering the additional impact of scaling.

Tab. 3.4 Priors

Parameter		Quantification				Quantification
Name	**description**	**min**	**max**	**ref**	**pdf**	
OHWFB	Correction factor for film boiling	0.65	1.3	1.0	*U()*	KWU tests with 25 rods bundle analyses /VOJ 82/
OHVCF	Correction factor for steam forced convection	0.85	1.25	1.0	*U()*	Literature /GOT 85/ and expert judgement
ODBUN	Correction factor for relative velocity in the bundle geometry	0.64	1.6	1.0	*U()*	dp, T_c in several FEBA experiments
OENBU	Correction factor for entrainment onset velocity	0.6	1.5	1.0	*U()*	dp, m_{co} in several FEBA experiments
CQHTB	HTC at quench front	$2*10^4$	$3*10^5$	10^5	*U()*	Shape of T_c in several FEBA experiments
ZT	Droplets number in evaporation model	10^9	10^{10}	10^9	*U()*	T_c in several FEBA experiments
ZB	Vapour bubble number	10^9	10^{10}	10^9	*U()*	T_c in several FEBA experiments

3.8.3 Quantities of Interest

In our study we define Quantities of Interest (QoI) as the output variables of the simulation code that are directly observed in the experimental test facility or can be directly linked with simple numerical adjustment (practically “extension” of the simulation model). As the IUQ approach here is being applied to the reflooding modelling of ATHLET, the QoIs are related to important aspects of these highly complex phenomena. The QoI are the following:

- Cladding temperature
- Characteristic bend of the cladding temperature to identify the quenching time
- Pressure drop
- Quench front evolution (upper/lower)

Considering all of the above gives a multi-dimensional approach helping to not overfit with compensating errors one or another phenomenon when quantifying uncertain input parameters. Nevertheless, each of the listed QoIs has several values (either in time or

time and elevation) making their application highly complex. Therefore, it has been decided to increasingly consider them in order to adequately test each step of the development process in the IUQ approach, and to make sure it is statistically sound and stable before moving to a higher dimensionality. As one of the most informative parameters is the cladding temperature this has been considered first. The plan was to sequentially add more and more and address any arising issue in the modelling and quantification process.

3.8.4 Complication of the likelihood function

As in this work we decided to not use any meta-modelling, the whole ATHLET simulation is run each time when it is called in the IUQ process. It is way more complicated and complex as opposed to a meta model including a black box, simplified approach linking the uncertain input parameters in question with the output variables (i.e. QoI).

Exact inference for model parameters, or the approximation of the likelihood with an acceptable computational budget is difficult (perhaps even impossible) in this case. This is not a rare case not only in the thermal-hydraulics field, but in Bayesian applications outside of the nuclear field as well. There is an increasing interest in statistical methods for models that are relatively straightforward to simulate but lack tractable transition densities or likelihood functions. In this situation, the general framework is as follows: there is a complex stochastic process $\{X_t\}$, with unknown parameters θ. For any given θ, simulation is possible. The observations are $y = f(\{X_{0:T}\})$. Estimation of θ should be done, but generally the calculation of $p(y|\theta)$ is not feasible, as this would entail integration over the realisations of $\{X_{0:T}\}$. It is an important feature to note, that no assumptions concerning the probabilistic properties of $\{X_{0:T}\}$ or y is made. This is the idea and motivation behind a likelihood free method. The models with the specific inputs are relatively easy to simulate, so can provide several outputs that can be compared to observations and rejected/accepted based on a set of specific criteria.

3.9 Approximate Bayesian Computation method

Likelihood free methods date back to the 1980s /DIG 84/, /RUB 84/. Some more recent examples are, e.g., Indirect inference /GOU 93/, Approximate Bayesian Computation /MAR 11/, bootstrap filter of /GOR 93/ or the Synthetic Likelihoods method of /WOO 10/. A likelihood free method does not mean that likelihood is not part of the analysis, but that it is approximated instead of being derived exactly. One of the main advantages is that

no strict assumptions (including, e.g. unintentional oversimplification) of the models must be made, that often describe exact inference examples. Furthermore, there is no need to assume anything regarding the probabilistic features of the model components.

In our approach we decided to utilise the ABC method mainly due to its intuitive link to our problem formulation and its flexibility in defining the summary statistics. As in our approach we aim to derive the uncertain input parameters based on comparison with experimental data (IET, CET), the ABC approach seem to be a suitable analogue, which can help avoid having to oversimplify our model (no meta-modelling) while also bypassing the derivation of the likelihood (which would be impossible, or extremely costly). Furthermore, its flexible summary statistic handling is expected to be useful for applying our method to different problems with minimal adjustments as well. Additionally, its implementation is somewhat easier due to readily available libraries within PyMC.

Tab. 3.5 Short overview of some likelihood free methods

Method	Description	Pros	Cons
Indirect Inference	Uses an auxiliary model to estimate parameters indirectly by matching simulated summary statistics with observed data.	- Robust for complex models without closed-form likelihoods - Reduces dimensionality of estimation	- Requires careful choice of auxiliary model - May lead to biased estimates if the auxiliary model is not specified correctly
ABC	Avoids likelihood calculation by comparing simulated and observed data based on summary statistics within a tolerance level.	- Useful for models with intractable likelihoods - Flexibility in defining summary statistics	- Sensitive to the choice of summary statistics and tolerance - Computationally intensive with high dimensional data
Bootstrap Filter	Particle-based method for estimating posterior distributions, also known as Sequential Importance Sampling, widely used for dynamic systems.	- Effective for sequential data or time-series models - Can handle non-linear, non-Gaussian models	- Prone to particle depletion - Requires a large number of particles for accurate estimates → computationally intensive
Synthetic Likelihoods	Replaces likelihood with a synthetic likelihood computed based on summary statistics that approximate the likelihood of the observed data.	- Good for complex models with high-dimensional summaries - Avoids direct likelihood computation	- Sensitive to the choice of summary statistics - Computationally expensive due to simulation requirements for each likelihood estimate

In general, the ABC approach consists of 3 steps (Fig. 3.7):

- Sampling of θ^* from $p(\theta)$
- Simulation of a data set y* using θ^*
  → return the simulated set of QoI (the same dimension as the observed data y)
- Comparison of the simulated y* with the experimental data y using a distance function Δ and a tolerance threshold ϵ

As a result, a sample from the parameter distribution $p(\theta|\, \Delta(y, y^*) \leq \epsilon)$ is obtained. If ϵ is sufficiently small $p(\theta|\, \Delta(y, y^*)) \approx p(\theta|\, y)$.

Sequential Monte Carlo (SMC) methods are a class of algorithms used to approximate probability distributions through a series of weighted samples, called particles. In each step, particles are propagated forward using a proposal distribution, re-weighted to reflect their likelihood under the target distribution and resampled to maintain particle diversity.

Sequential Monte Carlo ABC iteratively transforms the prior into the posterior by propagating the sampled parameters through different proposed distributions $\Pi(\theta_i)$ and weighing ($\omega_i \propto \frac{p(\theta_i)}{\Pi(\theta_i)}$) the accepted ones ($\theta_i$). It combines the advantages of traditional Sequential Monte Carlo methods - such as the ability to sample from distributions with multiple peaks - while eliminating the need to evaluate the likelihood function.

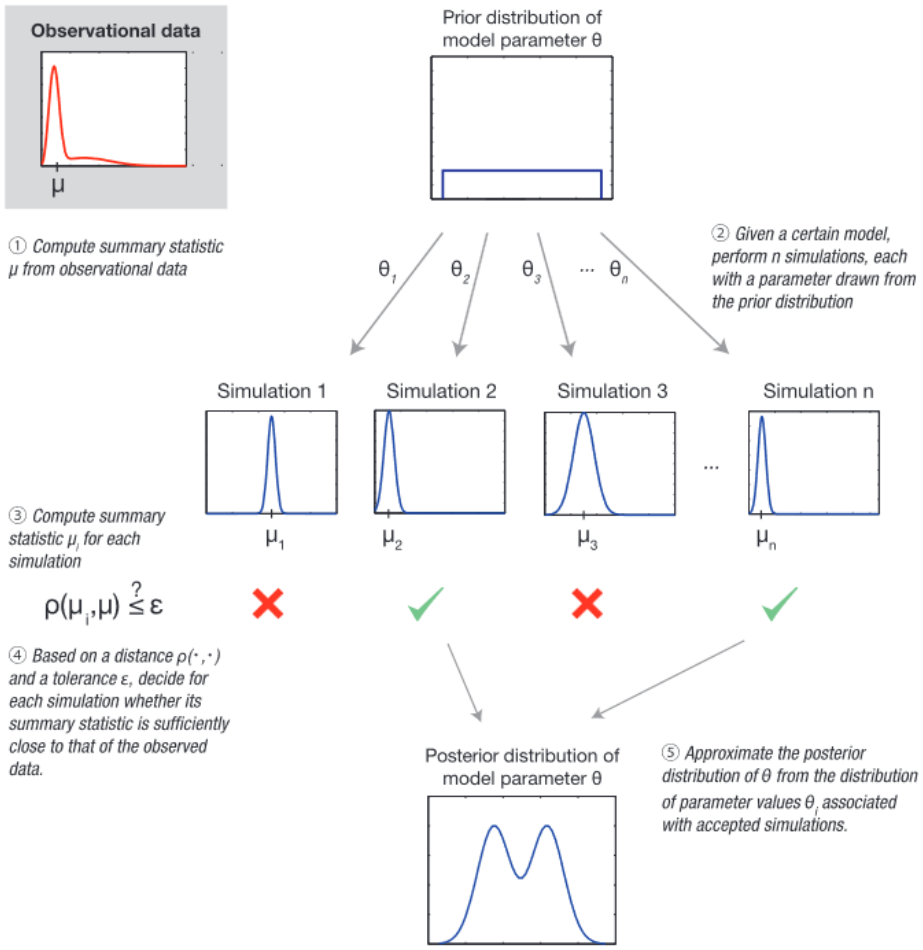

Fig. 3.7 Overview of parameter estimation using the ABC approach /SUN 13/

3.10 ABC method implementation for the reflooding modelling in ATHLET

The implementation of the Approximate Bayesian Computation method for this study was approached with a structured and incremental strategy. Initially, a phased approach was adopted: starting with a basic test case involving a single point value, followed by cases with multiple values, and progressively extending to multiple time series. This allowed for a systematic expansion of the observational data base to accommodate more complex scenarios and address any arising issue. This approach made it easier to identify the sources of problems and deal with them without introducing compensating errors in the methodology.

Where feasible, established software packages for certain functions were utilized to streamline development, while keeping in mind potential licensing restrictions where applicable.

As the basis for the development and quantification process the FEBA216 was chosen as it offered a robust basis for the analysis. The first variable selected was cladding temperature due to its high informativeness, capturing critical data on quench front progression, quenching time, and the maximum cladding temperature. This variable provides a comprehensive insight into the thermal behaviour during the reflooding process.

The experimental error is postulated as $N(0,\sigma)$, while the model bias is not explicitly addressed in the present analysis. As the simulator is run multiple times and creates several output values based on a set of varied input parameters, this has been considered as a sufficient first approach to take into account the variability of the output variables. Nevertheless, it worth to investigate possible overfitting caused by this in the future, and if necessary, addressing it.

3.11 Implementation

During the development work the implementation has been kept as modular as possible. This enables easier future maintenance as well as extension of the approach. Furthermore, it makes it easy to adopt the method to other problems, by only having to modify specific modules addressing the physical problem definition (mainly simulation set-up and observation data) while leaving most of the routines (especially the statistical ones) unchanged. This enables the user to focus on the problem definition and actual evaluation process, which can be aided by guidelines, but at the moment cannot be done without some degree of expert knowledge (see e.g. SAPIUM guidelines /NEA 18/).

Moreover, several of the routines created can be utilised in forward uncertainty analysis, as well as outside of the UQ domain. This has been intentionally done in order to make future applications easier, help and aid validation as well as CI/CD processes.

In the following some routines are described.

3.11.1 Experimental data base

Handling the experimental data base and creating the set of observation data including the experimental error is done in two dedicated routines. It is necessary to create a draw of the observation data in order to avoid overfitting, taking experimental error into account and to avoid the collapsing of the pdf to a single value point. This experimental data set

is then used as the basis for comparison with the simulation results in deriving the distance metric used for the rejection process in the ABC method.

To create such an observation vector/matrix, experimental data points and their respective experimental error are read in. The experimental error has been assumed to be normally distributed with $N(0,\sigma)$, i.e. successful calibration postulating there is no systematic bias and only aleatory uncertainty of the observation. The derived observational data set includes a 3σ (corresponding to the experimental error) coverage including 99.7% of the possible realisations. Depending on the selected observation vector/matrix size (SE) it can include several observational points for each measurement point.

Definition of the observation vector size is crucial. The higher its dimension (including multiple experimental points, physical properties, time series, elevations, etc.), as well as for each dimension a higher number of observational data points is created, the more significant the distance function becomes, yielding better-quality approximation of the posterior. At the same time, due to the MCMC convergence criteria as well as due to the accepting/rejecting procedure, the computational need grows exponentially. Some examples of the generated observational data for FEBA216 is shown in Fig. 3.8, Fig. 3.9 and Fig. 3.10.

Summary

Draws from the experimental data base is generated in order to take experimental error into account. Zero offset (successful calibration) with 3σ coverage is assumed.

Input:

- Name and location of the experimental files (csv, xlsx format)
- Indices/Names of the observational set up (T, dp, t, etc.)
- Size of the observational vector/matrix to be created at each data point (SE)

Output:

- Observational data base formatted to ABC applications
- Visualisation of the observational data base

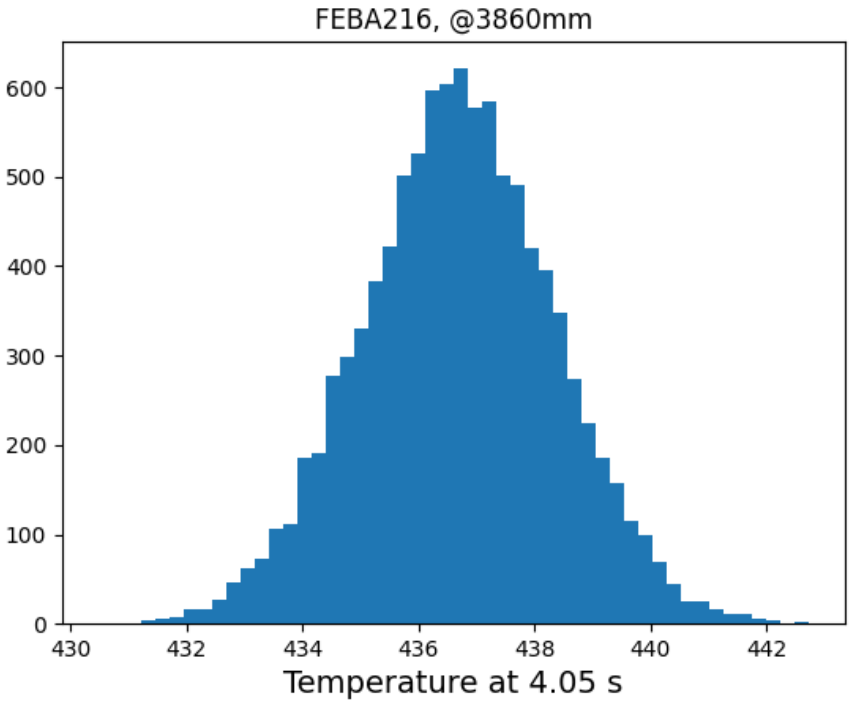

Fig. 3.8 Example of an observational data set with SE = 10 000

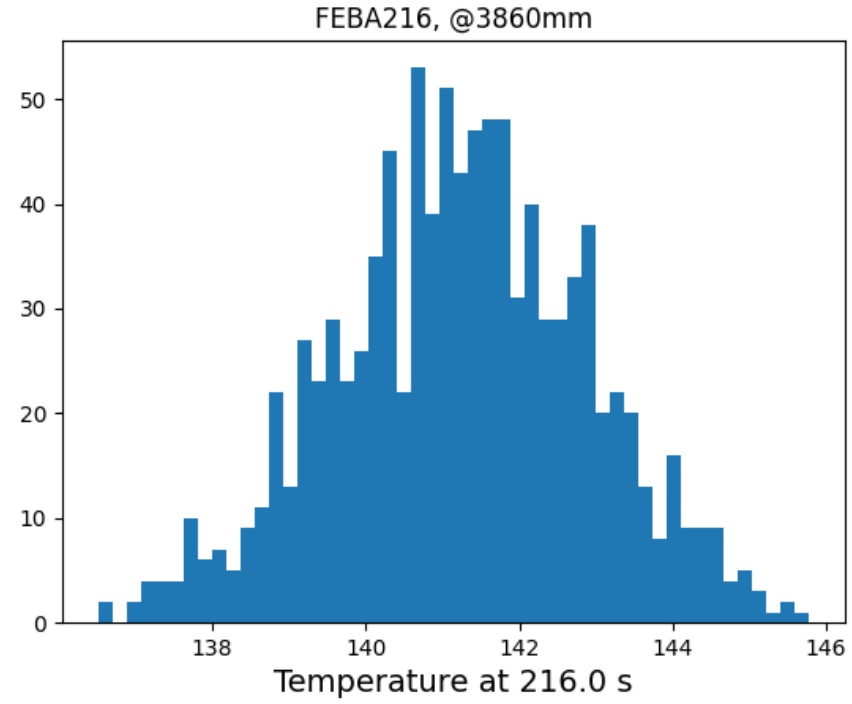

Fig. 3.9 Example of an observational data set with SE = 1000

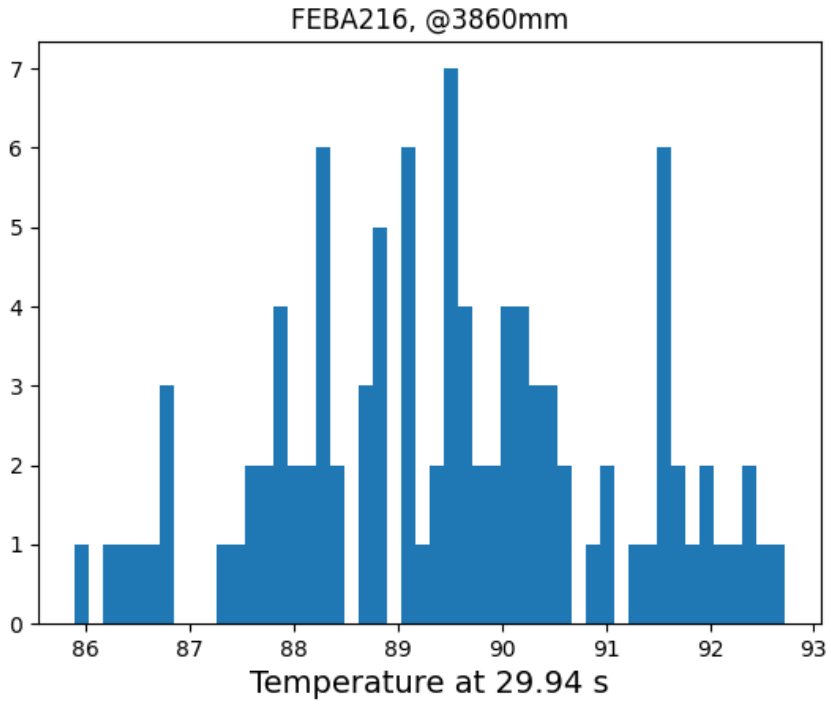

Fig. 3.10 Example of an observational data for with SE = 100

3.11.2 Simulator

In order to simulate the data with the ATHLET code, a so-called simulator has been created, that could be implemented in the ABC modelling with the help of PyMC. This has been modularised to address each phase of the simulation process (preparation, running the simulation, evaluation) and their subprocesses (input creation, queuing, checking for finished simulations, data collection, data extraction, formatting, etc.). Here some of the main features are summarised.

Input creation

For the implemented routine, a manually created input data set is necessary. This should cause no problems, since for any application, be it IUQ or FUQ, a best estimate case has to be defined and without an input deck there is no ATHLET analysis. This input deck is adapted slightly by placing appropriate place holders for any model settings or inputs to be varied under the CW `PARAMETERS` section (recommended option for easier debugging, but the created routine can replace the place holders anywhere in the input template). However the choice of the place holder can be easily adjusted by the user, the only restriction is that each input parameter must be uniquely identifiable.

As a result, an input file, with specified naming convention (easily adjustable by the user) is created. Within each input deck the place holders for the uncertain input parameters are replaced with a realisation provided by the ABC sampling.

Note: In our case there are two parameters in the reflooding modelling (OHWFB, OHVFC see chapter 3.8.2) for which the uncertain distributions are kept fixed. They have been successfully quantified based on SET therefore, not our aim to further quantify them. Nevertheless, they have been included in this study to account for the impact of their uncertainty on the reflooding modelling, thus avoiding associating uncertainties caused by known sources to our quantification process. Therefore, another short routine has been created to appropriately sample these parameters given their respective pdfs as listed in Tab. 3.4.

Running the ATHLET simulations

All input decks are copied to separate folders. This is not a necessary step for ATHLET, but when moving to AC^2 applications (coupled ATHLET-CD-COCOSYS, or simple COCOSYS) it is, since COCOSYS does not let multiple simulations be carried out in the same folder. The runs are started via shell scripts, that are flexibly written to start any kind of AC^2 (stand-alone as well as coupled) simulation on the GRS HPC server. The job IDs are saved and used for periodically checking if the given instance is finished, before moving to the next one.

Evaluation of the simulation data

There are several routines tailored for various data types, ranging from simple single-value formats (primarily used for testing) to more complex data structures with multiple time-dependent variables. The evaluation process has been based on AC^2's new output option using a HDF5 format, which is expected to become the standard. This planned transition to HDF5 and its straightforward integration with Python made it the logical choice.

Data extraction occurs at pre-defined designated time points (aligned with the observational data set) from the appropriate objects. Necessary adjustments for future application (other experimental data set, different model or even different code, etc.) include defining the series of time points and variable information based on the hierarchical structure realized in the hdf5 files: Object name > Model name > Index > Variable name (e.g. *['QU-CORE'] ['QUENCH'] ['NODE1'] ['ZQBOT'])*. Finally, the results are formatted into a

vector/matrix compatible with the set of observational data for the ABC method comparison.

As thousands (usually tens of thousands) of simulations are needed to reach convergence, each of the run has been removed after the necessary data has been extracted in order to save on storage capacity and to avoid having to handle TBs of data.

Summary

The simulator runs the necessary number of calculations to match the created observational data base. It automatically evaluates the results and provides the necessary matrices for the distance comparison within ABC.

Input:

- Size (same as SE in experimental data base, see Chapter 3.11.1) and extent of the observational data base
- Input template name and location
- Shell script name/location for queuing the jobs

Output:

- Matrix of simulation results matching the observational data form

3.11.3 ABC model

The ABC approach is implemented with the help of PyMC. This offers some flexibility and saves time, but during the analysis some problems have been also encountered, especially when dealing with multiple time series inputs. Therefore, in the future, if resources allow, an independent, more tailored implementation using only basic numerical packages (NumPy, Pandas, PyTensor, etc.) would be useful to address these issues. Furthermore, it would provide the flexibility to implement specific distance metrics that could be more appropriate for our applications than the standard ones.

Choice of summary statistics is crucial for the ABC analysis. The basic recommendation is to apply sorted data lists, which compares sorted values, but might miss detailed distribution characteristics /FEA 12/. For comparing basic data features, descriptive statistics (mean, median, variance, skewness, etc.) are the most suited. Empirical distribution

functions (e.g. CDF) are ideal for capturing the overall shape of the distributions. For comparing frequency distributions, histogram binning could be useful. In case of analysing high-dimensional problems, Principal Component Analysis (PCA) could be useful.

In our study as a first step sorted analysis has been planned and carried out, due to its simplicity and lower CPU demand compared to other metrics. As a next step, descriptive statistics should have been looked at, but this was not possible due to the constraints for the present analysis. In the future, however, this is an important point to be further investigated. Establishing a guideline for different application types would be beneficial for potential users but needs to be investigated in a follow-up project.

The distance metric chosen is also influential. It determines how the similarity between the simulated data and the observed data is measured. Some commonly used distance measurements are listed below.

Euclidean Distance (3.7) is the straight-line distance between two points in multidimensional space. It's widely used when data is continuous, and each dimension is equally important.

$$d(x,y) = \sqrt{\Sigma_{i=1}^{n}(x_i - y_i)^2} \tag{3.7}$$

Manhattan Distance (L1 Norm) (3.8), also called "city block distance" measures the absolute differences along each dimension. Useful for high-dimensional data (dimensions being independent), where individual dimensions might have varying importance. It's robust to outliers, as it doesn't square differences like the Euclidean distance.

$$d(x,y) = \Sigma_{i=1}^{n}|x_i - y_i| \tag{3.8}$$

Mahalanobis Distance (3.9) accounts for correlations between variables, which is useful when data dimensions are not independent. It is a good fit for multivariate, correlated data and normalizes differences by the covariance structure

$$\mathrm{d}(x,y) = \sqrt{(x-y)^T \Sigma_{cov}^{-1} (x-y)} \tag{3.9}$$

Kullback-Leibler (KL) Divergence (3.10) measures the difference between two probability distributions. It can measure how well a simulated distribution aligns with observed patterns.

$$D_{KL}(P||Q) = \Sigma P(x)\log\left(\frac{P(x)}{Q(x)}\right) \quad (3.10)$$

Wasserstein (Earth Mover's) Distance measures the minimal "effort" required to transform one distribution into another. It is based on the concept of moving "mass" in probability distributions, often solved using optimal transport theory.

Maximum Mean Discrepancy (MMD) (3.11) is a kernel-based metric for comparing two distributions, commonly used in machine learning. Useful for comparing complex distributions without requiring parametric assumptions. Depends on a kernel function k(x,y):

$$\mathrm{MMD}(P,Q) = \mathrm{E}_P[k(x,x`)] + \mathrm{E}_Q[k(y,y`)] - 2\mathrm{E}_{P,Q}[k(x,y)] \quad (3.11)$$

Cosine Similarity (3.12) measures the angle between two vectors, treating each data point as a vector in a multi-dimensional space. Useful in high-dimensional data applications, especially where the magnitude of the values isn't as important as their direction or pattern.

$$\text{Cosine similarity} = \frac{x \cdot y}{||x||\ ||y||} \quad (3.12)$$

Gaussian distance metric is derived from the Gaussian (normal) distribution and is often used in ABC to facilitate smoother sampling and more realistic posterior distributions. It is especially useful in cases where there is noise or uncertainty in the data, as it assigns higher weights to closer values and gradually reduces the weight for more distant values, rather than imposing a hard cut-off. Unlike other metrics that simply count all distances below a certain threshold, the Gaussian distance metric provides a continuous, smooth weighting for each parameter sample. Instead of discarding simulations based on a fixed tolerance threshold, the Gaussian metric lets each sample contribute to the posterior estimation, with a weight based on its distance from the observed data (3.13). (No discarding of data, which is particularly useful for expensive simulations.)

$$\omega = \exp\left(-\frac{d^2}{2\sigma^2}\right) \tag{3.13}$$

- d is the Euclidean distance between the observed and simulated data.
- σ is a scaling parameter that determines the spread of the Gaussian function, controlling how quickly the weights decay as distance increases.

As a shortcoming its computational demand can be high in large datasets due to calculating the Gaussian weight for each sample.

As a first approach the Gaussian distance is applied due to its smoothness and utilization of all simulated data.

In the future different distance function could be investigated. Establishing a guideline for different application types would be beneficial for future applications.

Post processing

In PyMC, NetCDF (.nc) files are commonly used to save the results of a sampling process, including Approximate Bayesian Computation (ABC) results, which may require a large number of simulations. NetCDF is a flexible and efficient file format that allows storage of multidimensional data, making it well-suited for the type of hierarchical data often generated in Bayesian inference.

It includes the posterior samples, metadata, and configuration of the sampling process. This allows the storage of the results for later post processing without needing to rerun the full computation, which is extremely resource-intensive in our case. It allows users to further inspect and analyse their results without having to define every output analysis set ahead of running the analysis. This persistence can be crucial in long-running ABC processes where intermediate results need to be stored. Furthermore, it can be directly loaded into ArviZ, a Python library for exploratory analysis of Bayesian models. ArviZ can visualize and perform diagnostics on data saved in NetCDF format, making it easier to interpret the ABC posterior distributions and model diagnostics.

The general evaluation in this study included:

- *Autocorrelation Data* measures the correlation between samples at different time lags in the posterior distribution. High autocorrelation suggests that the samples are not independent, indicating slow mixing of the sampling process, which can be improved with larger sample sizes or adjustments of the algorithm.
- *Rank plots* help assess the convergence and quality of posterior samples. They show the ranks of parameter estimates across different chains, allowing users to visually detect any anomalies or biases in the sampling.
- *Forest Diagrams* visualize the estimated values of parameters along with their uncertainty intervals. They are useful for comparing multiple parameters or models, allowing for a quick comparison of credible intervals.
- *Violin Plots* show the distribution of posterior samples for each parameter, combining aspects of box plots and kernel density estimates. They provide a clear view of the spread and shape of the posterior distribution, making it easier to interprete parameter uncertainties.
- *Posterior PDF* (Histogram, Fitted Continuous): Main output of the analysis.

Additionally, some analyses have been directly performed including:

- Summary statistics including mean, standard deviation, highest density intervals: hdi 3%, hdi 95%, Monte Carlo standard error (mcse) mean, mcse standard deviation, Effective sample size (ESS) -bulk, ESS-tail, Rhat
- Posterior pdf (histogram, fitted continuous) for each quantified uncertain input parameters
- Trace plots

3.11.4 List of Python packages utilised in the development work

- ArviZ: A library for exploratory analysis of Bayesian models, including visualization and diagnostics for model comparison and evaluation. /KUM 19/
- Matplotlib: A comprehensive library for creating static, animated, and interactive visualizations in Python /HUN 07/.
- PyMC: A probabilistic programming library that makes Bayesian inference accessible using a user-friendly, high-level interface /SAL 16/.

- OS: A built-in Python library providing a way to use operating system-dependent functionality, like file and directory manipulation /PYT 23/.
- Time: A built-in Python module for handling time-related tasks, including time tracking and time-based operations /PYT 23/.
- Shutil: A standard library for high-level file operations, such as copying and archiving /PYT 23/.
- NumPy: A fundamental package for scientific computing in Python, supporting large multi-dimensional arrays and matrices, along with a collection of mathematical functions /HAR 20/.
- logging: A standard library module providing a flexible framework for logging messages from Python programs, useful for tracking events during code execution /PYT 23/.
- Seaborn: A data visualization library based on Matplotlib that provides a high-level interface for drawing attractive and informative statistical graphics /WAS 21/.
- pandas: A powerful data analysis and manipulation library for Python, offering data structures like DataFrames and tools for handling various data formats /MCK 10/.
- h5py: A Pythonic interface to the HDF5 binary data format, often used for storing large amounts of data, making it easy to handle large datasets in Python /AND 13/.
- SciPy: A library that builds on NumPy, providing a collection of functions for scientific and technical computing, including modules for optimization, integration, and statistics /ERI 01/.

3.12 Example and Gaussian function

To test all routines responsible for the statistical framework the simulator has been replaced with a Gaussian function. The target function has been $\mathcal{G}(10,2)$. The priors for the location and scale parameters have been set to μ(0,5) and σ (1), the latter being half normal (scale cannot be negative).This has been done to check the implemented statistical framework, the effect of chain and draw numbers, as well as to get familiar with the most common evaluation plots. The results obtained with 1000 draws and 4 chains are summarised in Tab. 3.6 can be seen in Fig. 3.11 to Fig. 3.19., while in case of 10 000

draws and 4 chains Tab. 3.7 summarizes the derived statistics and Fig. 3.20 to Fig. 3.28 depicts the results visually.

In both cases according to the hdi intervals the derived mean values are significantly different from zero and provide a good mean/sd relation. The Rhat values indicate basically no further room for potential scale reduction, which is good (Rhat should be < 1.05 or even better 1.01). Generally, an effective sample size of 10 000 is considered good enough (Kurschke), but here only a quick running sample has been created, therefore such low obtained ESS. An effective way to judge the ESS is to compare it to the total sample size (TSS): ESS/TSS values below 0.1 most likely indicate some problems that should be looked into. Otherwise, it says more about the effectiveness of the MCMC calculation as opposed to its convergence or quality. Naturally the lower the ESS, the more simulation points are not taken into account, slowing down the procedure as well as increasing the computational cost. In both cases the initially higher autocorrelation generally becomes lower, though with a higher sample size (as expected) it is more consistent. In case of the lower sample size the trace plots show a trend that should not be present, while with D = 10000 no trend can be observed underlying the better convergence with the higher sample size. Furthermore, the shapes of the derived values raise questions in case of the lower sample size. Similar conclusions can be drawn when looking at the ranks. The forest diagram as well as the violin diagrams are alternative representation of the derived values.

Tab. 3.6 Results of the ABC simulation with D=1000 and C=4

	mean	sd	hdi3%	hdi97%	mcse mean	mcse sd	ESS bulk	ESS tail	Rhat
μ	9.984	0.065	9.865	10.105	0.001	0.001	1961	1500	1
σ	1.954	0.054	1.847	2.046	0.001	0.001	1583	1625	1

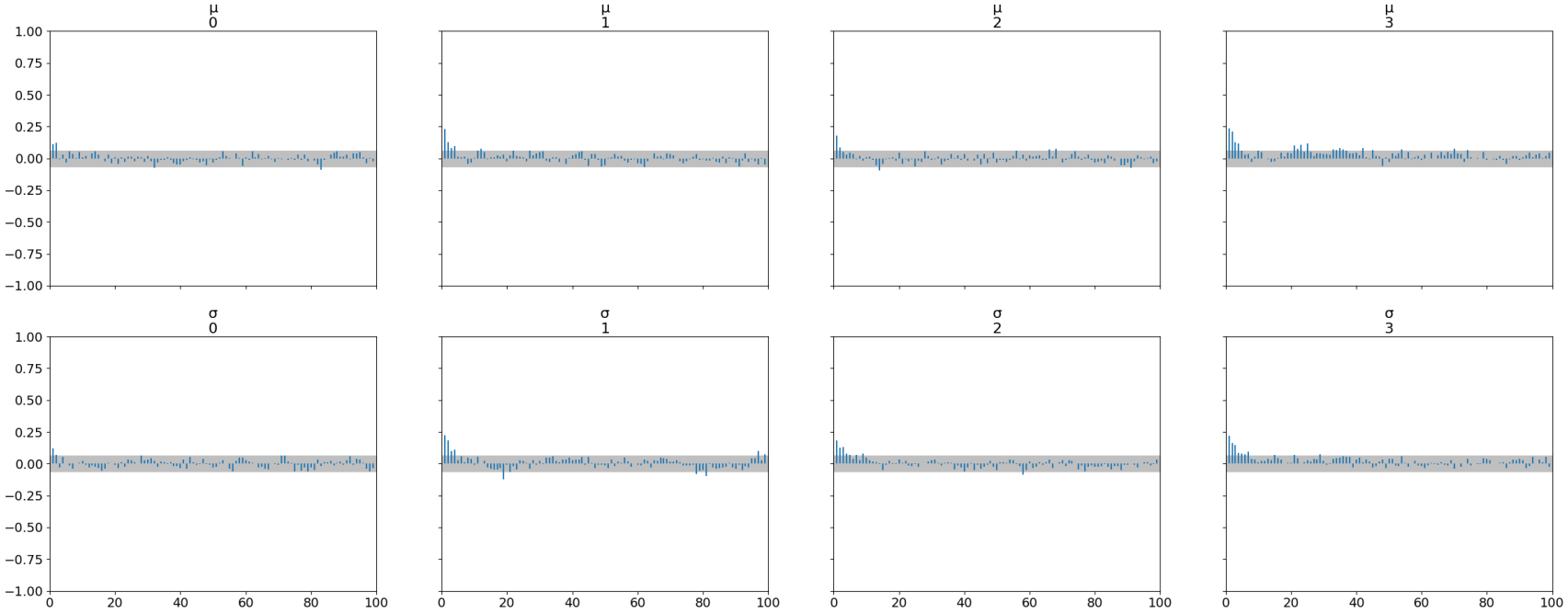

Fig. 3.11 Autocorrelation for D=1000 & C=4

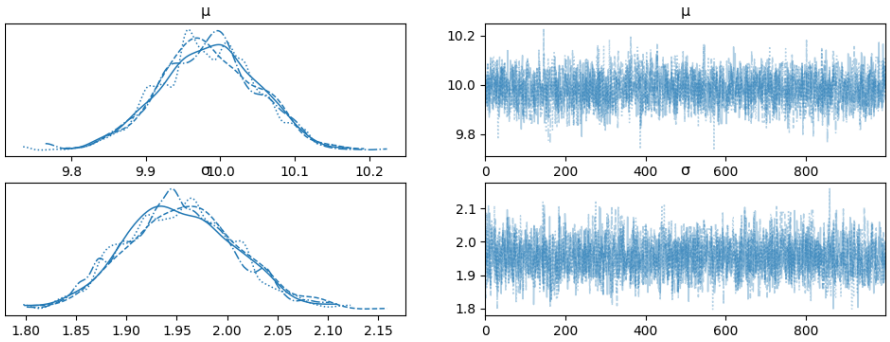

Fig. 3.12 Posterior and trace for D=1000 & C=4

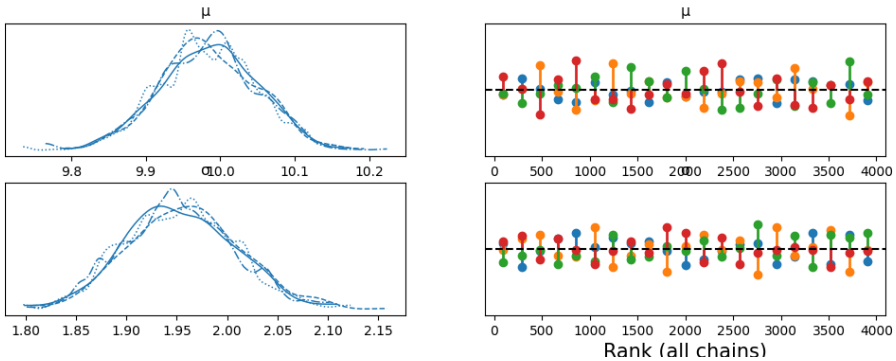

Fig. 3.13 Posterior and ranks for D=1000 & C=4

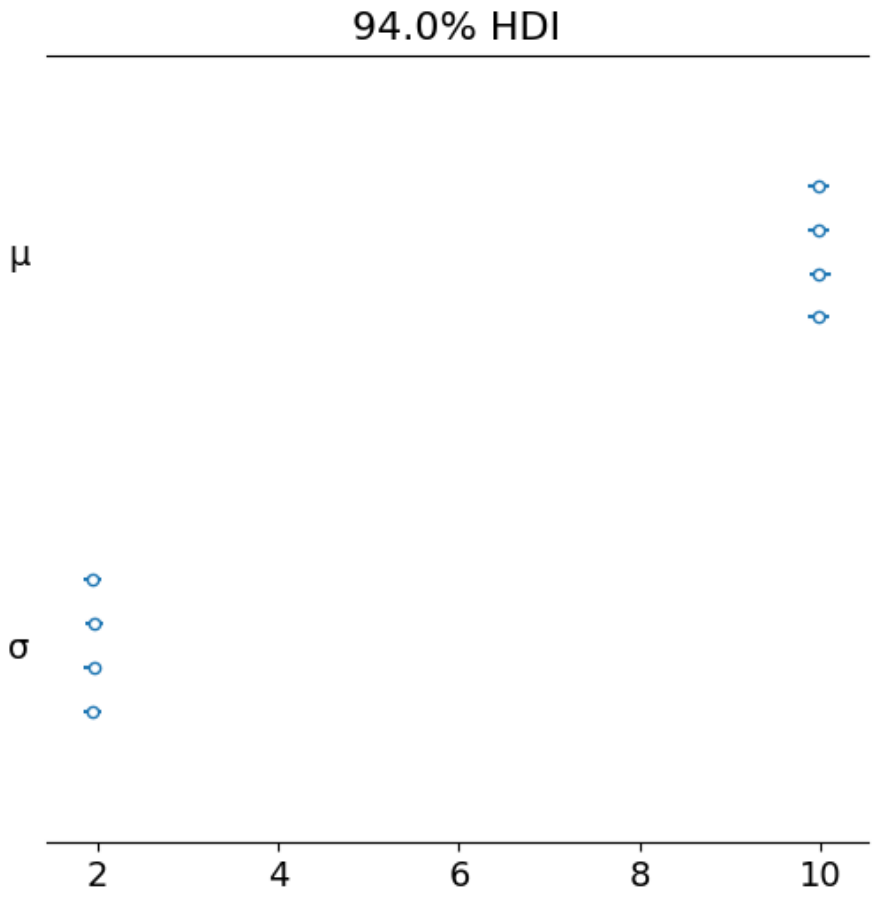

Fig. 3.14 Forest diagram for D=1000 & C=4

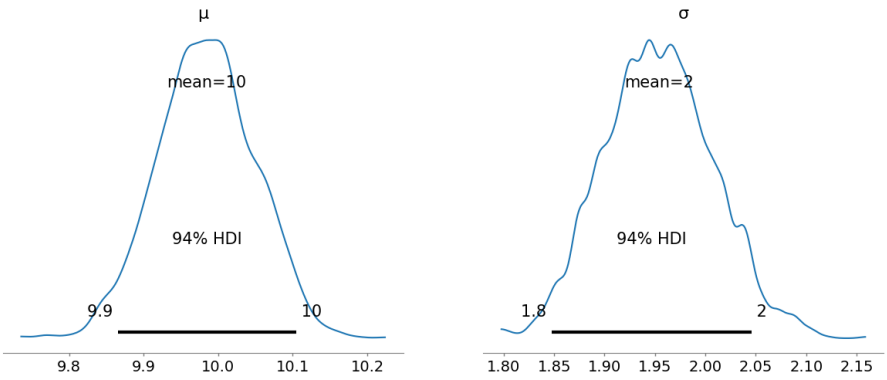

Fig. 3.15 Posterior and HDI 94% (CDI) for D=1000 & C=4

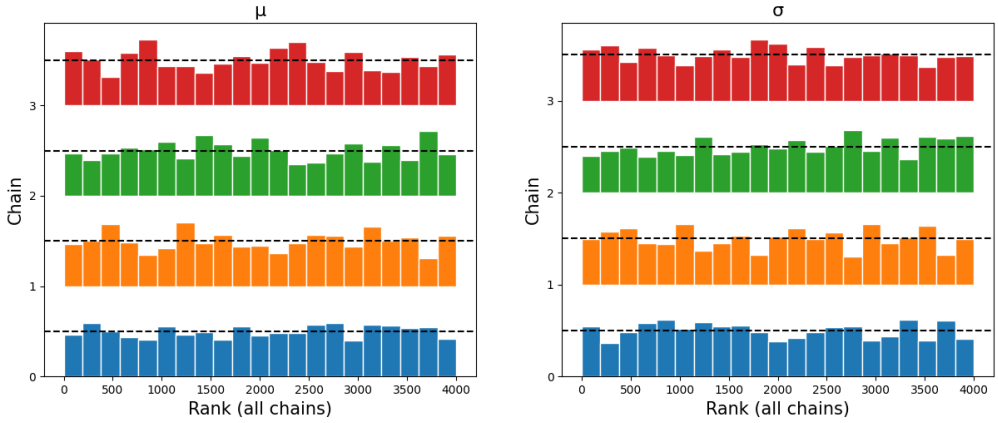

Fig. 3.16 Ranks for D=1000 & C=4

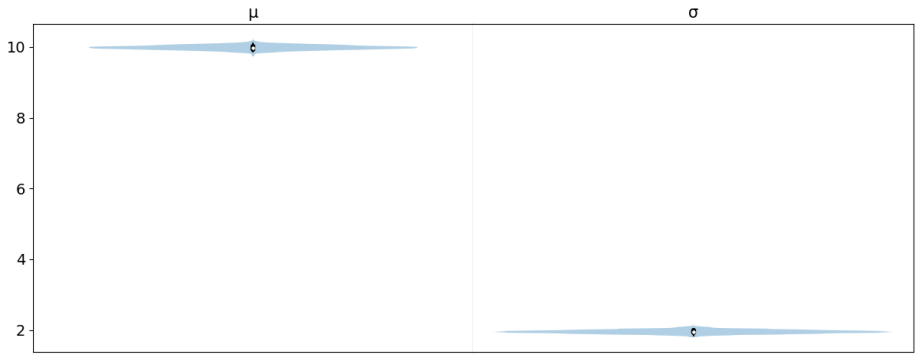

Fig. 3.17 Violin diagram for D=1000 & C=4

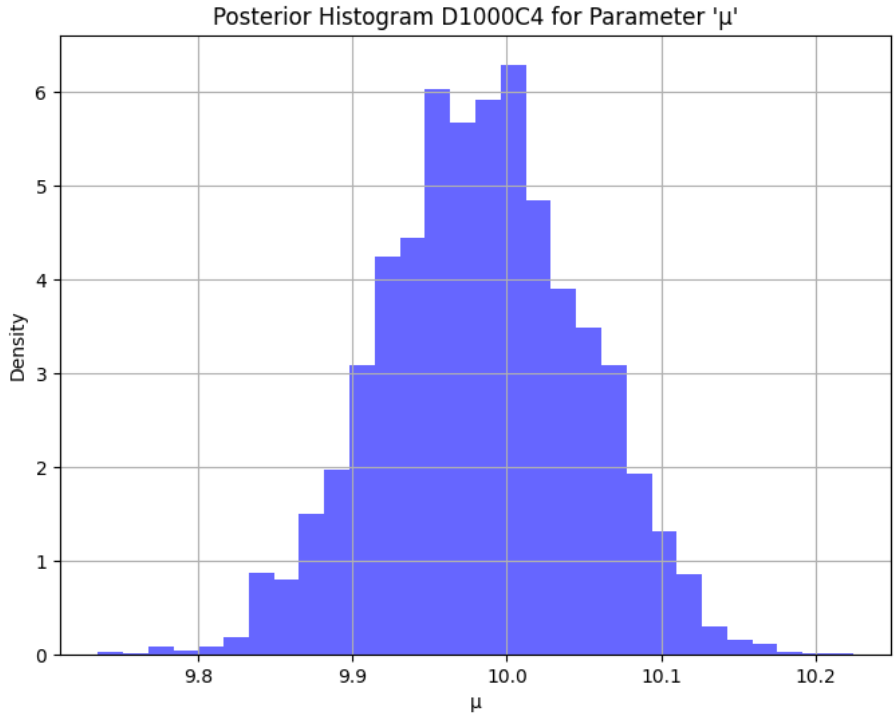

Fig. 3.18 Posterior for location parameter for D=1000 & C=4

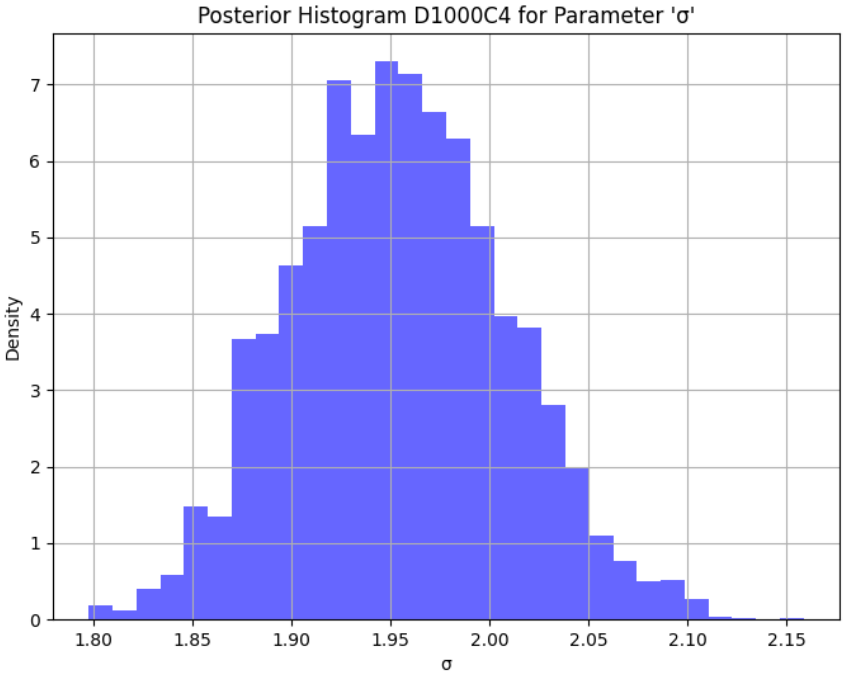

Fig. 3.19 Posterior pdf for scale parameter for D=1000 & C=4

Tab. 3.7 Results of the ABC simulation with D=10 000 and C=4

	mean	sd	hdi3%	hdi97%	mcse mean	mcse sd	ESS bulk	ESS tail	Rhat
μ	9.942	0.069	9.815	10.073	0.001	0.0	16724	12147	1
σ	1.928	0.053	1.824	2.024	0.0	0.0	21403	18160	1

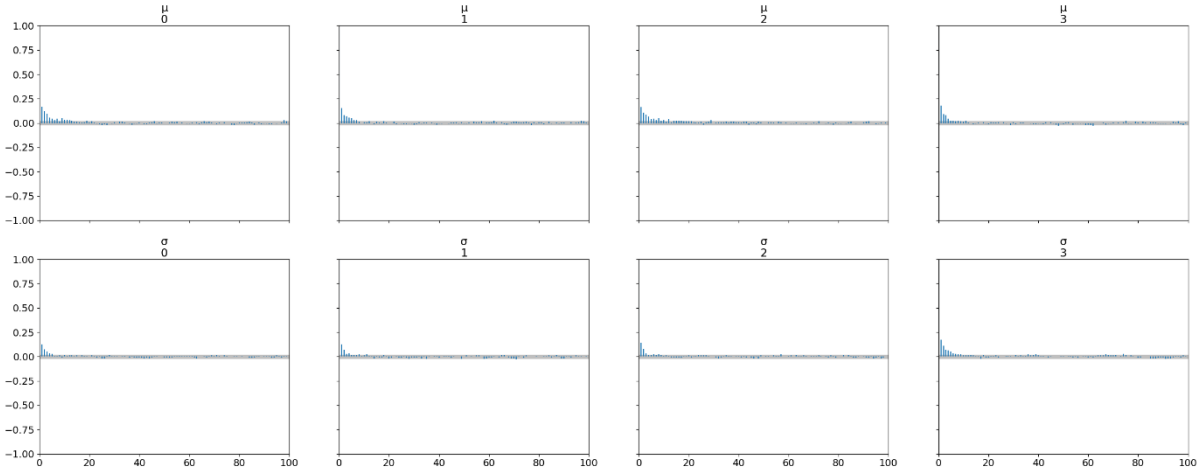

Fig. 3.20 Autocorrelation for D=10 000 & C=4

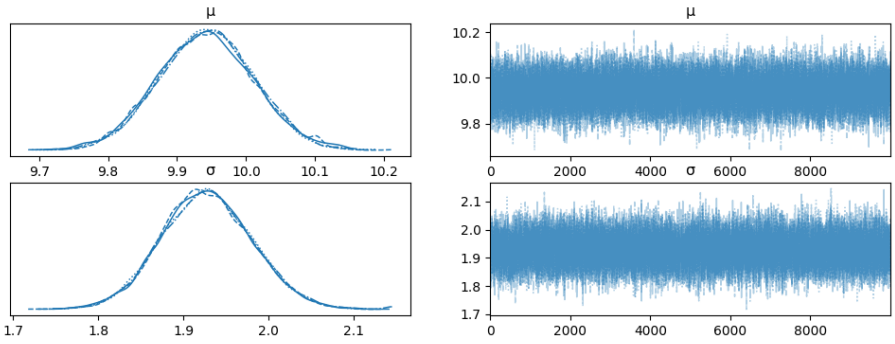

Fig. 3.21 Posterior and trace for D=10 000 & C=4

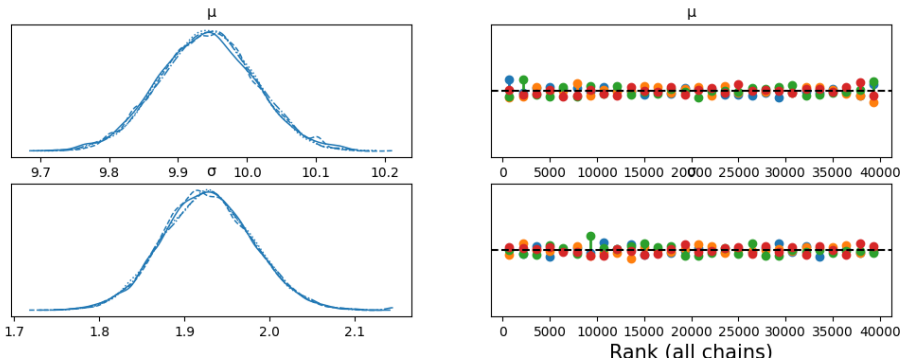

Fig. 3.22 Posterior and rank for D=10 000 & C=4

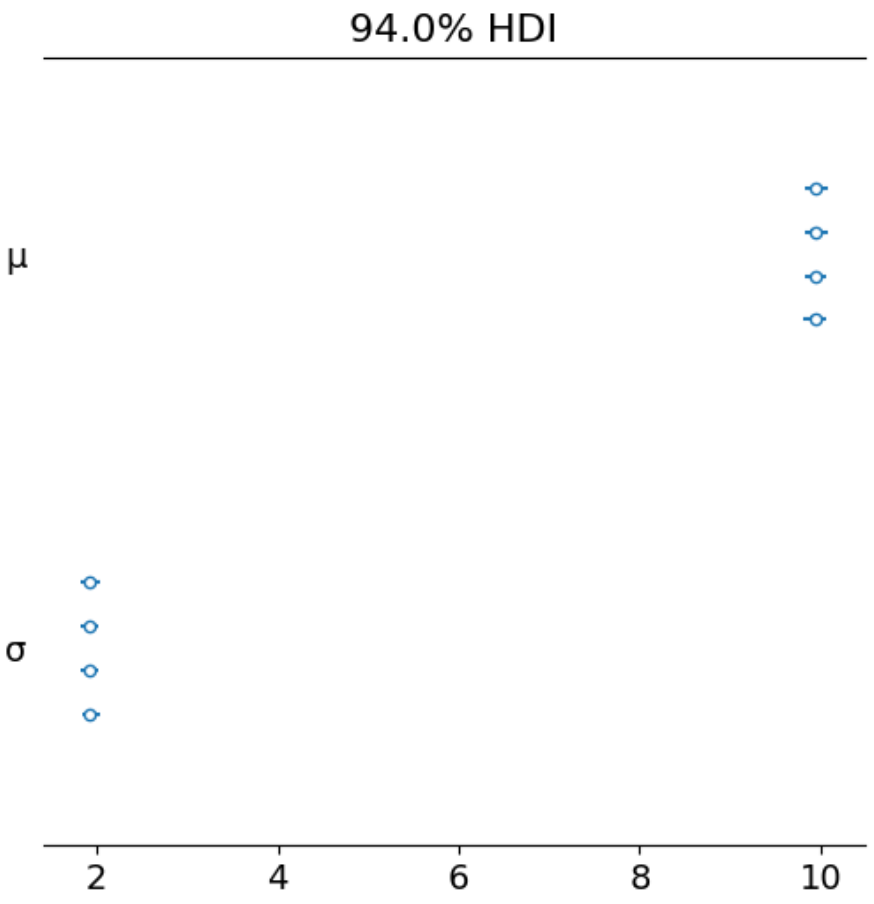

Fig. 3.23 Forest diagram for D=10 000 & C=4

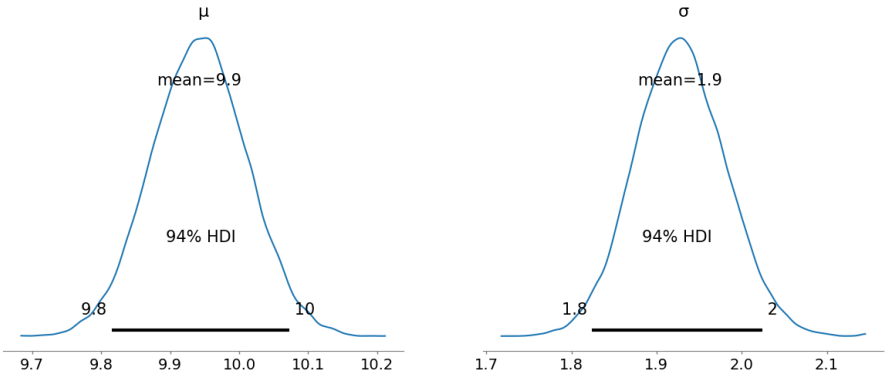

Fig. 3.24 Posterior and CDI 94% for D=10 000 & C=4

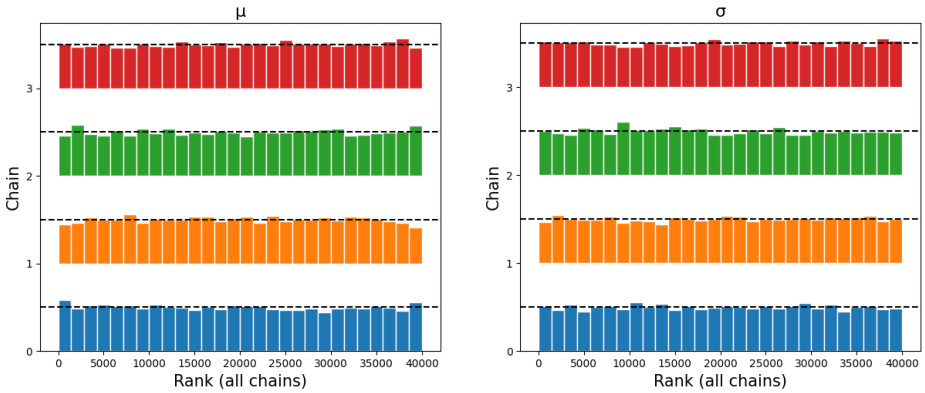

Fig. 3.25 Ranks for D=10 000 & C=4

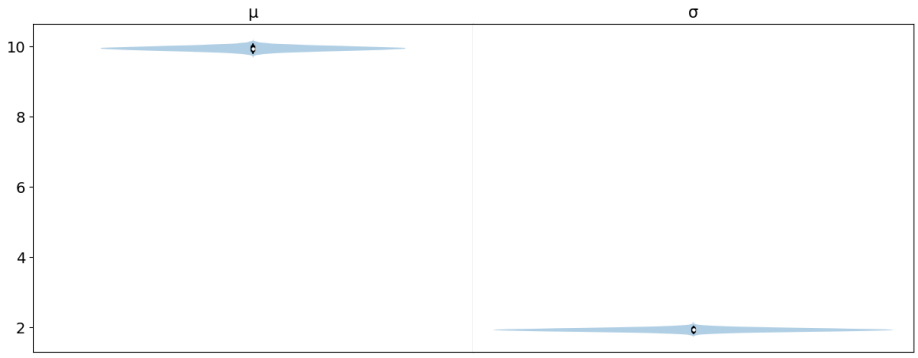

Fig. 3.26 Violin diagram for D=10 000 & C=4

As the results show the higher the number of draws, the betterthe quality of the results. Not only does it depict better convergence features, but also the derived results are

closer to the theoretical ones. This shows that even with such an easy example having chain sizes in the tens of thousands is necessary.

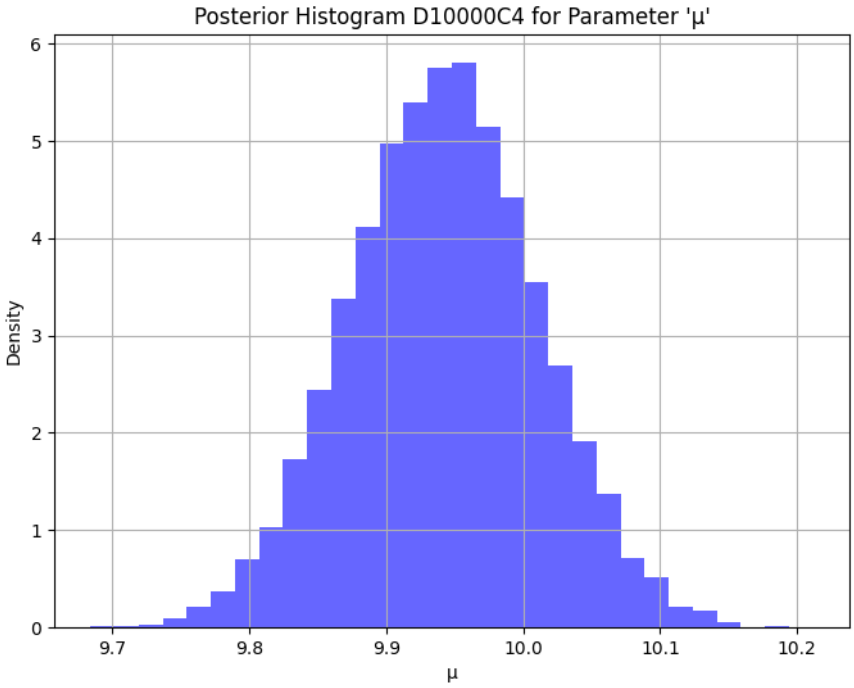

Fig. 3.27 Posterior of the location parameter with D=10 000 & C=4

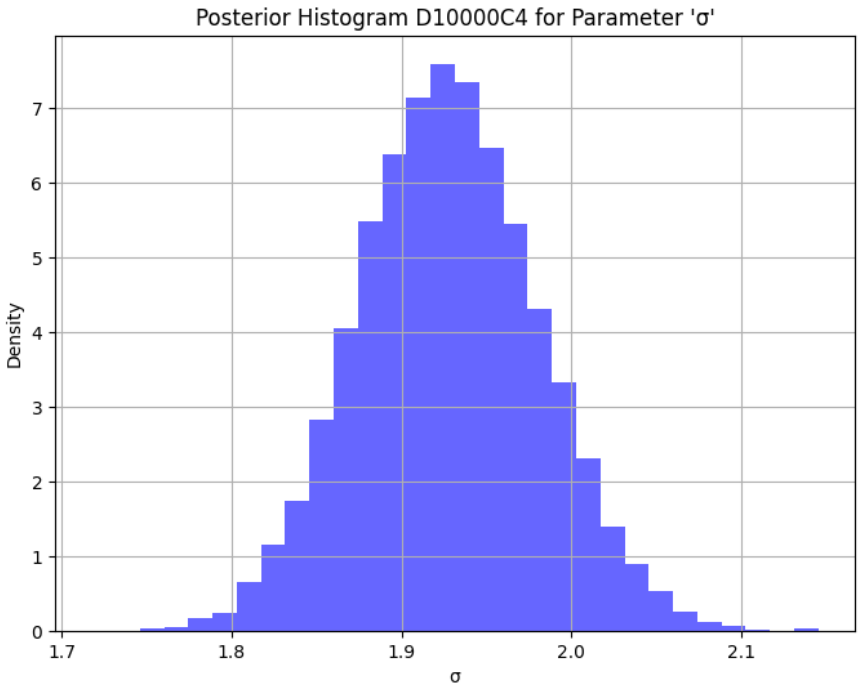

Fig. 3.28 Posterior of the scale parameter with D=10 000 & C=4

3.13 Verification and Validation

Validation of the derived uncertain input parameters are carried out by applying them in a forward uncertainty analysis and comparing the results with the observations. In order to check their validity a “simple coverage” (i.e. enveloping the experimental values) has been considered. At the same time informativeness is kept in mind. The derived uncertainty limits should encompass the observation values, but still be useful in including information about the code uncertainties, that could be used as a mathematically solid base for e.g. licencing. Having uncertainty limits that encompass all observation point, but at the same time are “worse” than conservative results do not further any analysis or licencing process. A simple validation application will be provided in the final version of this report.

Verification can be based on the same experimental test facility (even the same test), while for validation it is a minimum requirement to include different tests, or even better to apply the derived u.i.p. on the same phenomena in a different test facility. Which observational data is used for quantification and which for validation should be addressed

at the beginning of the UIQ process. Some guidelines are provided in the SAPIUM project /BAC 20/.

Both verification and validation involve FUQ. This is based on the Wilks' approach shortly described in section 3.1.

3.14 Requirements

A short summary of the main requirements of the established statistical method is provided here.

- Selection of the uncertain input parameters to be quantified.
  - Result of the problem definition
  - Overview in the simulation model
  - Guidelines: see SAPIUM /BAC 20/
- Definition of experimental data base
  - Selection of adequate experimental data base
  - Guidelines: see SAPIUM /BAC 20/
- Definition of priors
  - Any prior knowledge can be incorporated
  - No a-priori knowledge causes no problem
- Data base structure compatible output definition of ATHLET
  - Simulation outputs at specific times, location corresponding to the experimental matrix
- Choice of summary statistics
  - Descriptive: mean, variance, skewness, etc.
  - Quantiles
  - Empirical CDF
  - etc.

- Choice of distance metric
    - Used for the comparison of simulated data and observation
    - See section 3.11.3
- Choice of epsilon
    - Acceptance/rejection criteria in the ABC model
- Definition of MCMC features (# draws, # chain)
- Convergence check
    - See section 3.6
- Validation
    - FUQ to validate the quantified u.i.p.s
    - Enveloping, etc.

3.15 Problem of server and resource availability

A main issue in carrying out an MCMC based Bayesian inference is the considerable demands of the method with regard to computational resources. Depending on model complexity and the applied options it can take anywhere upwards of a couple of thousand simulations to converge and reach a reliable and stable result. Therefore, it is important to make considerations at the beginning how to allocate the necessary resources for the application. Tab. 3.8 includes some aspects we considered for this project, that can help decide how to best allocate the resources.

Tab. 3.8 Comparison of resources

Factor	Windows desktop PC	Unix HPC cluster with queuing system
Computational speed	• Slower • limited by fewer cores less RAM	• Faster • high core count • large RAM • potential GPU
Parallelisation	• Limited	• Extensive

Factor	Windows desktop PC	Unix HPC cluster with queuing system
	• often only within a few cores locally	• multi-node and multi-core capabilities
Suitability	• Prototyping • small- to medium-scale models	• Large, complex models • extensive sampling
Convenience	• Easier setup • immediate access	• Managed access • requires job scheduling
Scalability	• Limited • not ideal for large data or high-dim models	• Highly scalable • ideal for large Bayesian applications
Queuing overhead	• None	• Requires queuing, scheduling management
Availability over time	• Limited by regular updates requiring restart of the operating system • Excellent for the short term	• Theoretically long-term stable for simulations • Reasonable for the medium and long-term as a shared resource

Generally, for smaller models and testing purposes a Windows® desktop PC suffices. This would enable fast iteration and testing. Furthermore, with the help of WSL (Windows Subsystem Linux) some Python functions only available under Linux can also be conveniently tested. Large and expensive models with the possible need for parallelisation should be moved to a HPC cluster that is better suited to handle them, (in theory) significantly reducing runtime.

As a result, it has been decided to carry out the development work under a Windows desktop PC as well as the testing of the method. Once the actual ATHLET models were used, the study has been moved to the GRS HPC cluster. Furthermore, to avoid any problems whiling keeping the limited HPC resources in mind, it has been decided that the actual ATHLET simulations are sent to the queuing, while the framework (way less resource intensive, acts sort of a scheduler) is run directly.

In theory, this approach could have worked. Obviously, as the HPC cluster resources are limited, the queuing system causes some delay in actually starting each individual ATHLET calculation. In situations with a high HPC resource utilization, the resulting delays accumulate and can – and have been – considerable. Also, the GRS HPC cluster

is not optimized to handle a large number of small but I/O-intensive calculations, which further constrains performance of the approach. Nevertheless, these are no showstoppers for the calculation.

As we learned during the project, the reliability of the HPC server at GRS was not as high as expected. For the calculational scheme set up as described above, any server instability, including, e.g., a reboot of calculational nodes on the HPC server or an induced crash in scheduled ATHLET calculations, a loss of internal network connections during data transfer, can stop (and has stopped) the whole calculation chain. This has been partially addressed with the help of a terminal-multiplexer. However, this cannot be deemed a sufficient solution for any future applications, considering the delays experienced in this project. Unfortunately, these delays accumulated, and with the rollout of a server update, significant time was lost, preventing our simulation chains from completing within the project's timeframe. Nevertheless, we have identified approaches to address these limitations.

Based on the experiences in the project, any future set-up using the Bayesian inference approach investigated in this study should check if the following preconditions are fulfilled:

1. The calculational chain as a whole is assigned to dedicated HPC resources (as one calculational unit) in the HPC queuing system (obviously assuming sufficient and timely resources are available).

2. The HPC cluster needs to be configured so that it can cope with the I/O-intensive system code calculations that run in parallel on the limited set of calculational nodes. Invariably, this has to be achieved while such a HPC cluster will still have to support, e.g., CFD simulations with considerably different requirements.

3. The calculational set-up has to be stable and resilient against non-severe disturbances on the server. Individual system code simulations have to be restarted after a spurious crash automatically, without disproportionate delays or the need to redo substantial amounts of system code calculations.

   Alternatively, a standalone (or a set if necessary) desktop PC with sufficient resources, preferably running a Unix environment, can be fully dedicated to the simulations.

4. The data produced by the system code simulations has to be limited to the extent practicable to those needed to evaluate the QoI in the calculational chain.

5. The number of system code simulations has to be limited to the extent practicable. This will require further investigations into suitable methods.

3.16 Future improvements

Some possible future aspects to be investigated and further development work is listed below.

Investigating different ABC metrics

As mentioned in section 3.11.3 the choice of summary statistics as well as distance metric is crucial. Investigating different options could deliver useful information about the applicability of the method and could help develop guidelines concerning the application of the developed approach. Another aspect, that can be considered is the relaxing of the acceptance criteria (ϵ) in order to somewhat adjust the CPU.

Expanding the experimental data base

At the moment one experiment is considered (FEBA216) with only one time series. This could and should be expanded including not only several time series of this test, but also several tests. Furthermore, including more than one experiments (PERICLES; FLECHT; ACHILLES etc.) would be beneficial. Here, another point of the hierarchical approach can be investigated: including all at once, or deriving a chain of posteriors each improving the previous one as prior while adding information sequentially to the process. The latter might need also an outside iteration loop, but there might still be efficiency gains.

Saving chain-information mid analysis

At the moment no intermediate storage of chain information takes place (requiring no disturbances for the HPC resources used). Implementing such intermediate storage could address the stability problems associated to the HPC resource by making an already started chain restartable without having to repeat calculations already done. This, however, has to be implemented outside of the present framework and therefore would mean the implementation of the ABC method with only basic python packages. This is

feasible, but as all development and implementation work needs a sufficient time and resources.

Including modelling bias

Currently, the method only takes into consideration the experimental uncertainty and not the possibility of a model bias. This would have been the next step, but unfortunately due to the delays we experienced we could not reach this step. Nevertheless, it would be important to investigate its effects and possible inclusion in the quantification process. At the same time this would further complicate the identifiability problem. We only have simulation results and experimental values. Quantifying several u.i.p. at the same time generally can have several (local) optimums. Which one is the overall optimum, how the experienced uncertainty in results can be allocated to the specific u.i.p is a difficult question. It is possible to step outside of a set of specific u.i.p.s and apply only parts of this set, possibly combined with other u.i.p.s. For the latter there is a strong suggestion it might not be possible, but without further investigation no definite answer can be given at the moment.

Another aspect could be considering nested loops: first fixing either bias or experimental uncertainty, then moving to the next one.

Better resource allocation

Both on the HPC server as well as on the input processing and I/O side of ATHLET optimisation could be done to reduce the resource intensity of the approach and also to provide stable CPU capacity to perform the analysis. Approaches like, e.g., available in COCOSYS for restricting the plot vector to user defined output and turning off other data saving could significantly reduce the time needed for an ATHLET calculation, particularly at high plot frequencies needed in this scheme. This, however, has to be addressed at the code development level. It also might be possible to directly transfer the relevant result data to the governing calculation chain, without having to go via plot data written to a file. With its hash map functionality, ATHLET is prepared for such an approach. Reserving exclusive resources on the HPC cluster is already feasible, nevertheless it has drawbacks. Initial allocation takes longer and therefore testing before live calculations would be more difficult. Furthermore, if the chains do not converge within the set limits, the analysis might have to be repeated. This could be addressed by saving chain information before convergence.

The question of the likelihood function

Revisiting the consideration of constraining the prior and posterior pdfs in order to easier derive the likelihood could be beneficial. Likelihood free methods are quite powerful, but generally more resource intensive. There is a trade-off here between generality (general application of the method to fundamentally different problems) of the approach and resource requirements to derive the pdfs in a specific question.

Surrogate modelling

To make the quantification more feasible for more complex models an initial step of creating a meta-model seems necessary. Carrying out the vast number of simulations needed to properly explore the posterior distribution is only feasible if the run time of each simulation is negligible. In this study an ATHLET run averaged about 5 minutes of runtime. Nevertheless, as experience showed, this already causes problems, especially taking into account the fragility of the calculation chain for any disturbances coming from the HPC resources (see section 3.15).

4 CET and SET

4.1 Reflooding experiments

As a basis for a subsequent uncertainty analysis, recalculations of a number of experiments done in three different experimental facilities, namely FEBA, FLECHT, and PERICLES, were performed. Several of these experiments were used before to evaluate the capabilities of earlier version of ATHLET /TIB 15/, and one experiment of each facility (resp. two in case of FLECHT) is also part of the regular ATHLET validation matrix /HOL 23/.

For the new calculations presented here, the existing ATHLET inputs were used as basis, and they were extended for a convenient calculation of a larger number of experiments, making extensive use of the PARAMETERS feature of ATHLET. During the revision of the input decks, all geometry and material definitions in the input were thoroughly reviewed; where no data were provided by the operator of the facility, values were taken from established publicly accessible databases. Furthermore, some inaccuracies in certain parts of the model could be removed. This includes, for example, the more accurate modelling of the temperature dependence of the material properties and the correction of some errors in defining the materials in the rod objects. For all input decks, a few parameters were checked and changed, if necessary, to follow up-to-date modelling recommendations. The general level of detail of the model, and the nodalization, were not changed, however. In these aspects, there are thus some differences between the models for the different test facilities.

From a modelling point of view, all three facilities consist basically of a heated vertical rod bundle in a channel filled with steam, which is then flooded with liquid water to simulate reflooding after a loss of coolant accident. Despite this similarity, the modelling approaches are slightly different; in particular, the temperature initialization is done in different ways, which is also a result of the available information and data for each experimental setup.

The ATHLET model for each of the test facilities is described in the following section, together with a short description of the experiments themselves. The parameters for the selected test cases are given in the subsequent section together with the results of the ATHLET calculations and a comparison with available measurement data.

For all calculations shown here, the standard 6-equation model is used, and the quench front model of ATHLET is applied. Axial heat conduction in the rod is not considered in all models. The current version 3.4.1 of ATHLET is used. Recent versions of ATHLET tend to improve the calculation of reflooding, compared to ATHLET 3.2 or earlier, as documented in /HOL 23/. A comparison of different thermofluid-dynamic models (6 equations vs. 5 equations) for a few of these test cases was done in /TIB 15/.

4.1.1 Experimental facilities and modelling approach

4.1.1.1 FEBA facility

The FEBA (Flooding Experiments with Blocked Arrays) test facility was built at Institut für Reaktorbauelemente at the Kernforschungszentrum Karlsruhe (Germany) /IHL 84/, /IHL 84/. It was designed for a separate effect test program under idealized reflood conditions, with constant flooding rate and constant back pressure, and without the effect of a reactor cooling system. Eight series of experiments were carried out with different percentages of blockage (to model the effect of clad ballooning) in a range between 62% and 90% flow blockage.

The test section consists of a 5x5 full length rod array placed in a square stainless steel housing with a wall thickness of 6.5 mm. A schematic representation of the cross section of the rod array can be seen in Fig. 4.1. The test section is well insulated to reduce heat losses to the environment. The heater rods, shown in Fig. 4.2, are electrically heated with a flat cosine power profile (peak to average ratio 1.19), approximated by 7 steps of different specific power, as shown on the left side of Fig. 4.3. The outer diameter of the rods is 10.75 mm, the pitch is 14.3 mm, and the inner length of the quadratic housing is 78.5 mm, thus resulting in a hydraulic diameter of 13.47 mm for all channels. The active heated rod length is 3.900 m. There are 7 spacer grids (in equidistant positions, shown in Fig. 4.3) which reduce the flow area by 20%. During the experiment the cladding, the housing and the fluid temperatures, mass flow rate, flooding (inlet) velocity, pressures and pressure drops have been measured throughout the entire length of the test section.

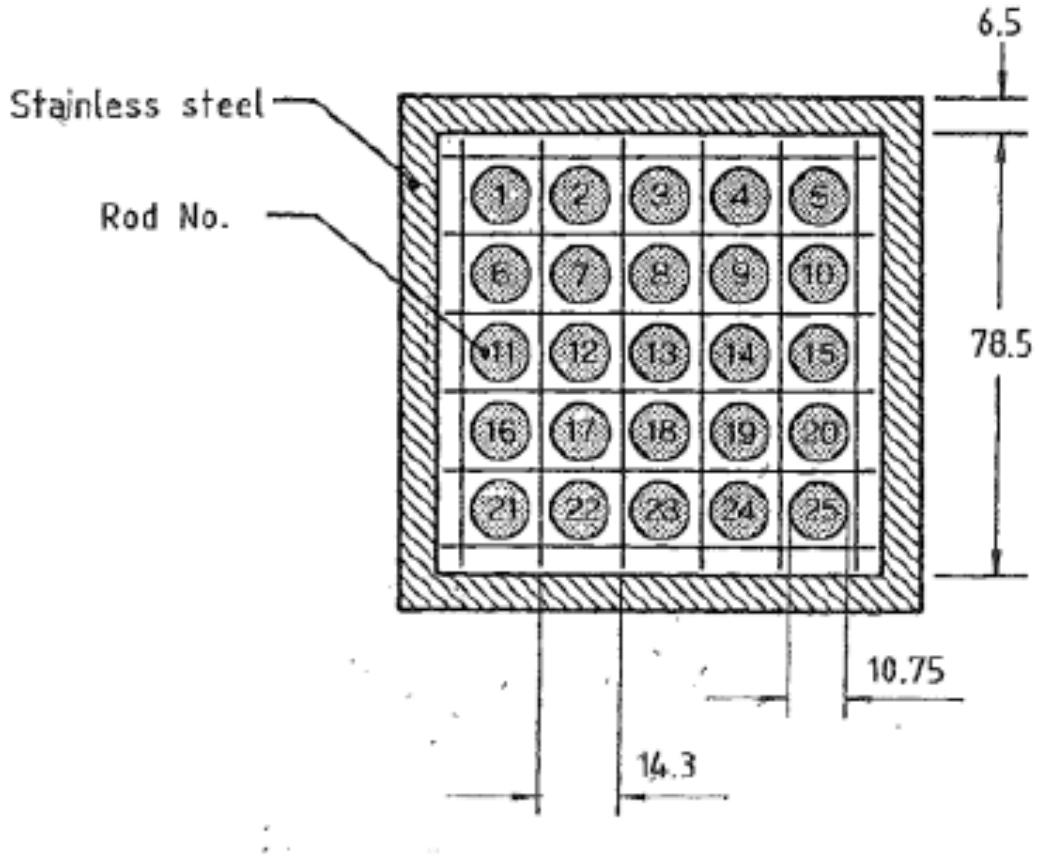

Fig. 4.1 Cross section of the FEBA 5x5 rod array; dimensions in mm (from /IHL 84/)

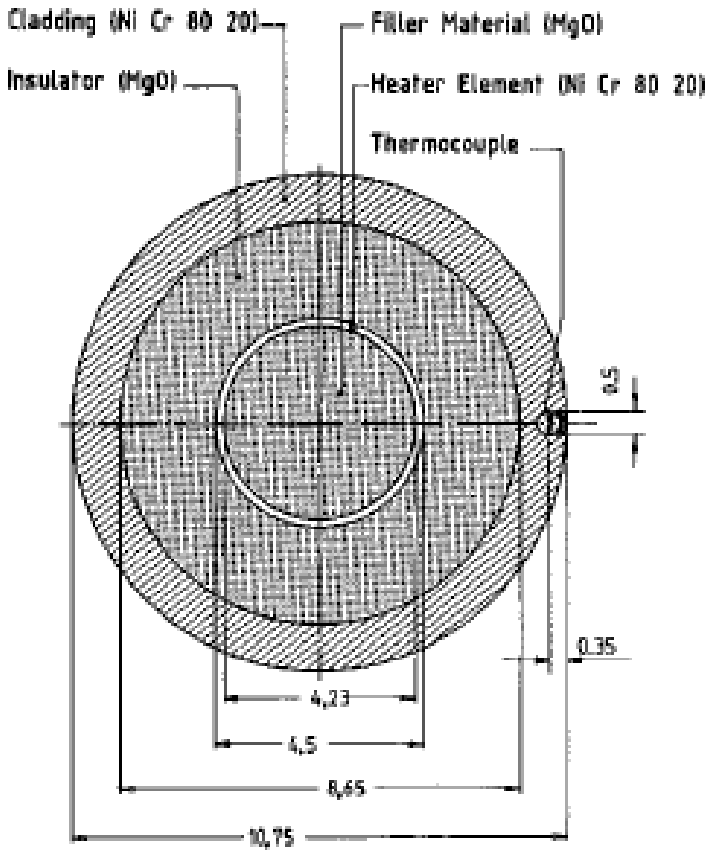

Fig. 4.2 Cross section of a FEBA heater rod; dimension in mm (from /IHL 84/)

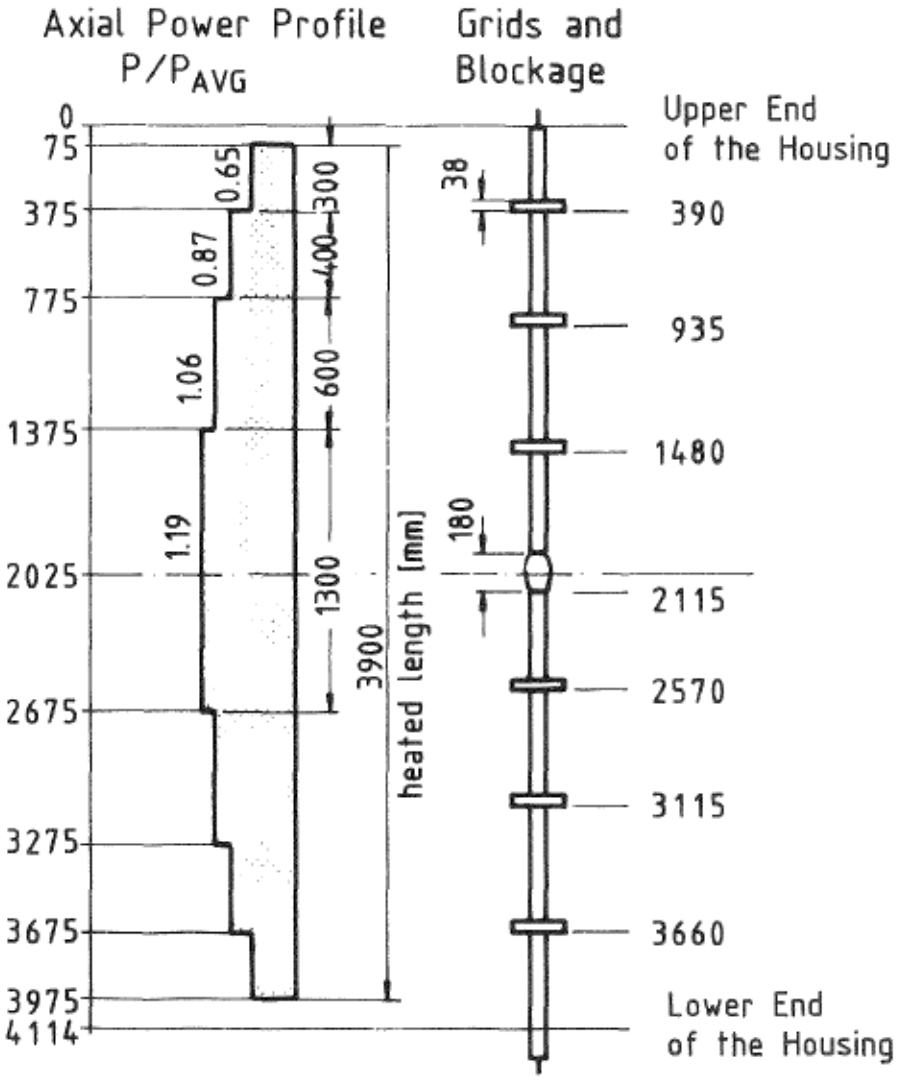

Fig. 4.3 Representation of the power distribution in a FEBA rod (left) and positions of the spacers (from /IHL 84/)

Most of the instrumentation consisted of thermocouples to measure cladding, sleeve, local channel, and housing temperatures at various locations. Furthermore, pressure differences as well as the amount of water separated in the water-steam separator above the test assembly were continuously measured.

The parameters, which were varied in the runs of the eight test series, are the bundle geometry (percentage of blocking), the presence of grid spacers, the flooding rate, and the system pressure. Pressure, flooding rate, and feedwater temperature were kept constant during each experiment run. For the recalculations presented here, only

experiments from the reference test series 1 were chosen, that is, undisturbed bundle geometry without any blockage and with all spacer grids present.

4.1.1.2 FLECHT facility

The aim of FLECHT-SEASET (Full Length Emergency Core Heat Transfer - SEparate And System Effect Tests) reflood tests was to examine the heat transfer and entrainment phenomena in the heated bundle during the reflooding phase of a LOCA and to provide a reflooding database for use in the development and validation of reflooding predictions. The test facility was situated at the Idaho National Engineering Laboratory (USA) /LOF 80/. It was based on the existing FLECHT facility which was modified to use different heater rod bundles with dimensions typical to newer PWRs.

The FLECHT rod bundle (Fig. 4.4), consists of 161 heater rods, and 4 instrumentation thimbles, 12 steam probes, 8 solid rectangular fillers and 8 spacer grids. Its layout is rectangular similarly to the then 17x17 Westinghouse fuel bundle design. The housing is a cylindrical vessel with an inner diameter of 193.7 mm, and a thickness of 4.78 mm. The heater rods have an outer diameter of 9.5 mm, the nominal pitch is 12.6 mm, resulting in a hydraulic diameter of 11.8 mm. The actively heated length is 3.66 m, and has a stepped, cosine power profile (peak to average ratio 1.66). This study includes also some test from the previous FLECHT-SET program using a skewed axial power profile /ROS 77/. Both profiles are show Fig. 4.5. In the runs selected for this study, some of the rods were disconnected, resulting in 159 resp. 157 heated rods in the bundle.

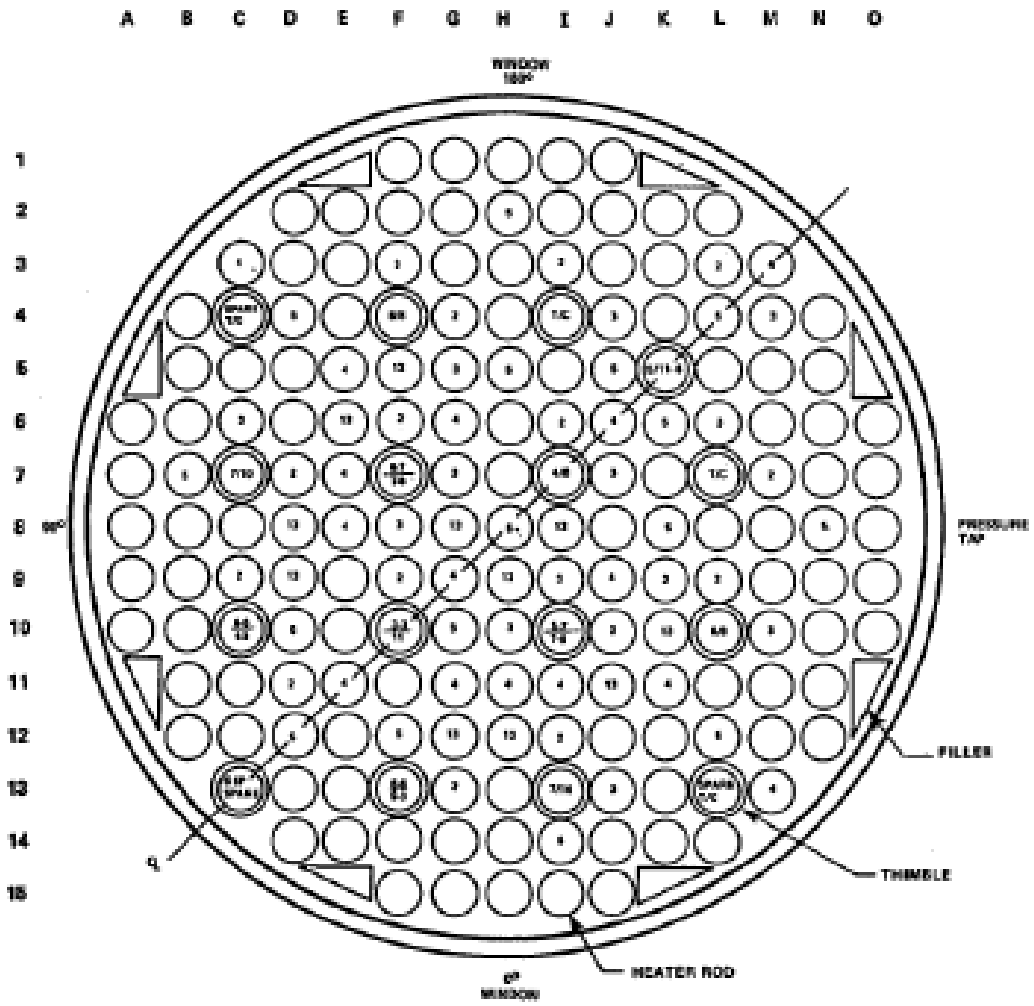

Fig. 4.4 Cross section of the FLECHT rod bundle (from /LOF 80/)

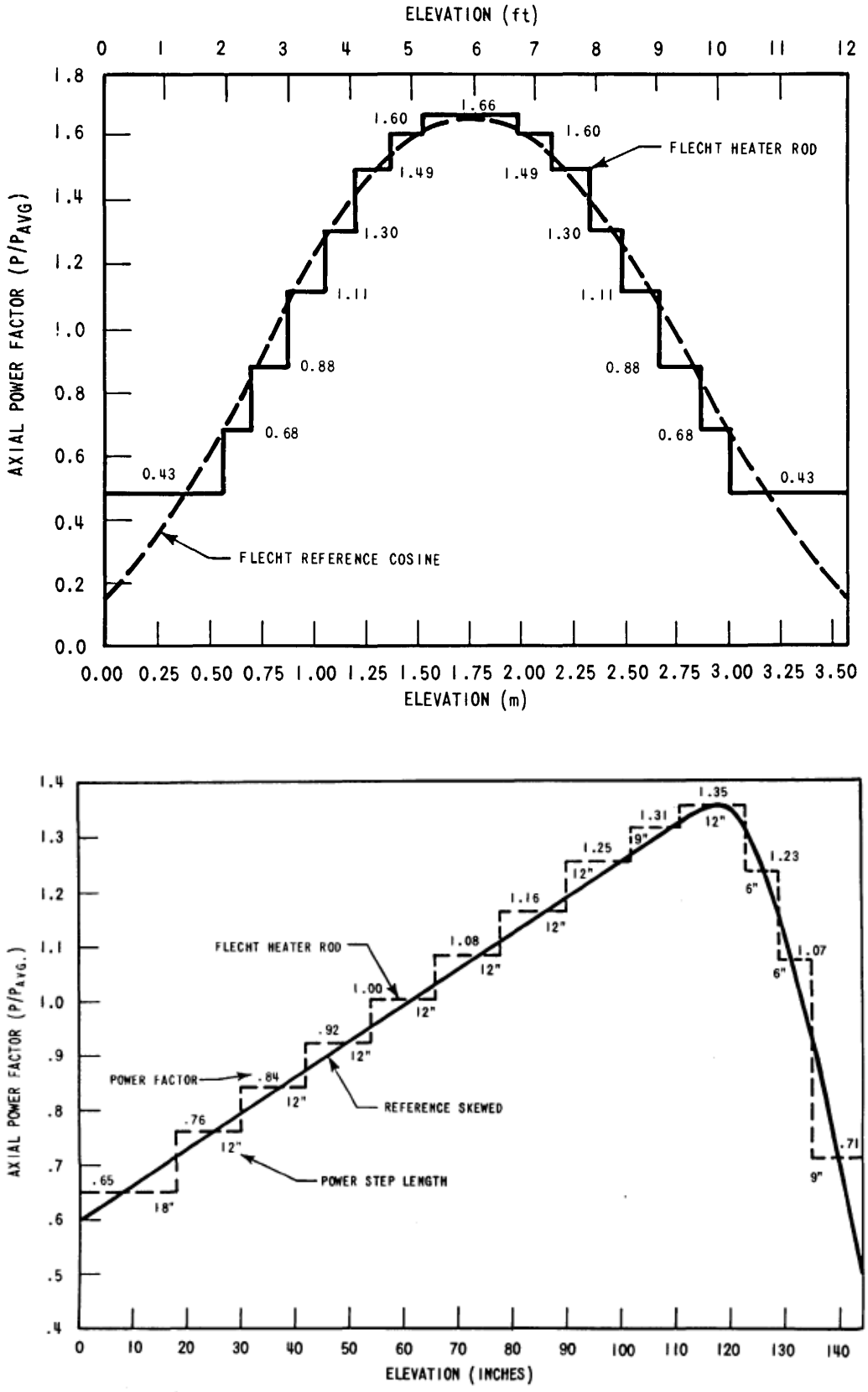

Fig. 4.5 Axial power profile of a FLECHT rod: cosine profile (from /LOF 80/), and skewed profile (from /ROS 77/)

The instrumentation of the facility was designed to measure temperature, pressure, flow, liquid level, and power. Most relevant data for the comparison with the calculation are again rod temperatures, pressure differences, and the level of the liquid collected in the carry-over tank.

The tests in the whole program examined the effects of initial clad temperature, variable stepped flooding rates, rod peak power, constant low flooding rates, coolant subcooling, and system pressure.

4.1.1.3 PERICLES facility

A series of experiments devoted to the reflooding phenomena at low pressure was performed at Commissariat à l'énergie atomique (CEA) in Grenoble (France) in the PERICLES test facility /SKO 13/.

The facility consists of three rod bundles or assemblies (Fig. 4.6), each having 7x17 rods (Fig. 4.7); the total number of rods is thus 357. There are 8 spacer grids each reducing the flow area by 22%.

The aim of these experiments was to investigate the 2D effects in the PWR core in the case of a difference in power between bundles. The velocity of the quench front in the three bundles may be slightly different because of the different rod power between the hot and cold assemblies.

The test section is surrounded by a stainless steel housing (91.5 mm x 646.5 mm, wall thickness 7 mm). The heater rods (Fig. 4.8) are without gap, with stainless steel cladding, boron nitride as insulator and Nichrome V as heating material. They have an outer diameter of 9.45 mm, a pitch of 12.6 mm, and the distance between the rods and the wall are 8.25 mm and 7.95 mm, resulting in a hydraulic diameter of 11.4 mm. The power profile is a chopped cosine shape, with a peak to average ratio of 1.6, approximated by 11 steps as shown in Fig. 4.9. The active heated length is 3.656 m and is equal to the length of the channel.

The rods in the assemblies can be independently heated. The central assembly is also referred as "hot assembly" because in most runs, it has a higher heating power (given by the factor F_{xy}, with typically $F_{xy} = 1.435$ for runs with non-uniform power distribution) than the two lateral assemblies.

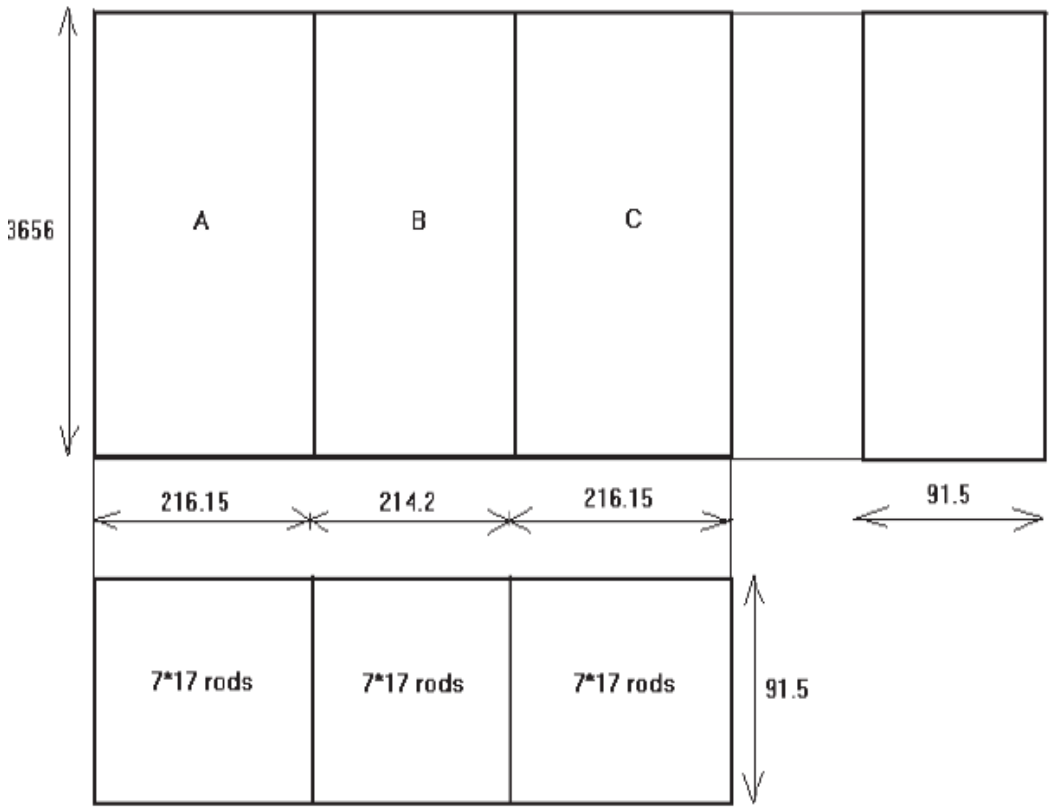

Fig. 4.6 Arrangement and dimension (in mm) of the rod assemblies in the PERICLES experiment (from /SKO 13/)

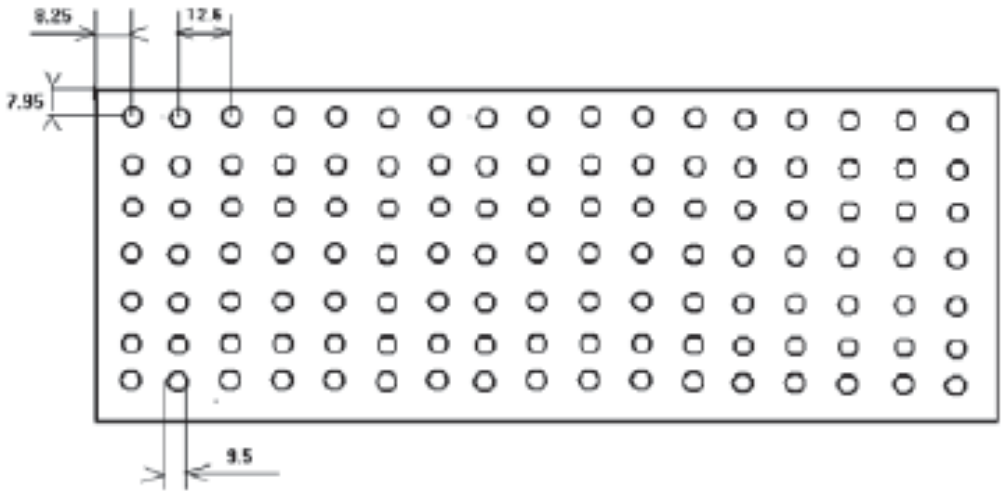

Fig. 4.7 Horizontal section of one PERICLES assembly; dimension in mm (from /SKO 13/)

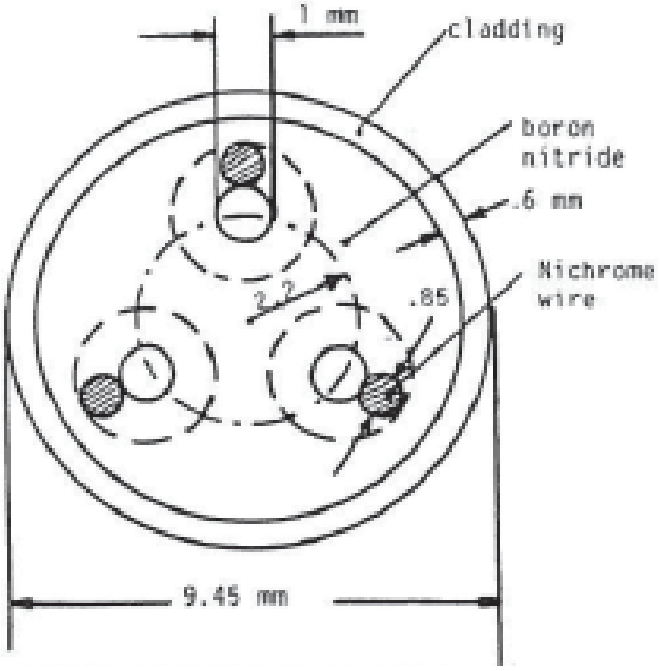

Fig. 4.8 Section of a rod in PERICLES (from /SKO 13/)

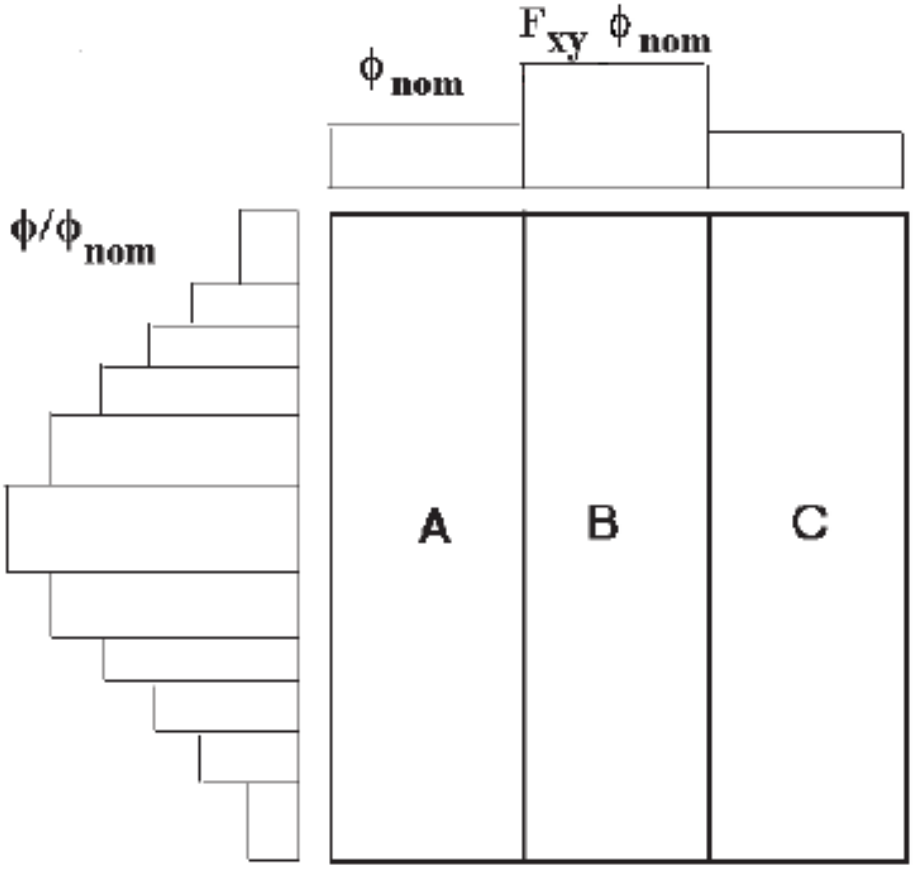

Fig. 4.9 Power distribution in the assemblies and along a heating rod in PERICLES (from /SKO 13/)

Besides pressure difference measurements, a large number of thermocouples provide cladding temperature data at different rods and different elevations, which can provide valuable insight into the quenching. For this study, however, only a very limited number of measurement data was accessible.

4.1.2 ATHLET Modelling for the Experiments

4.1.2.1 ATHLET model for FEBA

The FEBA test section (Fig. 4.10) is modelled in ATHLET as a one-dimensional pipe (object `CHANNEL`) with 23 axial nodes. The spacer grids are considered as reductions to the pipe's cross-sectional area and as form losses. The channel is initialized with stagnant steam, with the initial temperature to be the temperature distribution of the wall as in the experiment. Coolant is supplied by a fill at the lower end of the pipe. At the top end of the channel there is a time dependent volume (object `PH-BOUND`), which acts as a drain for the steam flow and the entrained water. Mass flow and enthalpy of the fill and the pressure and enthalpy of the time dependent volume are given by GCSM signals, with values taken from the experiment.

A single heater rod (object `HEATROD`) is modelled with 23 axial nodes and 5 radial layers. The correct number of the rods is considered by means of the multiplication parameter `FPARH`. The inner three layers are magnesium oxide, the outer two are made of nichrome. The object `RODINIT` in the centre of the rod is an auxiliary object to support the initialization of the rods with the correct temperature distribution. The housing consists of four layers and the outside temperature is controlled by a GCSM signal.

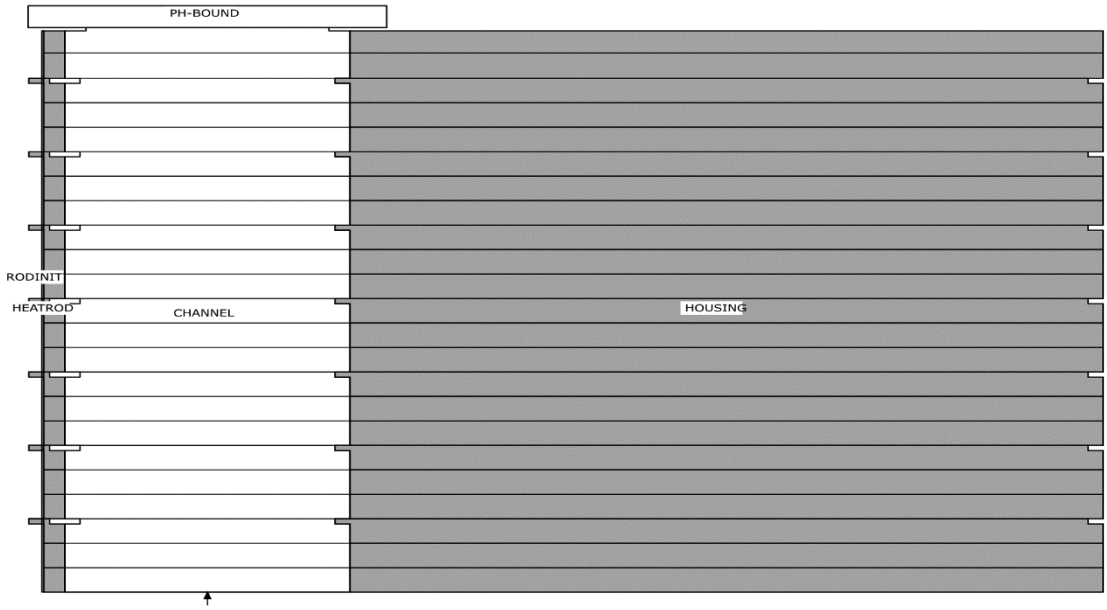

Fig. 4.10 ATHLET model and nodalisation for FEBA

In the calculation, the temperature initialization is completely done in the steady-state calculation, and the feedwater mass flow starts at t = 0 s, reaching its full value after 1 s. The feedwater temperature varies with time. The rod heating follows a decay curve.

Following the experience with this model during the general ATHLET validation (based on the single experiment run 216; /HOL 23/), the value of the quench front model parameter CQHTWB (maximum possible heat transfer coefficient on the wetted side of the lower quench front) is reduced (compared to its default value) to $3 \cdot 10^{-4} \frac{\mathrm{W}}{\mathrm{m^2\,K}}$, thus delaying the quench front progression.

For the recalculation of the different test runs, values of the system pressure (constant), the enthalpy of the vapor at this pressure, the density and the velocity of the feed water, the time-dependence of the feed water enthalpy, and the temperature profiles for the initialization must be provided in form of parameters.

4.1.2.2 ATHLET model for FLECHT

The FLECHT test section (Fig. 4.11) is modelled as a pipe (object `ROHR1`) with 12 control volumes. The spacer grids are considered only as form losses, not as reductions of the cross section. A branch object (`LUMP1`) below the main channel represents the lower plenum. The coolant is supplied into the lower plenum by a fill object (object `SJPFILL`)

and the steam-water mixture is drained in a time-dependent volume (object LUMP2) at the top of the channel. The initial temperature distribution of the heater rods (object HEATER, representing a single rod, with multiplication factor FPARH considering the number of similar rods in each experiment run) is taken as the initial temperature of the channel. The heater rod heat conduction objects (HCO) consist of 60 volumes axially, 5 per pipe section, and 8 layers radially. The two innermost layers are heated boron nitride, followed by four unheated layers of boron nitride and then two layers of stainless steel. The housing is not simulated, that is, there are no heat losses from the channel to the outer side.

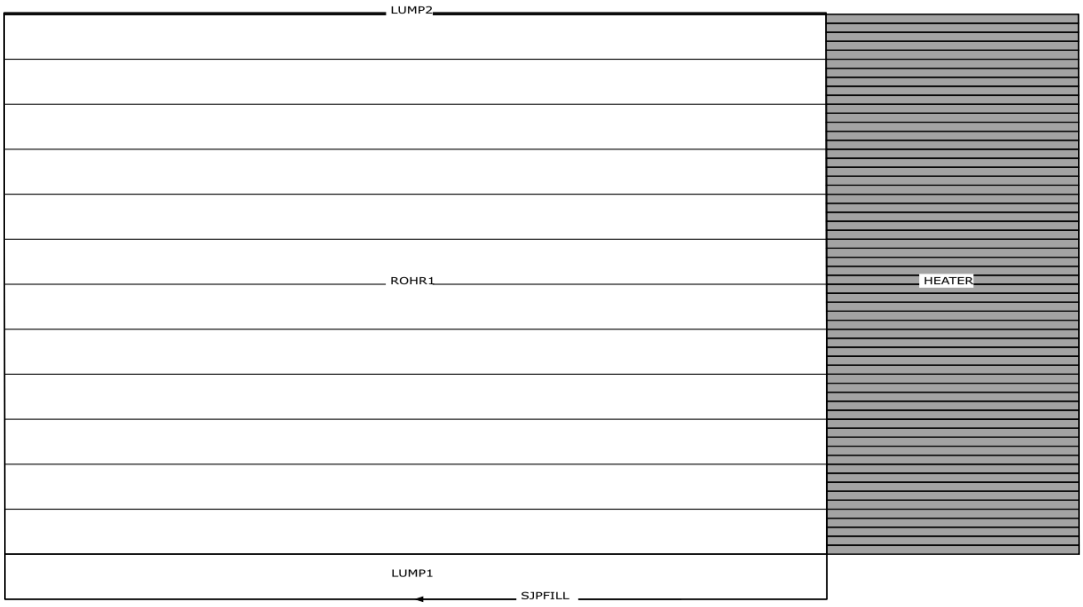

Fig. 4.11 ATHLET model and nodalisation for FLECHT

In the actual experiments, power is supplied to the heater rods until a given initial temperature value is reached in two of the designated thermocouples which initiates the water injection. In the calculation, the temperature initialization is done in the steady-state calculation, followed by a short (few seconds) phase of the transient calculation without feed water flow. The rod heating starts after 0.5 s and follows a decay curve applying the 120% ANS standard. The feedwater mass reaches its full value after 1 second, corresponding to t = 0 s.

For these experiments, default values for the parameters of the quench front model of ATHLET were used.

The parameters, for which values must be provided for each test run, include the number of connected rods, system pressure and temperature and enthalpy of saturated vapor at this pressure, the peak power at the begin of the rod heating, the coolant density and temperature, and the corresponding enthalpy (kept constant during a test run) and velocity, as well as the initial temperature profile.

4.1.2.3 ATHLET model for PERICLES

The PERICLES test section is modelled as 3 parallel pipes (objects `CHAN-HO`, `CHAN-C1`, `CHAN-C2`), each with 23 control volumes (Fig. 4.12). The central channel is connected to the other two with cross-connection objects (`CRS-CO1`, `CRS-CO2`), creating a quasi-two-dimensional representation. The spacer grids are considered as cross-section reductions and form losses. A branch object below (`IN-BRAN`) connects to all three channels. The coolant is supplied into this branch through a fill object (`GH-BOUN`). At the top, the channels and in a time dependent volume (object `PH-BOUN`), into which the steam-water mixture is drained. Each channel is linked to a heater rod and a housing object, which consist of 23 axial volumes each. The heater rod HCOs (objects `ROD-HOT`, `ROD-CL1`, `ROD-CL2`) consist of three radial layers, heated boron nitride, un-heated boron nitride and stainless steel, as a reasonable representation of the inner structure of the rods (Fig. 4.8) which cannot be exactly reproduced in ATHLET. The housings (objects `HOUS-HOT`, `HOUS-CL1`, `HOUS-CL2`) are simulated with two layers.

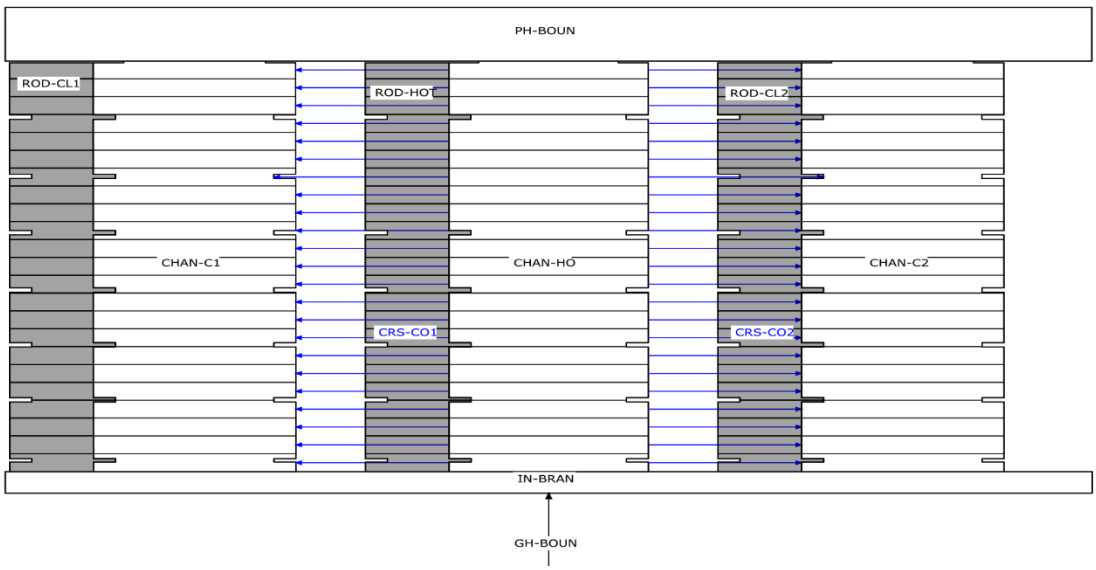

Fig. 4.12 ATHLET model for PERICLES; for clarity, the HCOs representing the housing of the assemblies are not shown

For the PERICLES calculations, the initialization is done in a different way than in the FEBA and FLECHT calculations, where data for the initial temperature distribution were provided. Here, for the steady-state calculation, a low steam mass flow at saturated conditions in the channels is assumed. At the begin of the transient calculation, the heating is switched on, and the flooding starts when a given cladding temperature is reached in the area with strongest heating. This procedure is close to the experimental procedure. Different to the FEBA and FLECHT experiments, where the heating follows a decay curve, the power is kept constant in the PERICLES experiments.

For the quench model, a value of $10^5 \ \frac{\mathrm{W}}{\mathrm{m^2\,K}}$ was chosen for the parameter `CHTWB` instead of the default $3 \cdot 10^5 \ \frac{\mathrm{W}}{\mathrm{m^2\,K}}$ (recommended range: $2 \cdot 10^4$ to $3 \cdot 10^5 \ \frac{\mathrm{W}}{\mathrm{m^2\,K}}$) to achieve a slower quench front progression, based on previous experience with recalculations of PERICLES experiments.

The parameters which must be provided for each test case are the system pressure and the corresponding saturation temperature and vapor enthalpy, the power of a single rod, the power shape factor F_{xy} for the central (hot) assembly, the cladding temperature at which the reflooding is started, the (constant) enthalpy of the feed water and the mass flux through the channels.

4.1.3 Results

4.1.3.1 FEBA

The FEBA tests selected for this study are all taken from test series 1 (base line tests with undisturbed bundle geometry; 7 grid spacers) and vary mainly in system pressure and the feedwater flow rate (Tab. 4.1).

Tab. 4.1 Parameters for the FEBA tests which were recalculated in this study

Test number	System pressure [bar]	Feedwater velocity [m/s]	Feedwater temperature [°C]		Cladding temperature [°C]	Housing temperature [°C]
			0-30s	end		
214	4.1	0.058	45	37	773	635
216	4.1	0.038	48	37	787	640
218	2.2	0.058	42	37	757	666
220	6.2	0.038	49	37	789	699
222	6.2	0.058	43	36	747	647
223	2.2	0.038	44	36	763	671

Cladding and housing temperatures given here are measured in the centre of the test section. Full profiles, as well as the full time dependence of the feed water temperature, are available in graphical representations in /IHL 84/.

The results of the calculations are shown in Fig. 4.13 to Fig. 4.18. Calculated values of the cladding temperature, the quench front, the pressure drop in different ranges along the rod bundle, and the water accumulated in the carry-over tank are compared with measured data. The measurement range of the water level in the carry-over tank was limited in the facility, thus this comparison can only be done in the early phase of the experiment.

The results of the reference case 216 (Fig. 4.14) show that in the beginning of the test, the temperature rises further as the coolant quickly evaporates when entering the test section and the steam cooling is not enough to compensate for the heating power. At lower elevations in the rod, the peak cladding temperature agrees well with the experiment, although the peak temperature is reached a bit earlier in the calculation. At higher

elevations, the temperature rises faster in the calculation and reaches a higher peak temperature. At all elevations, the cool-down starts earlier in the calculation, and the quenching happens too early as well. The difference in quench time is moderate for the lower elevations but becomes larger for higher elevations.

The calculated pressure drop is in reasonable to good agreement in all parts of the test section, and the same holds for the total pressure drop. The water carry-over tends to be slightly overestimated but is still in good agreement.

The general picture does not change very much when the pressure is varied. A comparison of the results for tests 214 and 216 shows that with a lower injection rate (test 216) the differences between the calculated and measured quench time increase. The best agreement in quench times is found for test 222 (Fig. 4.17), with higher pressure and higher injection rate.

In /TIB 15/ it was evaluated whether there may have been a deviation of the specified power profile in the experiments. It was found that a better agreement can be achieved with a slightly modified, top-shifted power profile. For the study presented here, however, the specified power profile as shown in Fig. 4.3 was used.

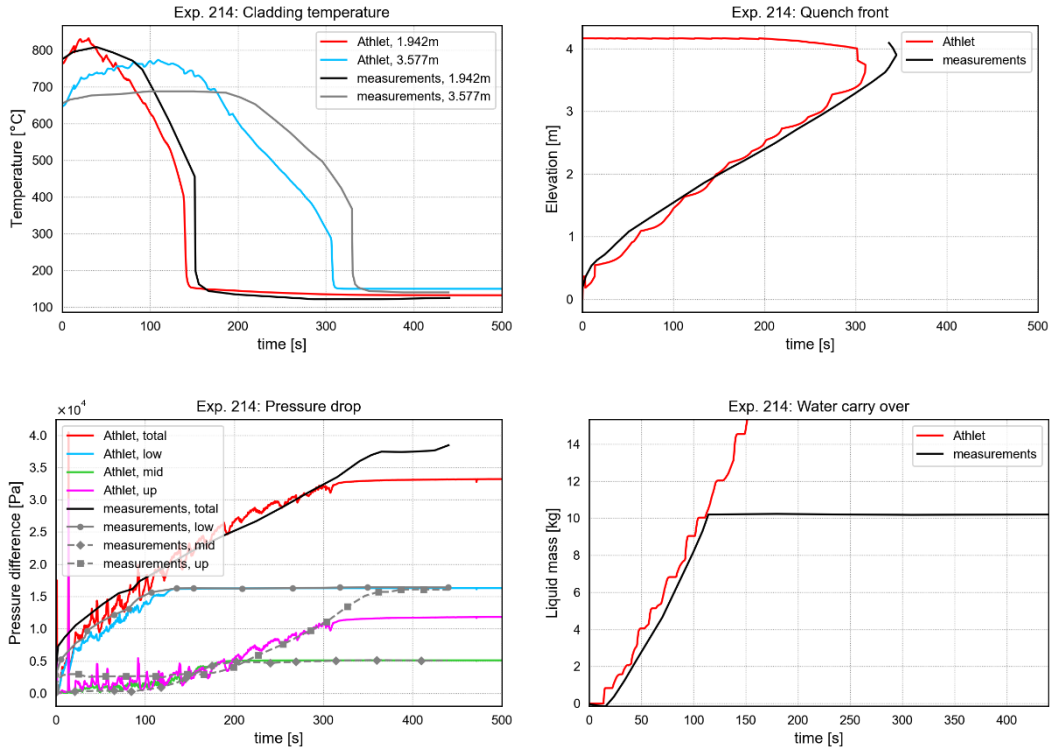

Fig. 4.13 Main calculation results for FEBA test 214 and comparison with measurements in the experiment

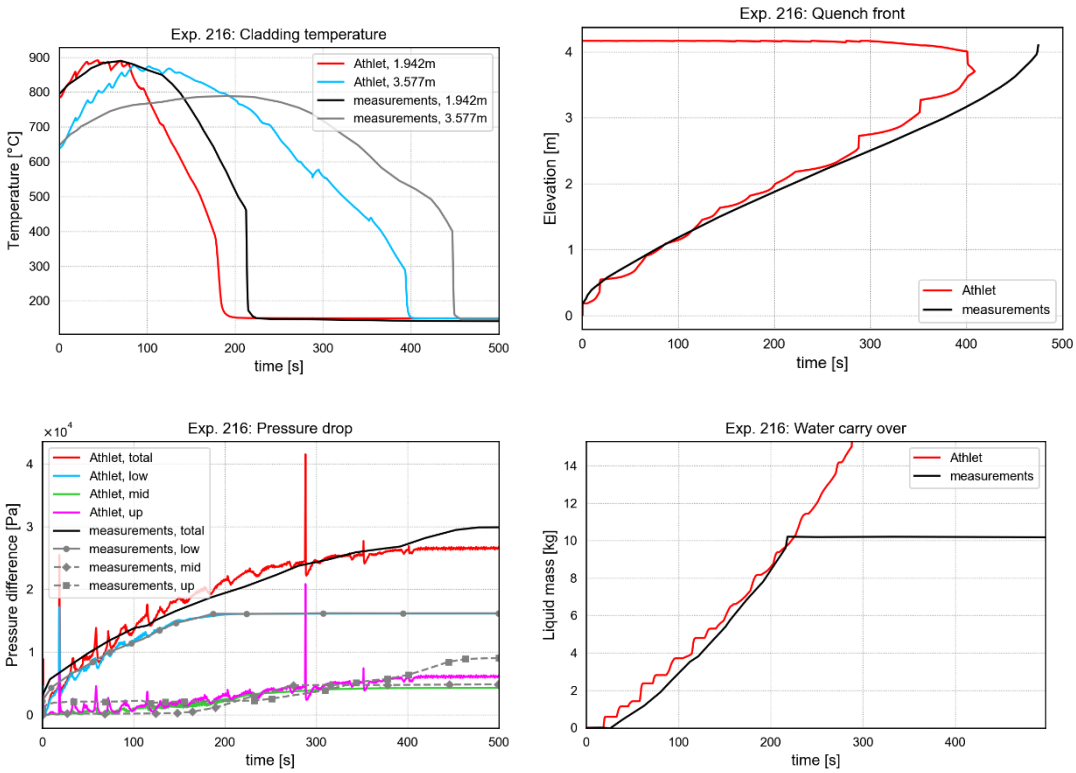

Fig. 4.14 Main calculation results for FEBA test 216 and comparison with measurements in the experiment

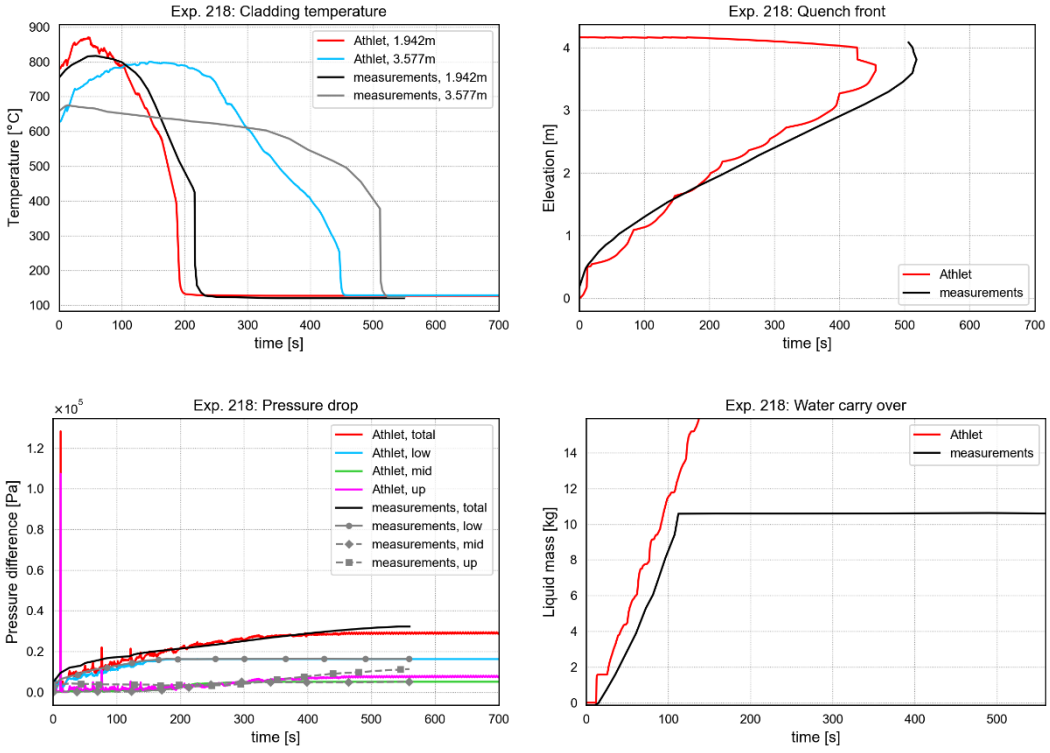

Fig. 4.15 Main calculation results for FEBA test 218 and comparison with measurements in the experiment

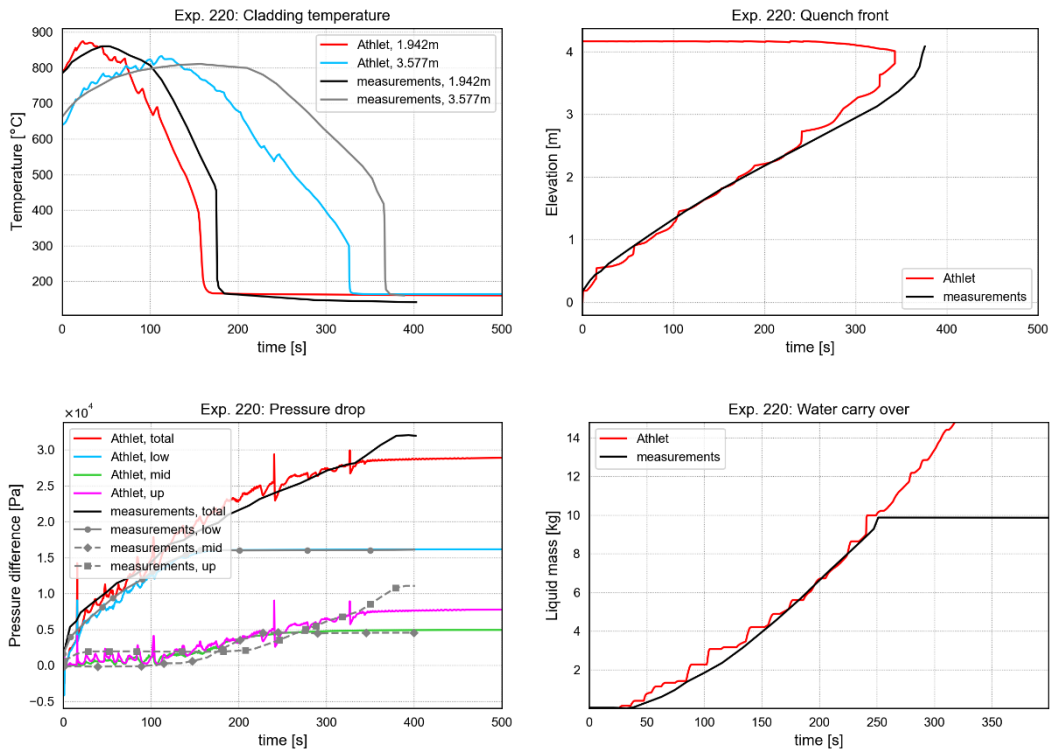

Fig. 4.16 Main calculation results for FEBA test 220 and comparison with measurements in the experiment

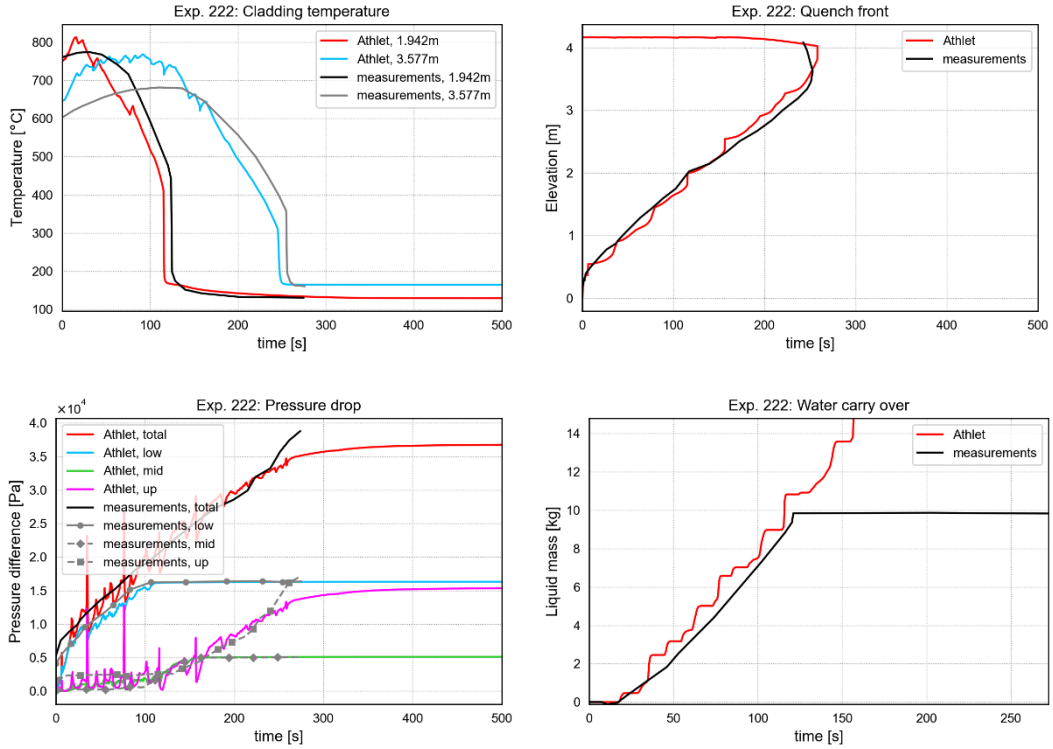

Fig. 4.17 Main calculation results for FEBA test 222 and comparison with measurements in the experiment

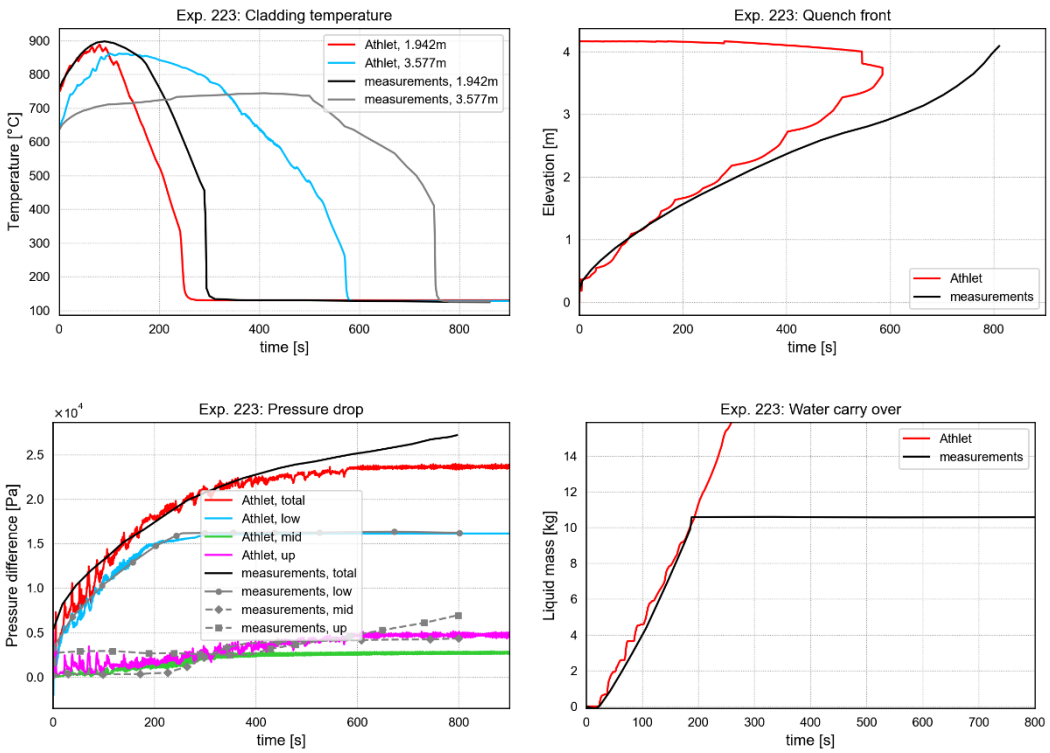

Fig. 4.18 Main calculation results for FEBA test 223 and comparison with measurements in the experiment

4.1.3.2 FLECHT

The FLECHT tests selected for this study originate from different test series. In the tests 1xxxx, a skewed axial power profile was used. In tests 3xxxx, a cosine power profile was used. Otherwise, the tests differ mainly in system pressure, feedwater temperature and flow rate (Tab. 4.2).

Tab. 4.2 Parameters for the FLECHT tests which were recalculated in this study

Run number	Power profile	System pressure [bar]	Feedwater velocity [m/s]	Feedwater temperature [°C]
31203	Cosine	2.76	0.0384	52
31504	Cosine	2.76	0.0246	51
31701	Cosine	2.76	0.155	53
31805	Cosine	2.76	0.021	51
32013	Cosine	4.14	0.0264	66

Run number	Power profile	System pressure [bar]	Feedwater velocity [m/s]	Feedwater temperature [°C]
13609	Skewed	1.38	0.0254	31
13812	Skewed	2.76	0.0254	84
13914	Skewed	1.452	0.0254	106
15305	Skewed	2.76	0.0203	53
15713	Skewed	2.76	0.0254	129

Cladding temperatures measured at the position of the power peak are not separately listed in this table. For all experiments, it was about 860 to 870 °C. The rod peak power was 2300 W/m in all cases. In test 15305, 157 rods were heated, while all other tests were done with 159 heated rods.

The results of the calculations are shown in Fig. 4.19 to Fig. 4.23 for the tests with a cosine power profile, and in Fig. 4.24 to Fig. 4.28 for the test with the skewed power profile. For comparison with experiments, measurement data of cladding temperatures at different levels, of the pressure drop over the whole rod bundle, and of the accumulated water in the carry-over tank can be used. Data for the full quench front are provided only for the tests with the cosine profile. These datapoints had to be extracted from plots with a cloud of points from temperature measurements at several positions in the rod bundle at different elevations, and the shown curve is thus only an approximation.

Note that in the ATHLET input for FLECHT, the quench front model is used with its default parameters, while in the simulations of FEBA and PERICLES experiments, the parameter CHTWB was reduced to slow down the quenching. Here, the parameter was kept at this value to remain in agreement with the modelling approach used for the recurring validation calculations in /HOL 23/. This parameter is obviously one of those to be varied in an uncertainty calculation.

Comparing the tests with the cosine profile (Fig. 4.19 to Fig. 4.23), it can clearly be seen that in the ATHLET calculations, quenching happens generally too early at all elevations. The difference in quench times is moderate for the more important lower quench positions and becomes larger for higher elevations. The largest error in quench times is observed for test 31805 with the lowest reflooding rate. A notable exception to this general observation is the test 31701 with high reflooding rate (Fig. 4.21): here, the quenching

happens too early in the calculation only at the lowest evaluated position (red curve), while for higher elevations, the calculated quenching is too slow, again with the error increasing with the elevation. The peak temperature at the low elevation is well reproduced, while for the higher elevations, it is slightly overestimated (and again roughly reproduced for the highest elevation), with the peak temperature typically reached too early. Again, the test 31701 with its high reflooding rate is an exception: here, the peak temperature is overestimated for all but the lowest elevation, with the largest error in the highest elevation.

The pressure drop is typically slightly overestimated by ATHLET, with the best agreement for the test 32013 with higher pressure. The water carry-over is clearly overestimated by ATHLET.

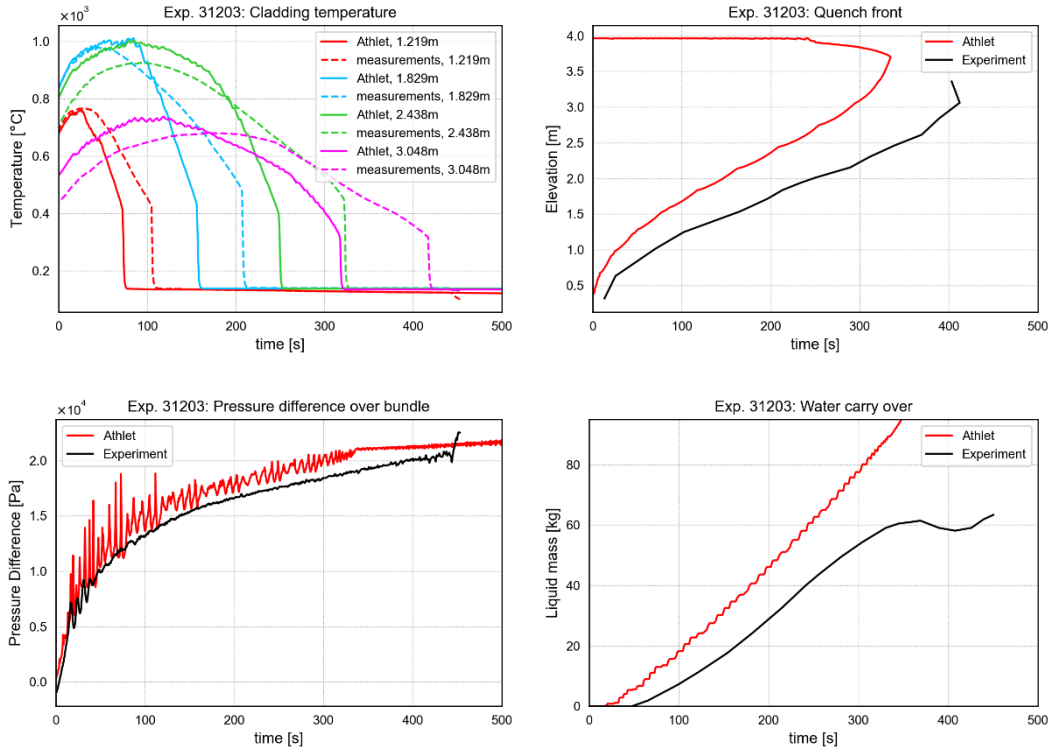

Fig. 4.19 Main calculation results for FLECHT test 31203 and comparison with measurements in the experiment

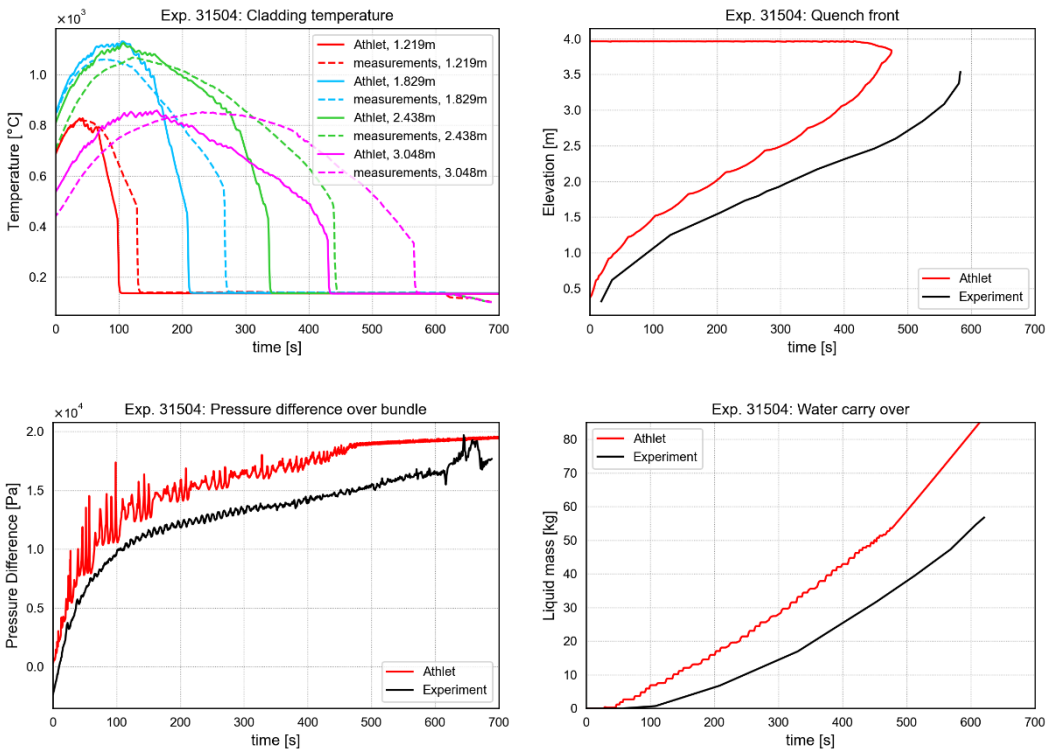

Fig. 4.20 Main calculation results for FLECHT test 31504 and comparison with measurements in the experiment

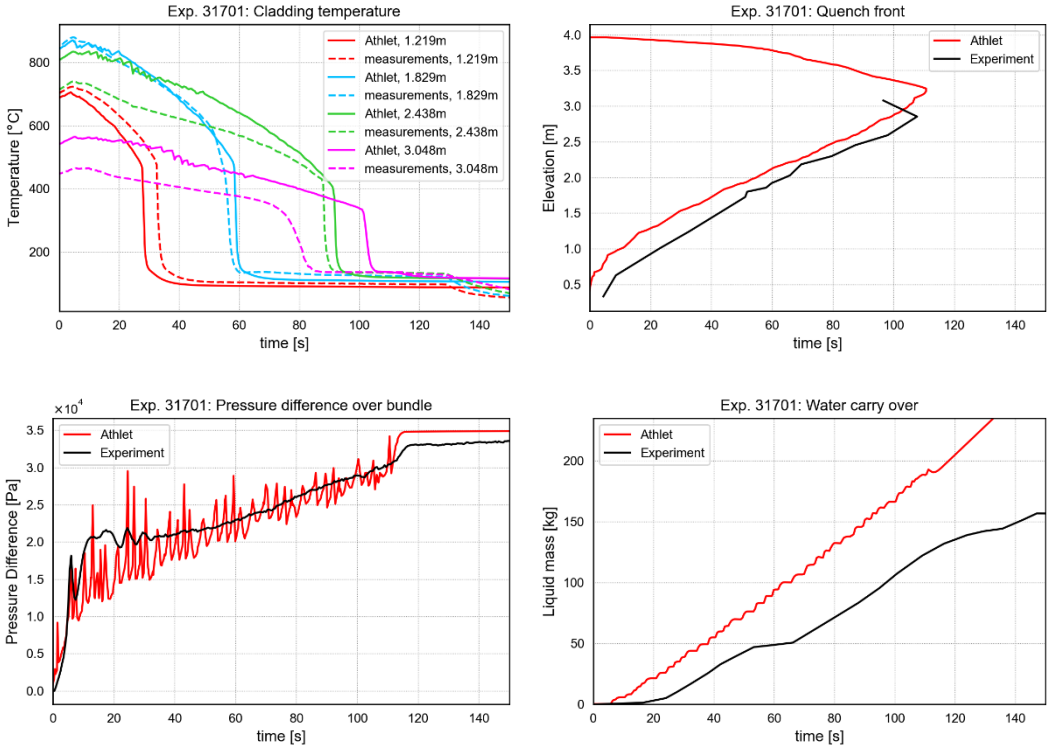

Fig. 4.21 Main calculation results for FLECHT test 31701 and comparison with measurements in the experiment

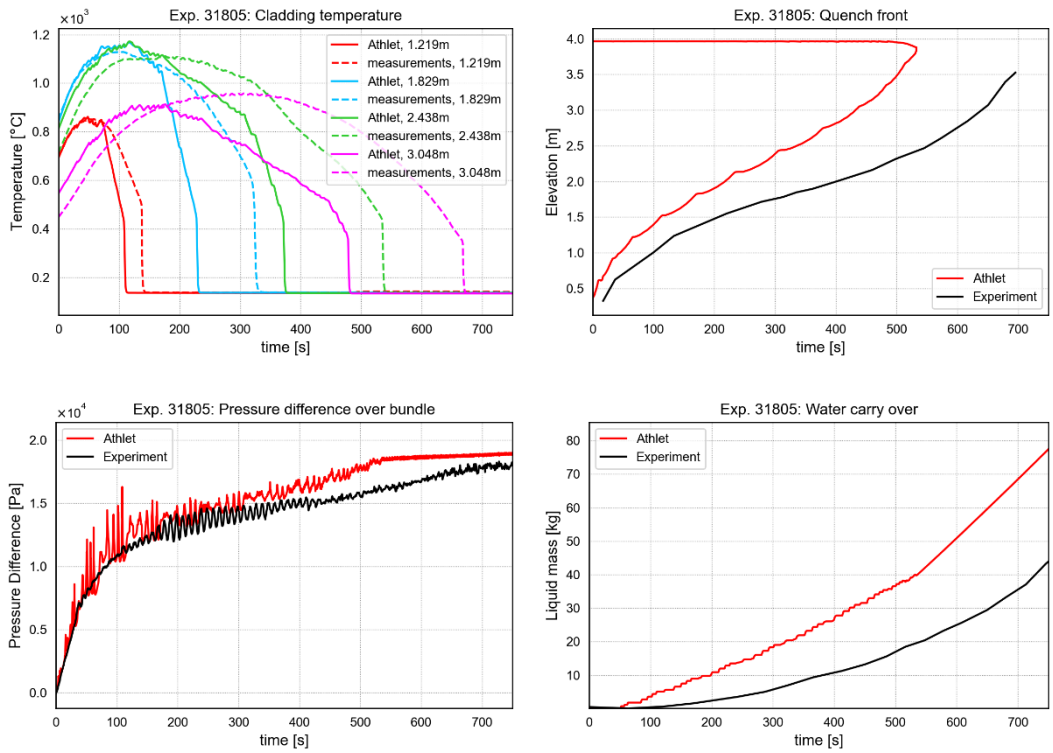

Fig. 4.22 Main calculation results for FLECHT test 31805 and comparison with measurements in the experiment

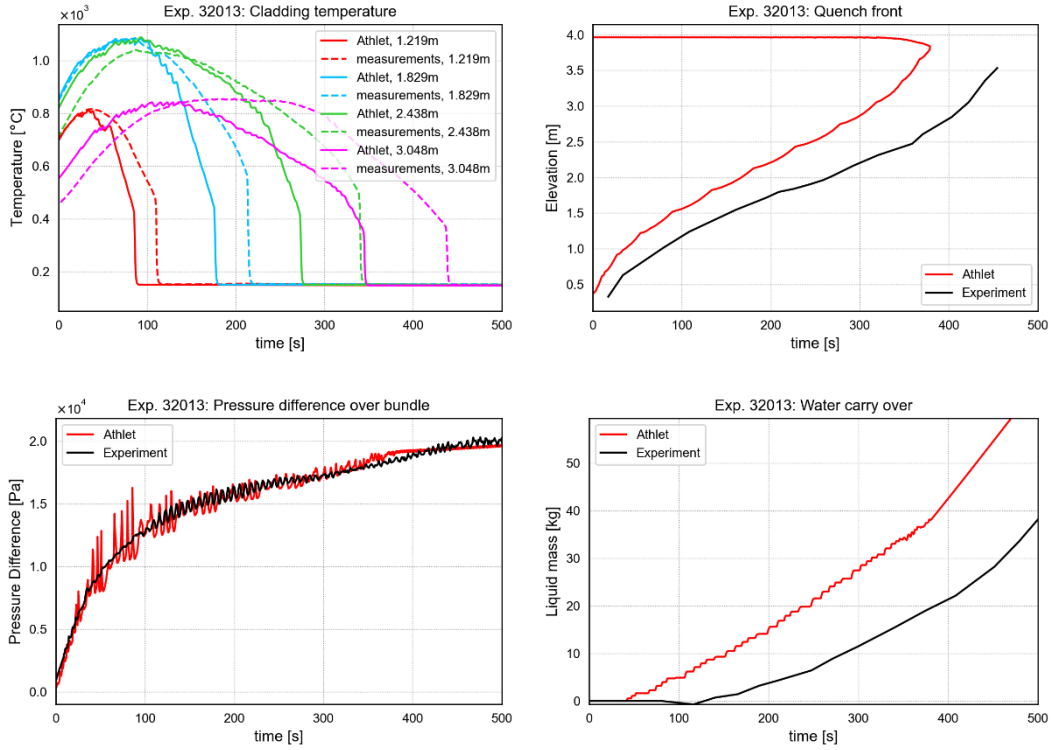

Fig. 4.23 Main calculation results for FLECHT test 32013 and comparison with measurements in the experiment

In the tests with the skewed power profile (Fig. 4.24 to Fig. 4.28), the temperature peak is better reproduced for lower elevations, with the highest error (overestimation) in the highest considered position, in agreement with the expectations for this different power profile. A good agreement in the lower elevations is found in the tests with higher pressure, 13812, 15305, and 15713. The error in the quench time at the highest position can exceed 200 s, with the largest error observed for test 13609, with low pressure and low feedwater temperature.

Other than in the tests with the cosine power profile, the pressure drop and the water carry-over both tend to be underestimated by ATHLET for the test with skewed power profile. The absolute error is however smaller than for cosine power profiles for both quantities.

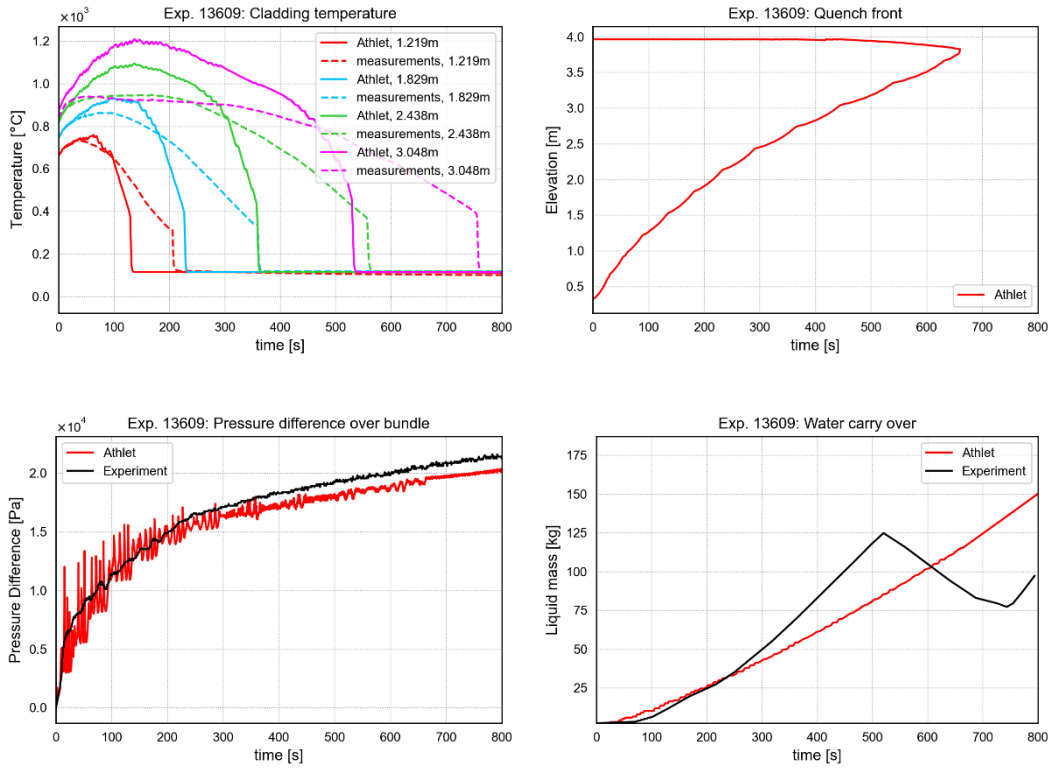

Fig. 4.24 Main calculation results for FLECHT test 13609 and comparison with measurements. Quench front data are not available

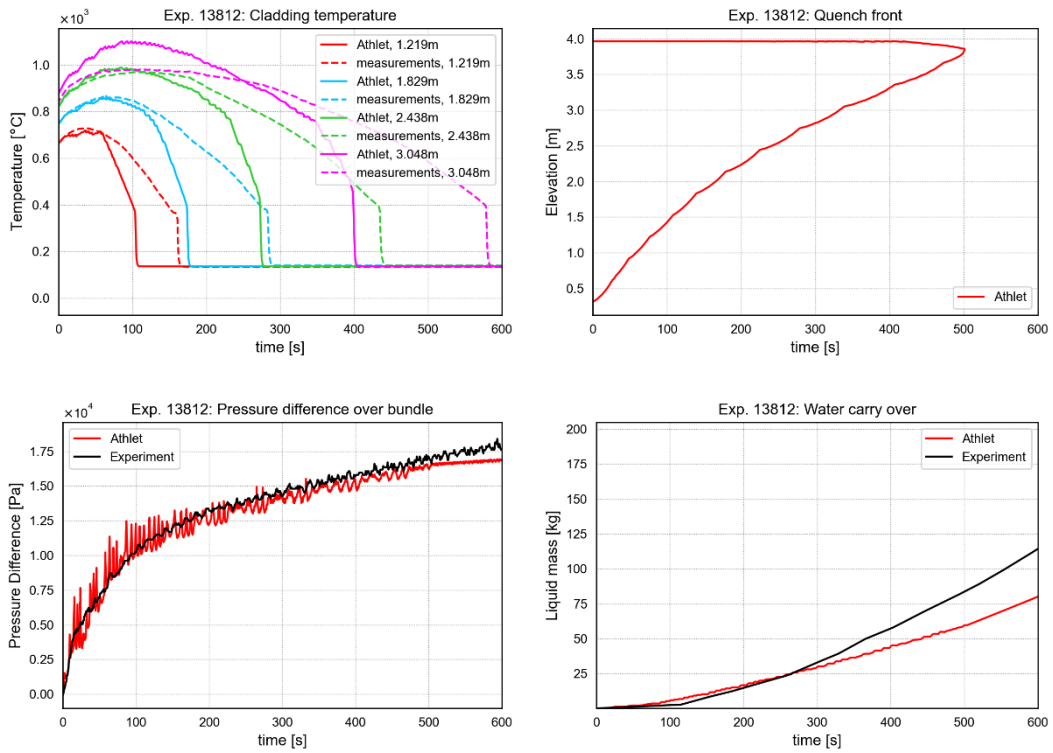

Fig. 4.25 Main calculation results for FLECHT test 13812 and comparison with measurements in the experiment. Quench front data are not available

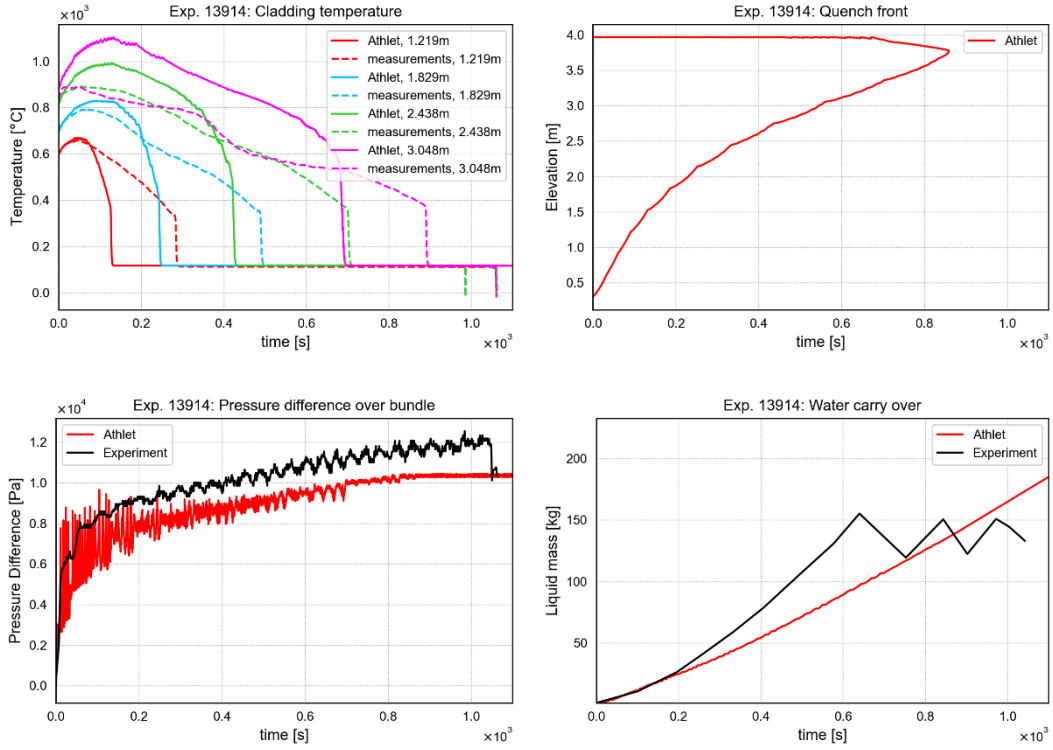

Fig. 4.26 Main calculation results for FLECHT test 13914 and comparison with measurements in the experiment. Quench front data are not available

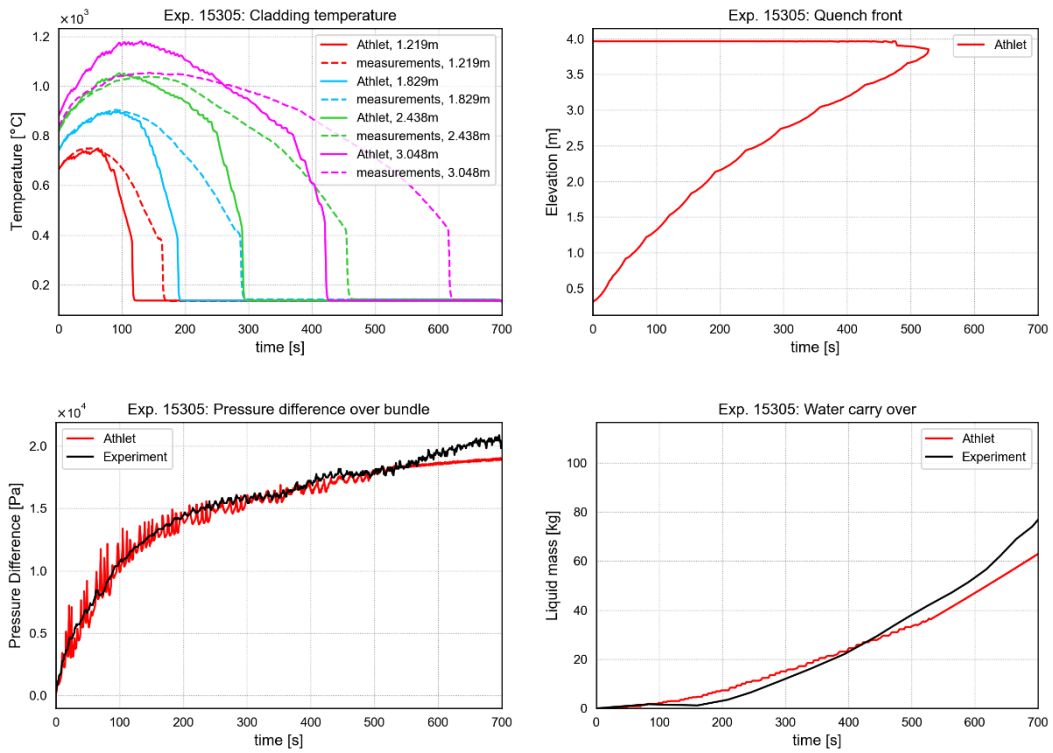

Fig. 4.27 Main calculation results for FLECHT test 15305 and comparison with measurements in the experiment. Quench front data are not available

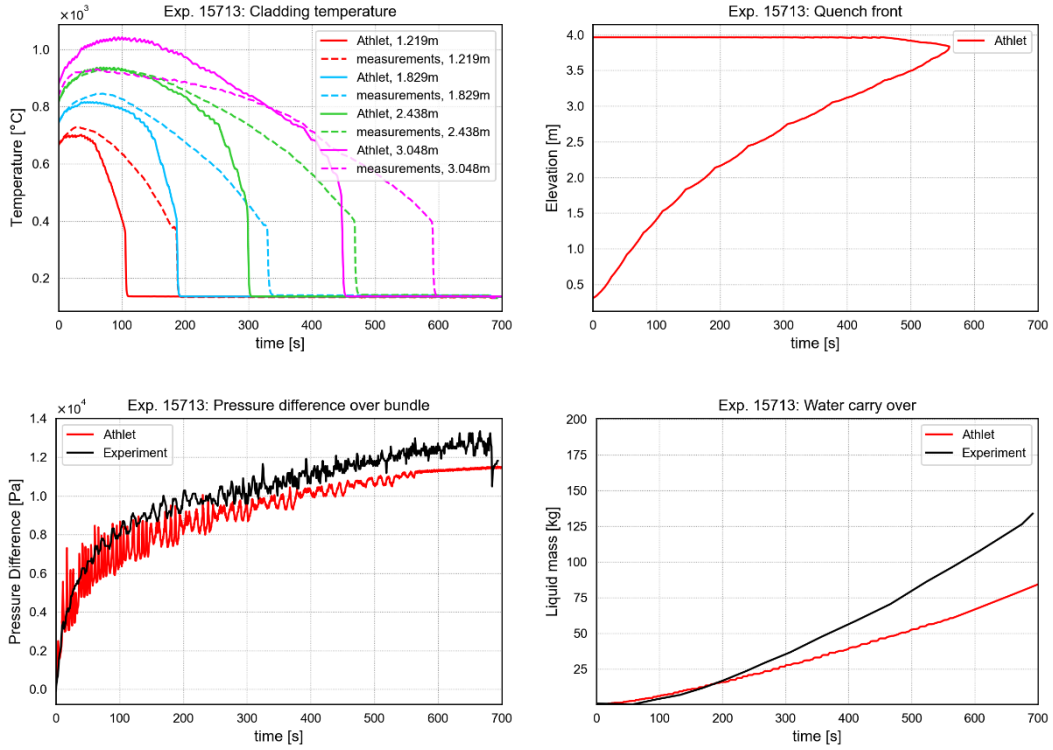

Fig. 4.28 Main calculation results for FLECHT test 15713 and comparison with measurements in the experiment. Quench front data are not available

4.1.3.3 PERICLES

The PERICLES tests vary mainly in system pressure, feedwater subcooling and flow rate, and, as the purpose of the tests was to study the 2D effects, the power factor for the central channel (Tab. 4.3).

Tab. 4.3 Parameters for the PERICLES tests which were recalculated in this study

Run number	Factor F_{xy} for power in central channel	System pressure [bar]	Feedwater mass flux [kg/(m²s]	Feedwater subcooling [°C]	Initial cladding temperature [°C]
062	1	3	36	60	600
064	1.435	3	36	60	600
069	1	3	36	60	475
079	1.435	3	36	90	600
080	1.435	3	50	60	600
086	1.435	4	36	60	600

The power in a standard (lateral) channel is 3180.21 W/m, corresponding to 29300 W/m^2 heat flux. Reflooding is initiated when the given initial cladding temperature is reached in the middle elevation of the rod.

The results of the calculations are shown in Fig. 4.29 to Fig. 4.34. Only a limited amount of experimental data was made available for the PERICLES experiments, namely cladding temperatures and pressure drop in both the central and the lateral assemblies. For some experiment runs, some of these data are missing and thus cannot be shown in the figures. Water carry-over is computed in ATHLET as well; however, because no measurement data are available for this quantity, they are not presented here. Measured quench front data are generally not available for the PERICLES tests.

A noteworthy difference to the results reported before for the FEBA and FLECHT experiments, where ATHLET typically underestimates the quenching time, is the slower quenching by ATHLET in the hot (central) assembly in the PERICLES experiments. In the lateral assemblies, the quench time is typically slightly too large in the lower considered elevation (centre elevation of the rod, with peak power), and too small in the higher elevation (as far as measurement data are available). In the hot assembly, the quenching is too slow at both elevations, and the error is larger for the lower (centre) elevation. Note

that there is no difference in power between central and lateral assemblies in test 062 and 069. The error in quench times at the lower elevation is for all cases in the range 50-60 s, a considerably smaller value is only observed for test 062, with no overheating of the central channel and late (compared to test 069) initiation of reflooding. There is a good agreement, or slight overestimation, in the peak temperature at the lower elevation, while it is underestimated for the higher elevation.

The calculated pressure drops show pronounced oscillations in all cases. In the early (heat-up) phase of the tests, the calculated pressure drop is always too high; in the later phase it is strongly underestimated.

The determination of the cladding temperatures at specified elevations in ATHLET can be inaccurate due to the coarse nodalisation of the HCO representing the rod (Fig. 4.12). Without axial heat conduction, and with the constant heating in this experiment, local axial temperature gradients may be large and not well resolved, thus the temperature in the considered (comparably large) heat conduction volume may not be a sufficiently good approximation of the actual temperature at the requested position.

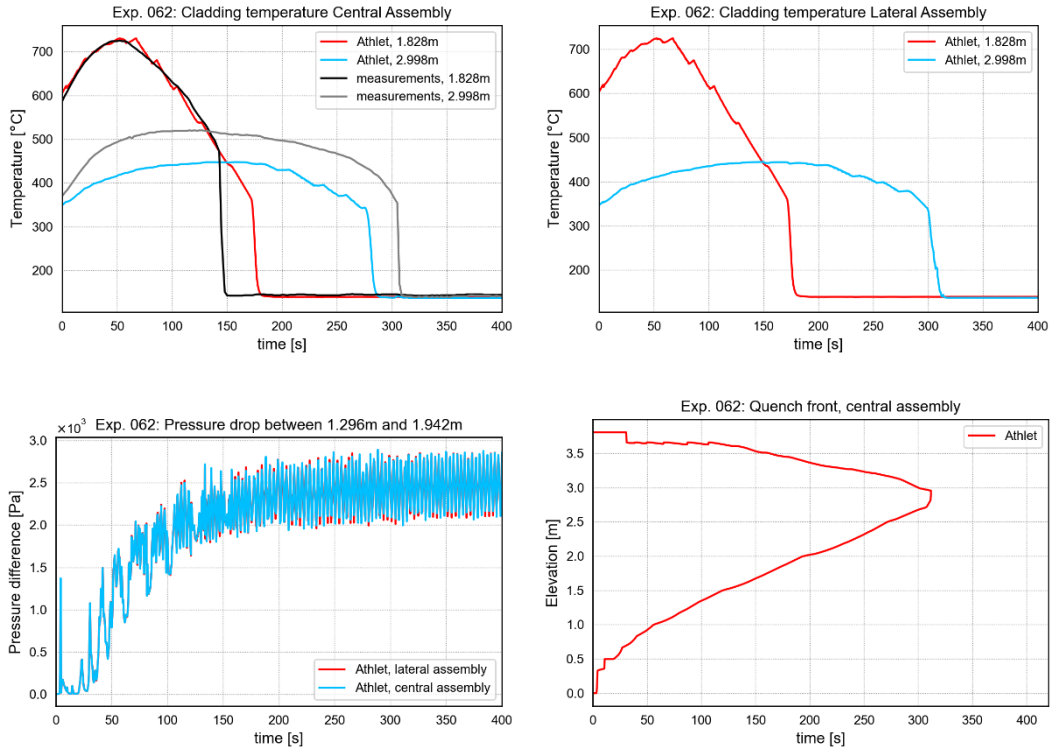

Fig. 4.29 Main calculation results for PERICLES test 062 and comparison with measurements in the experiment (some measurement data missing)

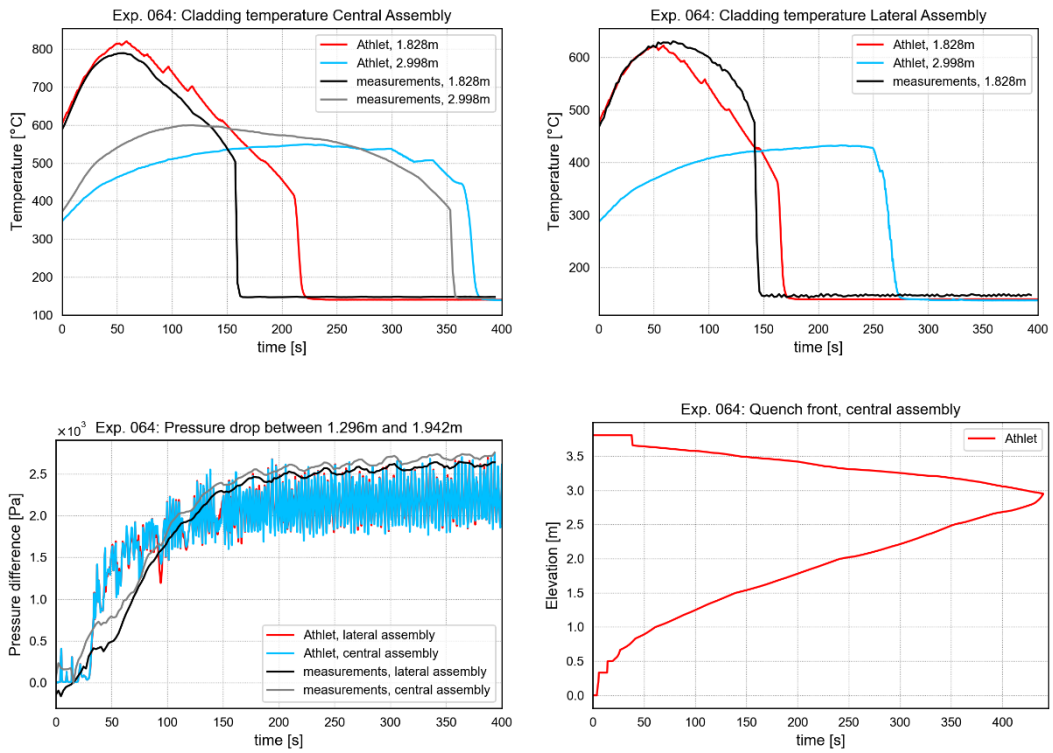

Fig. 4.30 Main calculation results for PERICLES test 064 and comparison with measurements in the experiment (some measurement data missing)

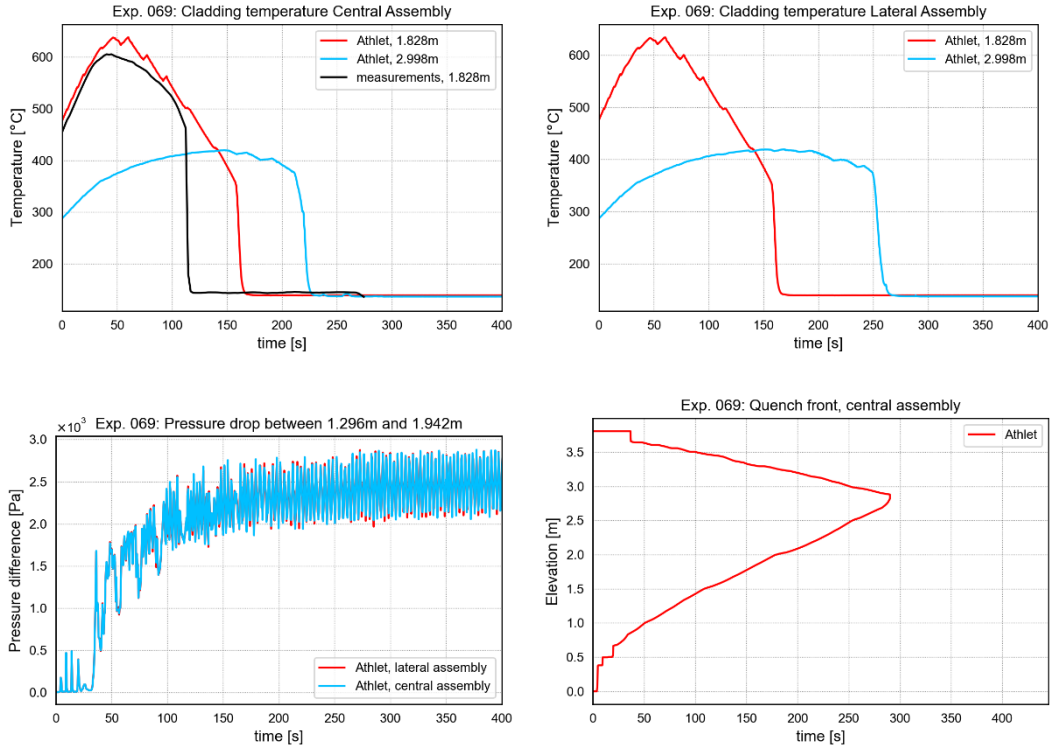

Fig. 4.31 Main calculation results for PERICLES test 069 and comparison with measurements in the experiment (some measurement data missing)

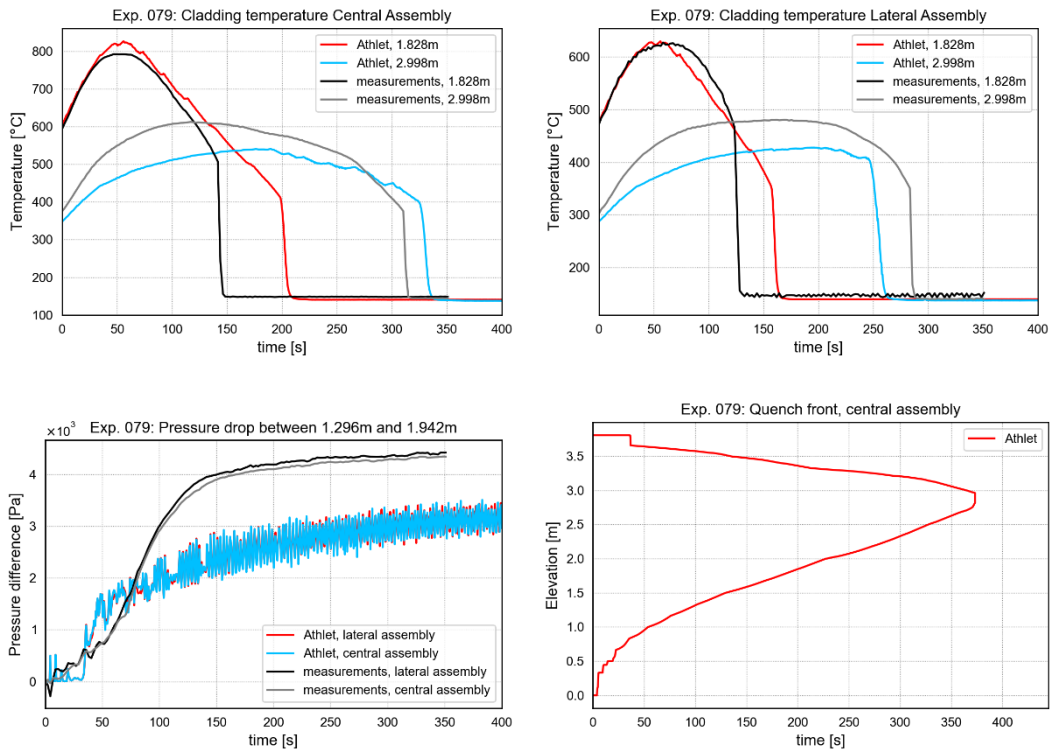

Fig. 4.32 Main calculation results for PERICLES test 079 and comparison with measurements in the experiment

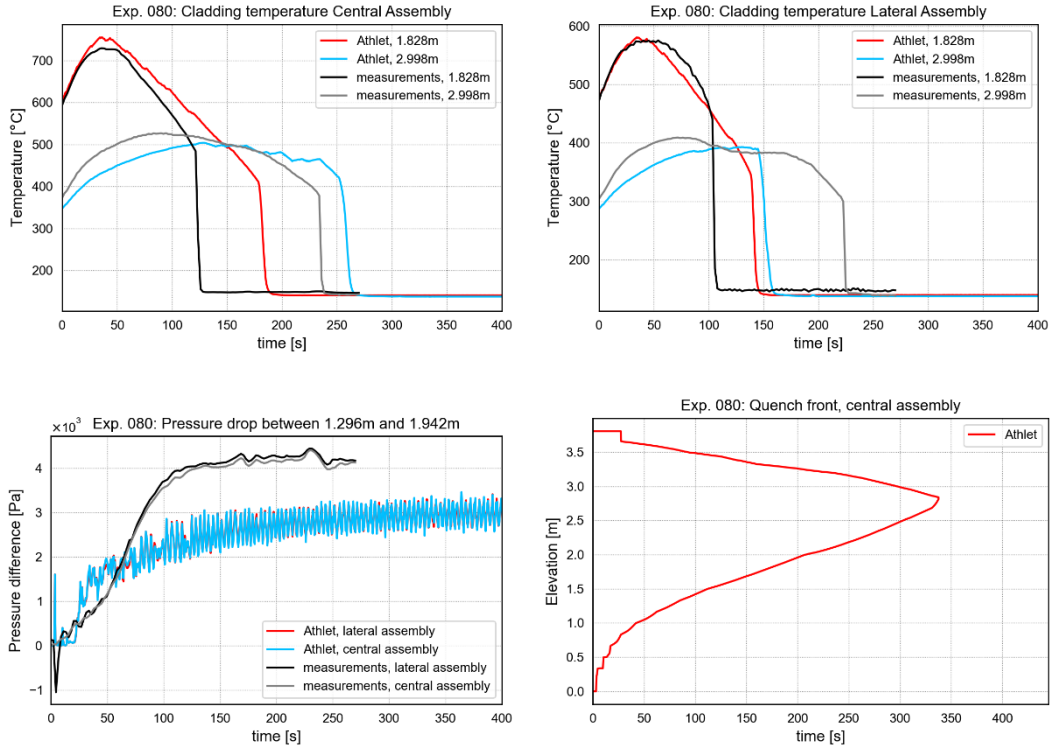

Fig. 4.33 Main calculation results for PERICLES test 080 and comparison with measurements in the experiment

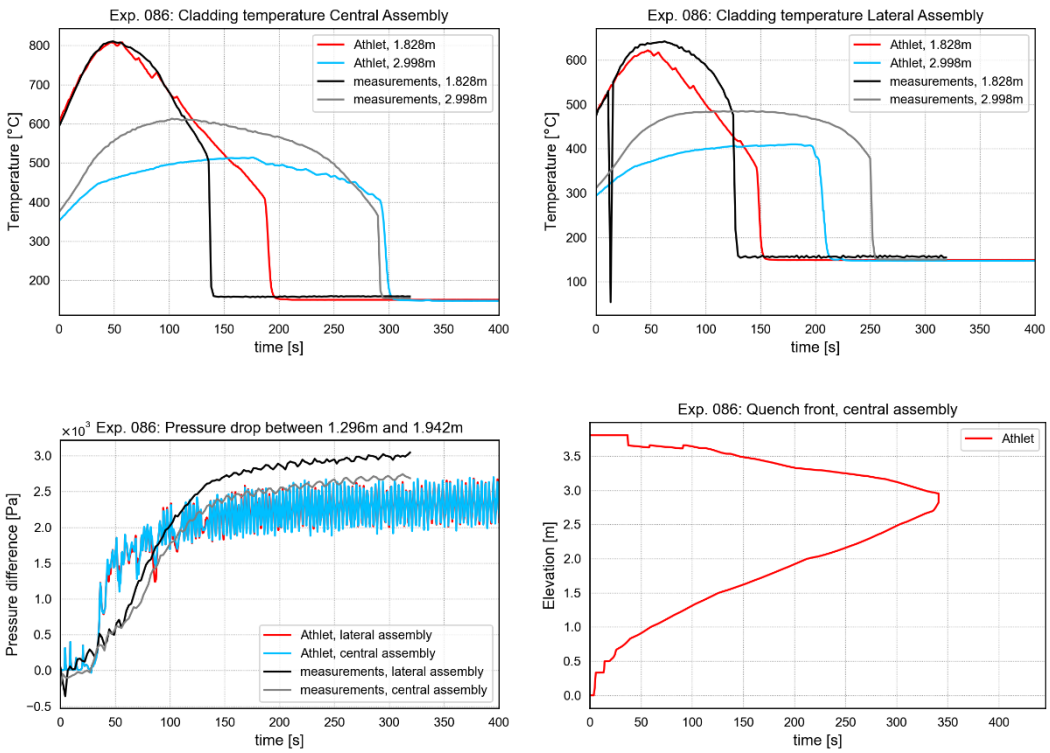

Fig. 4.34 Main calculation results for PERICLES test 086 and comparison with measurements in the experiment

5 Update of ATHLET documentation regarding uncertainties

5.1 Setting sample sizes for the GRS method

5.1.1 The formula of Wilks

The GRS method for uncertainty and sensitivity analysis in the safety analysis /HOF 85/, /GLA 08/ propagates uncertainties related to code model parameters as well as input deck parameters sampled from probability distributions for these parameters through the simulation model. In order to determine distribution-free tolerance limits for one figure of merit (FoM), order statistics and in particular Wilks' formula is used /WIL 41/, /WIL 42/, /SAC 69/. For multiple figures of merit, Wilks' formula can be applied as well to derive distribution-free percentile contours, as shown by /WAL 43/, extended by /TUK 47/, and summarized by /WIL 48/. The application of the GRS method for uncertainty analysis with AC² codes, including ATHLET, is supported by the GRS software SUSA /KLO 20/.

For the set-up of uncertainty analyses, a basic understanding of the main assumptions and mathematical formulas for the application of Wilks' formula to one as well as multiple figures of merit is necessary. Based on that, suitable sample sizes can be determined.

Wilks approach as extended by Tukey makes the premise that the figure of merit F is (or multiple figures of merit F_i are) a random sample, from which N samples are drawn. To obtain these samples for the FoM, which are results of code simulations with, e.g., ATHLET, input uncertainties have to be propagated through the simulation models. Mathematically, the simulation code induces a function A from a suitable (non-empty) set of the input parameter space of n input parameters (over time) to a suitable set of m figures of merit (over time):

$$A\colon X^n \subset \mathbb{R}^n \times \mathbb{R} \to V^m \subset \mathbb{R}^m \times \mathbb{R}, \qquad f^{(m)}(t) = A^{m\times n}(t)x^{(n)}(t) \tag{5.1}$$

For the code results at each point in time, $f^{(m)}(t_0)$, to be a random sample, the input uncertainties $x^{(n)}(t_0)$ have to be random variables at each point in time. In a lot of cases, particularly code model uncertainty parameters are independent of time, i.e., $x_i(t_0) = x_i$ and can be described by a (cumulative) uncertainty distribution, which without loss of generality is strictly monotonously increasing:

$$\phi_i: X_i \subset \mathbb{R} \rightarrow [0,1], \qquad u_j = \phi_i(x_j) \tag{5.2}$$

A random sample for the variable x_i can then obtained by drawing samples from the uniform distribution and computing $x_j = \phi_i^{-1}(u_j)$. This method is, e.g., implemented in the SUSA software /KLO 20/.

An explicit dependence of an uncertain parameter on time should generally be restricted to a time point in a simulation, e.g., the actuation of a safety system, is uncertain.

This allows to draw a few obvious inferences. If samples at high and/or low percentiles of the uniform distribution are excluded, a so-called truncated distribution

$$\phi_{1,\mathrm{tr}}: X_{1,tr} \subset \mathbb{R} \rightarrow [p_l, p_h], \qquad u_j = \phi_1(x_j), \qquad 0 < p_l < p_h < 1 \tag{5.3}$$

is effectively used for the truncated parameter range $[\phi_1^{-1}(p_l), \phi_1^{-1}(p_h)]$. Such a truncated sample is still random, albeit only for the truncated distribution, which can be trivially rescaled to a cumulative uncertainty distribution on its support. Obviously, if the excluded percentiles are large, any rescaled distribution will be substantially different to the uncertainty distribution specified for the uncertain parameter.

If, however, truncations would be performed based on code results for a figure of merit, i.e., for samples $f_j^{(1)}(t_0) = A^{1\times1}(t_0)\phi_1^{-1}(u_j)$, this would generally violate randomness. The reason is that the answer of the code does not depend on a strictly monotonous relation to sample values in the generic case. The dependency might be highly complex, introducing non-random distortions in the sample. (Basically, it will be next to impossible to guarantee that the resulting subset $U^n \subset [0,1]^n x [t_l, t_h]$ is sampled uniformly.)

If multiple uncertain parameters are estimated independently via such uncertainty distributions, a random sample will be obtained by drawing samples from each parameter distribution. Truncating percentiles for each of these distributions individually would still result in a random sample (with different actual uncertainty distributions). Again, disregarding samples based on FoM values would generally result in loss of randomness.

If higher-dimensional submanifolds of $\phi^{(n)}([0,1]^n)$ are excluded, the situation becomes more complex. One example would be a set of three parameters $\{p_1, p_2, p_3\}$ with $p_1 + p_2 + p_3 \leq 1 \ \wedge p_1, p_2, p_3 \in [0,1]$. Sampling from $[0,1]^n$ and rejecting any cases where the above condition is not met would still lead to a random sample. And it would still be possible to rescale the effective distribution continuously to one over $[0,1]^n$. If, however, the allowed parameter space X_C^n would become disconnected (consider a case where the conditions above would augmented by $(p_1 - 0.1)^2 + (p_2 - 0.1)^2 + (p_3 - 0.1)^2 > 0.1$, i.e. we cut out a little ball around the point $\{0.1, 0.1, 0.1\}$), interpreting the result in terms of a reasonable and justifiable uncertainty distribution for physical parameters will be next to impossible. Moreover, it might become impossible to rescale the result to an (inverse) uncertainty distribution over $[0,1]^n$, guaranteeing randomness of the sample. Thus, we will require that the inverse uncertainty distributions (after truncating them) map to a convex (and at least simply connected) subset $U_C^n \subset [0,1]^n$ so that random sampling can be ensured.

Similarly, using non-random sampling techniques for a parameter distribution ϕ_i (or rather ϕ_i^{-1}) such as, e.g., latin hypercube sampling, will lead to loss of randomness. In this case the formula of Wilks does no longer apply (for the specified parameter distributions).

Given a random sample of N realisations of n uncertain input parameters $\{\phi^{(n)}(u_1), \dots, \phi^{(n)}(u_N)\}$, the code at a simulation time t_0 will provide N code results for one FoM $\{A^{(1xn)}(t_0)\phi^{(n)}(u_1), \dots, A^{(1xn)}(t_0)\phi^{(n)}(u_N)\}$. Then, Wilks' formula gives the probability content α for a coverage of s ranked samples ($1 \leq s \leq N$) of the overall N samples at the level of confidence of at least β via /WIL 48/:

$$1 - I_\alpha(s, N - s + 1) \geq \beta \tag{5.4}$$

Here, I_α is the regularized incomplete beta function defined as

$$\begin{aligned} I_\alpha(a,b) &= \frac{1}{B(a,b)} \int_0^x t^{a-1}(1-t)^{b-1} dt \\ &= \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \int_0^x t^{a-1}(1-t)^{b-1} dt \end{aligned} \tag{5.5}$$

with $B(a,b)$ as beta function defined via the gamma function $\Gamma(n)$. Notably $B(a,b) = \int_0^1 t^{a-1}(1-t)^{b-1}dt = \frac{(a-1)!(b-1)!}{(a+b-1)!}$ for integer coefficients a and b, illustrating the connection to the binomial coefficients and distribution. The coverage s applies after ordering the sample by size, i.e.

$$f_1^{(1)}(t_0) \leq f_2^{(1)}(t_0) \leq \cdots < \cdots \leq f_{N-1}^{(1)}(t_0) \leq f_N^{(1)}(t_0) \tag{5.6}$$

and removing $N - s$ of the extreme ranks, i.e. the lowest and/or the largest $N - s$ values for $f_i^{(1)}(t_0)$. The probability content α with $0 < \alpha \leq 1$ according to eq. (5.4) is then located between the values for the remaining ranks, i.e. in the interval $\left[f_{\text{low rank}}^{(1)}(t_0), f_{\text{high rank}}^{(1)}(t_0)\right]$.

The level of confidence β with $0 < \beta \leq 1$ is also called error of the second kind. Notably, the formula is also valid for a coverage of N, i.e., without any ranks being removed. The value of minimum N, for which a given probability content α at a level of confidence β can be obtained according to eq. (5.4) is the minimum sample size for uncertainty analyses with the GRS method, if one-sided confidence intervals are sought.

5.1.2 Wilks' formula for multiple figures of merit

The applicability of Wilks' formula has been extended to multiple figures of merits /WAL 43/, /TUK 47/. The proof by induction requires to remove (at least) one sample of extreme rank, i.e., either the highest sample or the lowest sample, depending on whether a high or low limit has been determined. Consequently, for a two-sided confidence interval for one figure of merit, at least one rank has to be removed /TUK 47/, /SAC 69/. As a recommendation, removing at least one extreme rank on (both) the high and/or low end of the ordering helps to keep two-sided intervals symmetric and ensures that there is at least one sample outside of the confidence bounds. This helps with interpreting uncertainty analysis results.

For multiple FoM, the approach by Wald as extended by Tukey requires removing one additional sample at the high and/or low end of the current ranking and then rank the remaining samples for the next FoM /WAL 43/, /TUK 47/, /POR 19/. Conceptually, we first order

$$f_{1,1}^{(m)}(t_0) \leq f_{1,2}^{(m)}(t_0) \leq \cdots < \cdots \leq f_{1,N-1}^{(m)}(t_0) \leq f_{1,N}^{(m)}(t_0), \tag{5.7}$$

remove the two extreme ranks $1{,}1$ and $1, N$ and then sort $f_{1,2}^{(m)}(t_0) \leq \cdots < \cdots \leq f_{1,N-1}^{(m)}(t_0)$ for the next FoM resulting in

$$f_{2,1}^{(m)}(t_0) \leq f_{2,2}^{(m)}(t_0) \ldots < \cdots \leq f_{1,N-3}^{(m)}(t_0) \leq f_{1,N-2}^{(m)}(t_0) \tag{5.8}$$

and so on, until m FoM are ranked. In the case for m two-sided limits, the remaining coverage is $s = N - 2m$. Importantly, eq. (5.4) for Wilks' formula is still valid, so the result is

$$1 - I_\alpha(N - 2m, 2m + 1) \geq \beta. \tag{5.9}$$

If $2m$ samples are removed following /TUK 47/, the resulting contour in the m-dimensional space of FoM is a hyperrectangle, where the interval in each dimension is given by the maximum and minimum of the set of $N - 2m$ remaining FoM samples $\{f^{(m)}\}$, i.e.

$$V_C^m = [\min\{f_1^m\}, \max\{f_1^m\}]_1 \times \ldots \times [\min\{f_m^m\}, \max\{f_m^m\}]_m \subset \mathbb{R}^m \tag{5.10}$$

Note that the bounds for the first $m - 1$ dimensions are not necessarily given by the ranks remaining after ranking for the first dimension (and so on). This is due to the fact that a later removal of a sample could remove the extreme sample for a FoM that was treated already before. Wilks' formula does not impose any restrictions on the order in which FoM are sorted. However, the ranks not removed for FoM with $m > 1$ will generally depend on that order. Thus, the specific extent of V_C^m does depend on details of the ordering process.

As long as for each upper or lower limit of a FoM at least one extreme rank is removed, there is no formal restriction on any removal of further ranks. The formula applies to any coverage, irrespective of the ranking scheme. Removing extreme ranks for two-sided bounds symmetrically and removing the same number of ranks for each upper or lower FoM confidence limit is, however, an obvious and sensible approach. If additional ranks

would be removed (a lower coverage), it would be possible following /TUK 47/ to constrain the contour V_C^m to other shapes, i.e. convex polytopes with $V_{C,c}^m \subset V_C^m$.

5.1.3 Determining confidence intervals for multiple FoM vs. contours for a set of FoM

The sample of code results $A^{M\times n}(t_0)$ can be used to compute confidence intervals for multiple FoM separately, i.e., by ordering first the results for FoM $f^1(t_0)$ and the resulting confidence interval and then by ordering the same sample for FoM $f^2(t_0)$ and determining the confidence interval. While this is allowed under the premises of eq. (5.4), there is then no possibility to infer a valid statement as to the confidence intervals for FoM $f^1(t_0)$ and $f^2(t_0)$ being fulfilled simultaneously with the specified level of confidence. For example, if the FoM would be the maximum primary side pressure and the maximum cladding temperature and the analysis would show that the confidence interval for the primary pressure is below its acceptance value and that the confidence interval for the peak cladding temperature is below its acceptance value, it does not follow the tuple of values (maximum primary pressure, peak cladding temperature) will be jointly below both acceptance values with the specified level of confidence.

In order to obtain valid results for multiple FoM with the specified level of confidence, the sample has to be ordered sequentially by the first FoM, extreme ranks have to be removed, then the sample has to be re-ordered for the second FoM, extreme ranks have to be removed, and so on (see 5.1.2).

Consequently, statements about multiple FoM being constrained simultaneously require a larger size of the sample, a systematic ranking approach and a coverage that is smaller than the sample size, see also eq. (5.9).

Notably, applying Wilks' formula to multiple FoM is still distribution free /WAL 43/, /TUK 47/, and we do not need to make any assumptions on the uncertainty distributions as well as the FoM. While /POR 19/ implies that FoM should be uncorrelated, this restriction is not formally necessary. For example, consider two completely (rank-) correlated FoM, where $f^2 = \lambda f^1$. Following the approach above for a given level of confidence for two-sided limits would result in removing twice $2m$ ranks for the same rank ordering. This will require a minimum sample size of N_2. As this is effectively a situation with one FoM, we know that we would only need to remove $2m$ ranks to meet the level of

confidence target, but at a much smaller sample size N_1. Thus, we have generated more samples and have a better constraint on the FoM than strictly needed, but Wilks' formula still holds. This example also illustrates that having uncorrelated FoM will be more effective (and lead to more efficient constraints) than using correlated FoM. This is hardly surprising, if the (rank-) correlation between FoM is known, this is additional information about the underlying uncertainty distribution. And any additional information about the distribution can be used to devise (at least theoretically, and often also practically) a more effective approach for non-distribution free contours than those available via Wilks' formula. /POR 19/ gives further examples for this insight.

5.1.4 Properties of the regularized incomplete beta function

For the following, a few properties of the $I_\alpha(a,b)$ are helpful /GRA 81/, /NIST 24/. These can be derived from its defining equation (5.5).

$$I_1(a,b) = 1 \tag{5.11}$$

$$I_\alpha(a,1) = \alpha^a \tag{5.12}$$

$$I_\alpha(1,b) = 1-(1-\alpha)^b \tag{5.13}$$

$$I_\alpha(a,2) = I_\alpha(a,1) + \frac{\alpha^a(1-\alpha)}{B(a,1)} = \alpha^a\big(1+a(1-\alpha)\big) \tag{5.14}$$

$$I_\alpha(2,b) = I_\alpha(1,b) - \frac{\alpha^1(1-\alpha)^b}{B(1,b)} = 1-(1-\alpha)^b(1+b\alpha) \tag{5.15}$$

$$I_\alpha(a,b) = 1 - I_{1-\alpha}(b,a) \tag{5.16}$$

$$I_\alpha(a,b) = \alpha I_\alpha(a-1,b) + (1-\alpha) I_\alpha(a,b-1) \tag{5.17}$$

For integer coefficients $N, k = N - s$, this can be expressed in terms of binomial coefficients:

$$I_{1-\alpha}(s, N-s+1) = I_{1-\alpha}(N-k, k+1) = \sum_{i=0}^{k} \binom{N}{i} \alpha^i (1-\alpha)^{N-i}. \tag{5.18}$$

Consequently, the formula of Wilks for the ranks is closely related to the Bernoulli distribution – this underpins Wilks' proof /WIL 42/ - with probability α, and we find for the number of ranks below or at the threshold k

$$\Pr(X_1 \leq k) = I_{1-\alpha}(N-k, k+1). \tag{5.19}$$

5.1.5 Symmetry of upper limits and lower limits

As $I_\alpha(a, b)$ describes a cumulative distribution, eq. (5.4) applied to a one-sided (upper) tolerance bound will provide the maximum rank (or minimum sample size) so that the probability content is covered. It will thus generally provide an upper limit to any percentile. If we are interested in the lower limit, the symmetrical condition must be that at most a probability content of $(1-\alpha)$ is in the lower tail and that the cumulative distribution does not exceed $(1-\beta)$, see eq. (5.19). We thus obtain the following:

$$\begin{aligned} & I_{1-\alpha}(s, N-s+1) && \leq 1-\beta \\ \Leftrightarrow \quad & 1 - I_\alpha(N-s+1, s) && \leq 1-\beta \\ \Leftrightarrow \quad & I_\alpha(N-s+1, s) && \geq \beta \\ \Leftrightarrow \quad & 1 - I_{1-\alpha}(s, N-s+1) && \geq \beta \end{aligned} \tag{5.20}$$

Upper and lower limits can be deduced with the same formula, as already stated for eq. (5.4), just by looking at the lowest ranks.

5.1.6 Tables for determining sample sizes

For planning an uncertainty study, knowing the required sample size is essential. Below, we provide a table with minimum sample sizes N for the pair $(\alpha \,/\, \beta)$. Given the number of ranks removed $k = N - s$, the minimum sample size can be looked up. We give values

in Tab. 5.1 for probability contents from 0.5 up to 0.9999 with confidence levels 0.9 to 0.99. In order to illustrate the consequences of a high number of figures of merit or of trying to substantiate very high success probabilities – corresponding to very low failure probabilities – the table includes a wide range of probability contents and coverages s.

Obviously, larger sample sizes (which can be driven by higher rank orders) will increase the statistical quality of the analysis both for the location of the confidence intervals as well as for other statistical information that can be obtained from the samples, like, e.g., location of percentiles, sensitivity indices, etc. Any added value has to be judged against the additional resources (time, computer power, etc.) to perform such additional calculations.

Tab. 5.1 Minimum number of calculations N for statistical tolerance limits at different coverages s

	Tolerance limits (α / β)									
$N - s$	**0.50/0.90**	**0.68/0.90**	**0.90/0.90**	**0.90/0.95**	**0.95/0.95**	**0.95/0.99**	**0.99/0.95**	**0.99/0.99**	**0.999/0.95**	**0.9999/0.95**
0	4	6	22	29	59	90	299	459	2995	29956
1	7	11	38	46	93	130	473	662	4742	47437
2	9	15	52	61	124	165	628	838	6294	62956
3	12	19	65	76	153	198	773	1001	7752	77535
4	14	23	78	89	181	229	913	1157	9151	91533
5	17	27	91	103	208	259	1049	1307	10511	105128
6	19	31	104	116	234	288	1182	1453	11840	118422
7	21	35	116	129	260	316	1312	1596	13146	131479
8	24	38	128	142	286	344	1441	1736	14432	144344
9	26	42	140	154	311	371	1568	1874	15702	157049
10	28	46	152	167	336	398	1693	2010	16959	169619
11	31	49	164	179	361	425	1818	2144	18204	182072
12	33	53	175	191	386	451	1941	2277	19439	194422
13	35	57	187	203	410	478	2064	2409	20665	206682
14	37	60	199	215	434	504	2185	2539	21883	218861

	Tolerance limits (α / β)									
$N - s$	**0.50/0.90**	**0.68/0.90**	**0.90/0.90**	**0.90/0.95**	**0.95/0.95**	**0.95/0.99**	**0.99/0.95**	**0.99/0.99**	**0.999/0.95**	**0.9999/0.95**
15	39	64	210	227	458	529	2306	2669	23094	230968
16	42	67	222	239	482	555	2426	2798	24298	243008
17	44	71	233	251	506	580	2546	2925	25495	254989
18	46	74	245	263	530	606	2665	3052	26688	266914
19	48	78	256	275	554	631	2784	3179	27875	278788
20	51	81	267	286	577	656	2902	3304	29058	290616
30	72	116	379	402	809	901	4064	4533	40686	406900
40	94	150	490	515	1036	1139	5201	5727	52064	520688
50	115	183	599	627	1260	1372	6323	6898	63281	632865
100	221	349	1134	1172	2353	2504	11799	12571	118071	1180784
200	428	675	2184	2236	4485	4691	22475	23527	224862	2248732
300	634	998	3223	3286	6587	6835	32995	34262	330080	3300928
500	1043	1640	5284	5365	10749	11065	53819	55427	538358	5383746
1000	2060	3233	10397	10508	21045	21483	105334	107569	1053577	10536005

5.1.7 Time dependent results and definition of figures of merit

The code results of a simulation code like AC²/ATHLET will be a time series of data. If we for now assume for notational effectiveness that each FoM is a code results, we will get a set of time series data for each sample calculation.

$$\left\{\left\{f^{(m)}(t_{l,1}),\dots,f^{(m)}(t_{h,1})\right\}_1,\dots,\left\{f^{(m)}(t_{l,N}),\dots,f^{(m)}(t_{h,N})\right\}_N\right\} \tag{5.21}$$

Importantly, the sample size for the FoM is determined by the number of samples drawn for the set of uncertain parameters $x^{(n)}$. We assumed above that for each set of parameters, a sample for each constituent parameter has been drawn once and then used throughout the calculation. Consequently, each set of time series data represents one sample for the purpose of eq. (5.4). Therefore, if we order the set of time series data for a FoM $f_1^{(m)}$ at time t_0 with $t_0 \in [t_l, t_h]$, we have to determine the nearest time point $t_{0,i}$ (these generally are not identical between simulations) in each time series and order the set as

$$f_{1,1}^{(m)}(t_{0,i_1}) \leq f_{1,2}^{(m)}(t_{0,i_2}) \leq \cdots < \cdots \leq f_{1,N-1}^{(m)}(t_{0,i_{N-1}}) \leq f_{1,N}^{(m)}(t_{0,i_N}). \tag{5.22}$$

Now, if we would order the same time series data for the same FoM $f_1^{(m)}$ at time t_1 with $t_1 \in [t_l, t_h]$ and $t_1 \neq t_0$, based on the general properties of the simulation model $A^{(m,n)}([t_l, t_h])$, this would not always result in the same order of the ranks. Consequently, a FoM at a different time t_1 needs to be treated as a different FoM for the purpose of eq. (5.4). For a one-sided confidence limit, one sample has to be removed from the coverage. That would require to remove the whole time series $\left\{f^{(m)}(t_{l,i}),\dots,f^{(m)}(t_{h,i})\right\}_i$ containing the extreme rank $f_{1,N}^{(m)}(t_{0,i_N})$.

If, however, a scalar FoM is derived based on the time series, e.g. through a representative value over the time series, such as, e.g., its minimum, its maximum or mean value, the time series data are projected to a vector of size N. For the maximum of $f_1^{(m)}$ we get

$$\left\{\max_t\left(f_{1,1}^{(m)}(t_{l,1}),\dots,f_{1,1}^{(m)}(t_{h,1})\right),\dots,\max_t\left(f_{1,N}^{(m)}(t_{l,N}),\dots,f_{1,N}^{(m)}(t_{h,N})\right)\right\} \tag{5.23}$$

This can be ranked easily, which allows to consider information over a whole time interval (up to the full calculation) within one FoM.

A simulation model consists of numerous control volumes and heat structure elements, which produce several output quantities like, e.g., local pressure, temperature, or heat fluxes. Instead of controlling, e.g., the fluid temperatures $T_{fl}(c_1,t_0),\dots,T_{fl}(c_k,t_0)$, in k control volumes as FoM individually, we can use, e.g., the maximum of those temperatures over the relevant control volumes as one FoM and look at

$$\left\{\max_c\left(f_{1,1}^{(m)}(c_1,t_o),\dots,f_{1,k}^{(m)}(c_k,t_0)\right),\dots,\max_c\left(f_{1,1}^{(m)}(c_1,t_0),\dots,f_{1.k}^{(m)}(c_k,t_0)\right)\right\} \tag{5.24}$$

Finally, it is also possible to combine such aggregations both on model elements like several control volumes and over a time period. Generally, all code outputs of the simulation model $A^{(M,n)}(t)$ can be considered.

In order to obtain a FoM that can be used to obtain consistent rankings throughout the calculation, the scalar function g used to aggregate over code k outputs should comply with the following conditions.

$$g\colon V^k\subseteq(\mathbb{R}^M\times\mathbb{R})^k\to\mathbb{R},\qquad y=g\left(A^i(c_1,t_{j_1}),\dots,A^i(c_k,t_{j_k})\right) \tag{5.25}$$

$$\begin{aligned} g\left(A^i(c_1,t_{j_1})\right)\le g\left(A^i(c_1,t_{i_1}),A^i(c_2,t_{i_2})\right)\le\cdots \\ \le\ g\left(A^i(c_1,t_{j_1}),\dots,A^i(c_k,t_{j_k})\right) \end{aligned} \tag{5.26}$$

$$A^i(c_1,t_1)<A^i(c_2,t_2)\Rightarrow g\left(A^i(c_1,t_1)\right)\le g\left(A^i(c_2,t_2)\right) \tag{5.27}$$

$$A^i(c_1,t_0) < A^i(c_2,t_0) < A^i(c_3,t_0)$$
$$\Rightarrow g\left(A^i(c_1,t_0),A^i(c_2,t_0)\right) \leq g\left(A^i(c_1,t_0),A^i(c_3,t_0)\right) \quad (5.28)$$

$$A^i(c_0,t_1) < A^i(c_0,t_2) < A^i(c_0,t_3)$$
$$\Rightarrow g\left(A^i(c_0,t_1),A^i(c_0,t_2)\right) \leq g\left(A^i(c_0,t_1),A^i(c_0,t_3)\right) \quad (5.29)$$

The set of code outputs that are aggregated via g should lead to a monotonously increasing function with regard to individual input values, number of objects considered and the length of the time period. This will ensure that the ranking of samples obtained on this FoM is internally consistent. Obviously, if g is monotonously decreasing, this would also lead to consistent results, if the ranking is performed by ordering from the highest to the lowest results. Note that for Wilks' formula to be valid, there are no formal restraints on g (except, perhaps, that it should not be a constant function), but for interpreting the results of such a FoM, a consistent ordering of ranks and a clear relationship between the underlying code results and the FoM is essential. Note also, that the aggregation function might also be defined as $g: V^k \to \mathbb{Z}$.

The aggregation for such a FoM could, theoretically, be done by postprocessing the code output data. However, AC²/ATHLET provides several output variables like, e.g., total core power or maximum cladding temperature, that already do such an aggregation. Beyond these, a GCSM model should be implemented in the input deck, which computes the aggregation operations for the specified FoM and writes it into the plot data. This facilitates working with tools like SUSA significantly.

Finally, we observe that the ranking operation per FoM can be described by a function (a partial order relation)

$$R: V^N \to \mathbb{N}^N,\ \{r_1,\dots,r_N\} = R\left(f_{i,j_1}^{(m)},\dots,f_{i,j_N}^{(m)}\right) \text{ with } f_{i,j_1}^{(m)} < f_{i,j_2}^{(m)} \Rightarrow r_1 < r_2. \quad (5.30)$$

5.2 Percentile bounds

Tab. 5.1 helps with determining the minimum sample size for the specified analysis. This then allows to determine the rank for the corresponding percentile as well (assuming one-sided limits). If other sample sizes are used, or if upper or lower bounds for other percentiles are sought, this can be calculated using eq. (5.4) as well.

For univariate distributions and percentiles corresponding to $\alpha \geq 0.5$, a naïve estimator of the percentile is (obviously) a coverage $s = \lceil \alpha\, N \rceil$. For the correct answer at confidence level β we will have to modify the coverage by an offset k. This results in

$$1 - I_{\alpha}(\lceil \alpha\, N \rceil + k, \lfloor (1-\alpha)N \rfloor - k + 1) \geq \beta \tag{5.31}$$

If we are interested in percentiles $\alpha \leq 0.5$ and as we are faced with a cumulative distribution, the approach for a lower limit (see section 5.1.5) would be

$$\begin{aligned} & I_{1-\alpha}(\lceil (1-\alpha)N \rceil + 1 + l, \lfloor \alpha\, N \rfloor - l) && \leq 1 - \beta \\ \Leftrightarrow \quad & 1 - I_{\delta}(\lceil \delta\, N \rceil + (1+l), \lfloor (1-\delta)\, N \rfloor - (l+1) + 1) && \leq \beta \end{aligned} \tag{5.32}$$

By transferring the lower percentiles to $\delta = 1 - \alpha$ and considering that the lower percentile offset is $l = k - 1$, which needs to be subtracted from the floor of $\lfloor \alpha\, N \rfloor$, the offsets for higher percentiles can be matched to the offsets of the lower percentiles. Obviously, eq. (5.31) can be used to estimate upper limits for the lower percentiles, and eq. (5.32) can be used to estimate lower limits for the higher percentiles.

In the tables below, the offset values for k are given for relevant percentiles of 0.32, 0.5, 0.68, 0.8, 0.9, 0.95 and 0.999 for a confidence level of 0.95, for selected N. Note that these off-sets switch between a higher and a lower value multiple times with increasing N because of the rounding procedure to arrive at integer coefficients for the incomplete normalized beta function and the threshold of β, which can be narrowly missed or exceeded. This is not readily evident from the table.

Tab. 5.2 Offset values for upper limits of naïve percentile estimator for different samples sizes and percentiles at confidence level 0.95

Size	**Offset for percentile α to upper limit at β = 0.95**						
N	**0.32**	**0.5**	**0.68**	**0.8**	**0.9**	**0.95**	**0.99**
59	7	7	6	5	4	2	N/A
93	8	8	7	7	5	3	N/A
124	9	10	9	7	6	4	N/A
153	11	11	9	8	7	4	N/A
181	11	12	10	9	7	5	N/A
208	12	13	11	10	7	5	N/A
220	12	13	12	11	8	6	N/A
240	13	14	12	11	8	6	N/A
260	13	14	13	11	9	6	N/A
300	14	15	14	12	9	7	3
350	15	16	15	13	10	7	3
400	16	17	16	14	11	8	4
450	17	18	17	15	11	8	4
500	18	19	18	16	12	9	4
550	19	20	19	16	12	9	4
600	20	21	20	17	13	10	5

For multivariate distributions, as explained for coverages above, the removal of samples for individual FoM and order in which FoM are ranked is subject to choices by the analyst. Thus, there is no single answer as to how these off-sets should be distributed to the FoM.

5.3 Treatment of code crashes

In uncertainty analyses with AC²/ATHLET, it can happen for a specific sample of input uncertainties, $x_j^{(n)}$, that the code calculation does not start at all or does not finish successfully. The question is then how to treat such a sample $A^{(m,n)}([t_l, t_h])x_j^{(n)}$ Obviously, if the simulation did not reach the time t_0 required for FoM $f_i^{(m)}(t_0)$, it is impossible to subject it to ordering and thus it cannot be used for Wilks' formula /POR 19/. If the simulation did not reach the maximum time needed for the set of FoM, it cannot support the analysis of all FoM included in the study. The effective sample size then has to be reduced by the k failed calculations, unless these can be redone successfully. If the

uncertainty analysis has been performed with the minimum number of samples N for the specified confidence level, see also Tab. 5.1, then eq. (5.4) can no longer be fulfilled. Obviously, inferences at a lower level of confidence β_1 or a lower probability content α_1 might still be possible if proven that the remaining samples are not only still random, but also representative of the adequately defined pdfs of all uncertain input parameters.

If a simulation for a sample does not succeed and we cannot simply repeat the calculation with the original settings, the most important question is then, if the remaining samples can still be used to derive confidence intervals. As the main premise for Wilks' formula is that the sample must be random, the question to answer is thus if removing the failed calculation(s) from the sample results still in a random sample or if the sample is distorted or biased making it non-random.

As pointed out in /POR 19/, applying Wilks' formula to a sample with crashed calculations might not be possible. It is implied that any crashed calculation would lead to a non-random sample. In the following, we explain why this is not always the case, and under which conditions an analysis with failed calculations can still be used and complemented.

If we add or subtract a randomly drawn sample $A^{(m,n)}([t_l, t_h])x_j^{(n)}$ using the original pdfs for the analysis, the resulting sample will still be random, and Wilks' equation will still be applicable.

Now, why would a failure of a simulation be random for this purpose, so that we can simply remove it from the sample? Examples relevant to ATHLET include:

- The simulation fails because of scheduled updates, shutdowns, loss of file server, power blackouts, etc., affecting the computer used for the simulation, if for some reason a restart calculation is not possible (see also section 5.3.1).
- Specific settings of input parameters are not accepted during input processing, e.g., in ATHLET the initial value for a mixture level must not be exactly at a CV boundary. A very minor change of the input value allows a successful recalculation.
- A run-time error, e.g., by leaving the ATHLET property package range or by running into a division by zero in some model, is triggered by complex race conditions. Valid examples would be spurious water hammers and pressure waves sometimes predicted by ATHLET for condensation of strongly subcooled vapour. A minor change

of unrelated numerics parameters under CW `INTEGRAT`, e.g., `EPS, ECKSCH` or the precision settings under KW `DRUFAN`, allows a successful recalculation.

Note that for the last case, a thorough analysis is needed to conclude that the failure of the calculation is indeed a spurious event that randomly occurs due to ATHLET's integration procedure. In addition, such cases point to a substantial weakness in the code and should trigger a bug report to the ATHLET development team.

A more problematic case is if uncertain model values, which are sampled, cause a simulation failure. This could be run-time errors or simulation stops due to too small time steps. Assuming that just one input parameter value causes the problem, we can observe the following:

- The failed simulations effectively truncate the underlying probability distributions (e.g., if the tail of the input uncertainty distribution exceeding a cut-off value is removed, like $]\phi_1^{-1}(p_h), 1]$ and all samples from this tail). The argument here rests on the assumption that we exclude a whole interval of the input distribution; just removing one point for the failed sample would not work. As explained above, the resulting sample after pruning the failed calculations is still a random sample (of code results) and could potentially be used for Wilks' formula. Whether it is representative of the desired pdf associated to the u.i.p in question needs to be investigated independently and confirmed before being able to apply it in the original setting of the problem.
- Wilks' formula does not even require that the samples, which are removed, are removed from the tails of the distribution so that $[\phi_1^{-1}(p_l), \phi_1^{-1}(p_h)]$ would be a contiguous interval. As long as the surviving samples are still random, there is no problem with Wilks' formula itself[2].

[2] A more formal version of the argument here observes that the sampling approach chosen here will ensure that any subset of the input parameter space will be sampled randomly. Assuming a large number of samples, we can (conceptually) remove small intervals, with a length much larger than the limiting numerical precision, around the crashed samples. We will have to accept that such an exclusion might remove parameter combinations, which would not crash. We would also have to remove successful samples if not separable from these intervals by a sufficiently large interval of their own. For a finite number of such (possibly overlapping) intervals (and small compared to the overall sample size), the remaining (non-contiguous) subset of the parameter interval will be sampled randomly. For more than one dimension, we need further assumptions on the removed subsets of the parameter space, e.g., that the be simply-connected subsets with the same dimension as the parameter space.

- However, if we remove samples, we end up with an effective uncertainty distribution for the parameter that is different to the one specified for the analysis. So, we get confidence intervals for a different question.

Against this background, the important question is: When would that be acceptable, so that we can replace failed calculations with new samples? Generally, it is not, and this is what, e.g., /POR 19/ warns against. The default approach should be to raise this as a code capability issue with the ATHLET development team, as these failures might indicate code performance issues or other code errors and redo the analysis. The few limited exceptions from the rule above require additional analysis and justification. This could pertain, e.g., to the following aspects, which should apply simultaneously:

- The uncertainty distribution specified for a model parameter leads to clearly unphysical results (e.g., a negative heat transfer coefficient or other unreasonable parameters). By rejecting these values, the effective distribution is limited to computational accepted values, for which the sample calculations were successful.
- It is possible to specify a revised model uncertainty distribution, which describes the actual distribution used for the analysis. This distribution is reasonable, and it can be justified that it is a suitable distribution for the analysis.
- A sufficiently small fraction of calculations failed so that redoing the analysis with an updated uncertainty distribution might not be proportionate.
- Combinations of uncertain parameters leading to simulation failures can be excluded as reasonable conditions by independent information, e.g., by determinations of uncertain parameters in the literature.
- The simulation failures are not due to input deck errors or other mistakes in setting up a simulation model with ATHLET.

In short: where a convincing argument can be made, that the effective distributions after removing samples are actually the appropriate distributions for the analysis, the analysis can be performed with the pruned sample. And, in case that the sample size needs to be increased, it would be possible to do this by adding samples drawn from the *original* distributions and with the original input deck. Note that this argument refers to the input uncertainty distributions, so that the remaining sample is still a random sample of the effective distributions. As stated above, looking at specific code results (i.e. FoM) will not

guarantee that the sample is still random. If such an argument cannot be made, repeating the whole analysis remains the only viable option.

At the next opportunity, the uncertainty distributions for model parameters should be updated to better values and distribution shapes, based on insights from the analysis and available (experimental) data. For uncertainty analyses – as for simulations in general – the old chestnut “rubbish in, rubbish out” does apply.

However, where the original sources indicate that uncertainty distributions correctly describe model uncertainties as determined from experimental data, simply removing failed simulation should never be pursued. In this case, improving the code capabilities of ATHLET should be the main objective and the ATHLET development team should be contacted.

If there is no sound understanding for the reasons why a simulation failed or which specific parameters caused the problem, re-doing the whole analysis with improved distributions and a checked and improved input deck should be the default option.

A more formal check, if a pruned sample is a random sample, could be performed by looking at the values from the uniform distribution, which were used to determine the samples for the uncertain parameters. If the unordered set of samples from the (multivariate) uniform distribution can be shown to be random (possibly after rescaling it to the support of the effective parameter distribution), Wilks’ formula can be applied. This hypothesis can be tested with a suitable set of standard tests for uniformity. Given that the original samples were obtained with random number generators, the test should be performed relative to the degree of uniformity of the original sample.

If small (random) modifications of a sample of an uncertain parameter allow to replace a failed calculation, this is generally acceptable as it is equivalent to minor changes to (unrelated) input deck settings and would not invalidate the sample as random. Relevant examples for successfully curing random calculation failures would be, e.g., small (random) changes to one or a few integration settings under CW `INTEGRAT` KW `DRUFAN`. If, however, a new sample is drawn from an updated, materially different distribution, this could introduce non-random distortions. And this would invalidate the arguments above to justify the overall sample is random, and that Wilks’ formula is applicable. The same problem occurs if new samples are placed into perceived “gaps” of the existing samples or are placed into the tails of distributions.

Finally, some remarks on the use of sample simulations obtained with different code versions or input deck versions. Generally, for an uncertainty analysis to be valid, both the code version as well as the input deck (apart from varied uncertain input parameters) should be identical for all samples considered. Different code versions or input decks could introduce systematic distortions into the sample, which would invalidate the assumption that the resulting (combined) sample is still random. Again, any exceptions from this rule need further analysis and convincing arguments, e.g.:

- It can be shown that different code versions do not produce different results for the specific analysis case.
- It can be shown that minor (random) changes to initial conditions (e.g., a change in initial temperatures in a marginal TFO by less than 0.1%) or numerical integration settings (e.g., changes in H0 of 1%) cannot, by themselves, lead to non-random distortions that would materially influence the order statistics for Wilks' formula.

If it cannot be shown that any changes due to different code versions or input deck versions will effectively be random, then no samples may be added or combined from these. The uncertainty analysis probably has to be redone, if more samples need to be created. It is for this reason that only input decks under version control, e.g., via git /GIT 21/, should be used for an uncertainty analysis, and only approved code versions from a source code under version control should be used.

Overall, there will always be a risk that specific calculations in an uncertainty analysis will fail. As explained above, there are several cases where these failed simulations can be removed from the sample and a valid analysis can be done on the pruned sample. In order to make this possible and avert re-doing a whole analysis (or doing additional calculations much later), specifying more than the minimum number of samples, see Tab. 5.1 is a reasonable approach. Alternative, a higher rank order than the minimum acceptable rank order (i.e., a lower coverage) for the analysis could be specified.

5.3.1 Using restart data

AC^2/ATHLET allows the restart of a calculation from a time point in the simulation for which restart data have been saved. A restart with ATHLET should result in identical results compared to the original calculation for time steps and physical data - if done on the same computer with the same program version, input deck, and system libraries. If

this is not the case, the restart procedure needs improvements, because not all internal state variables are saved to the restart file. There are, however, a few known exceptions:

- Certain plug-ins are not capable of restarts. This specifically includes CONDRU, which will not produce a consistent restart. In addition, there are certain models in ATHLET-CD that are not fully consistent in a restart. While differences for ATHLET-CD tend to be small, some deviations will often occur.
- In user-provided plug-ins, internal state variables might exist (perhaps unintentionally). In this case, the plug-in will not produce a consistent restart, unless the user saves the internal state variables for restart time points. However, most plug-in interfaces to not provide restart functionalities.
- If ATHLET is run in a coupled calculation, the information at the coupling interface is often not available for a restart. Consequently, coupled calculations can produce non-identical restart runs, if the coupling interface includes some kind of memory effect.
- An interactive change of the ATHLET simulation model can have an influence on the simulation. Manually requiring to save an additional restart point will potentially change the numerical integration and could lead to differences, if a prior restart point is used. Any change of ATHLET state variables via external access to ATHLET global variables, e.g. facilitated via ATLASneo, would lead to different behaviour, if it occurred after the last restart point.

So, in most cases a restart with ATHLET should be an appropriate solution for a failed simulation. Even if the restart is not completely consistent, in most cases any differences should be minor and smaller than the effect of minor changes to numerical settings, which have been argued above to not invalidate a sample as still random. It is still recommended to check if a restart will produce consistent results in the specific simulation case.

5.4 Consideration of integration settings

While AC²/ATHLET is a deterministic computer code where code results are completely determined by input data as well as any restart-data[3], numerics settings can have a noticeable impact on code results. In this regard, it is important to note that the 6-equation model used in ATHLET can become numerically unstable, which is a well-known characteristic of 2-phase flow models /DIN 03/, /PAN 18/. The implementation in ATHLET ensures that such instabilities, including those from the non-hyperbolicity of the 6-equation model, generally do not lead to qualitatively wrong predictions. However, in highly transient processes such as condensation, water hammers, or sudden evaporation of highly super-heated water, ATHLET's models arrive at their limitations. Then, the error controls in the integration scheme of ATHLET might no longer be effective. Such a situation can be exacerbated by coupled multi-physics models (such as neutron physics, heat conduction and transfer, etc. routinely used in ATHLET), where in certain situations the overall error control of ATHLET might no longer be fully effective. Moreover, if GCSM models trigger actions by reaching certain threshold values (or not), this can also induce a relevant dependence on details of numerical integration. Similarly, while ATHLET-CD includes some error control mechanisms, e.g. time step size limitations, it is not included in the ATHLET error control, so that core degradation models show such dependencies in certain situations.

The consequence of this observation is that an ATHLET calculation does not necessarily converge weakly to the "real" solution. Calculations might diverge to rather different results for rather small changes to integration settings only. Integration settings will also influence the mass errors (variables `RESMASS`) produced by ATHLET throughout the calculation – and the presence or lack of 0.1 % of overall water inventory can have an influence on results as well, e.g., if a certain water level is reached or not. In addition, changes in model parameters do influence the numerical integration process. The resulting changes in time step sizes can, by themselves, accumulate to differences in predictions. Ascribing these to model settings would be incorrect.

[3] This assumes that no program errors interfere. Unassigned variables influencing program execution could cause random differences. ATHLET has been checked for such effects and identified errors have been removed. However, the EIMMB model in ATHLET 3.4.1 shows random inconsistencies in parallel calculations. Also, there will be differences depending on CPU architecture and system library versions provided by the compiler or the operating system. So, there are some qualifications to consider here.

For this reason, adding integration settings that influence time step behaviour to the uncertainty analysis should be strongly considered. This allows to identify if numerical integration effects are the main cause for differences, instead of having these as confounding variables hidden in the analysis. Input parameters under CW `INTEGRAT`, that should be considered as part of an analysis, include `EPS, ECKSCH, HMAX` as well as some of `FCLIMP, FCLIMT, FCLIMG, FCLIMA, FCLMWL, FLCMWV`, under CW `HEATCOND` the parameter `TFCONV` and under CW `NEUKINP` the parameter `DTNK`.

In addition, any time step limitations for GCSM blocks might have an effect. If the signal `SGHMAX` is used under CW `INTEGRAT`, its values (and small fluctuations thereof) will also influence the behaviour of the calculation. Using such influences for sampling purposes might be rather complicated, though, at least with regard to analysing and understanding the impact on simulation results.

5.5 Output data settings for uncertainty analysis

AC^2/ATHLET can easily produce a significant amount of output data per time point for a sufficiently large model, even if additional models, such as ATHLET-CD, or coupled codes are not present. An uncertainty analysis with, e.g., 200 simulation runs can thus easily impose a significant load on available (fast) disk space. In addition, writing substantial amounts of data to a file is a comparatively slow process that will noticeably increase runtimes if it happens with a high frequency. For those reasons, in most ATHLET calculations not every time step is saved into the plot data files.

5.5.1 Plot frequency settings

The relevant settings influencing plot frequency for ATHLET under CW `CONTROL` are the parameters `DTPLOT, ISPLOT` and `SGPLOT`. The following considerations apply:

- Unless via `ISPLOT = 1` every time step is written to the plot data (which might produce very large plot data files and slow down calculations), the plot data will be a sample of all available time steps. If there are large gaps and/or long time intervals between the output data, code results might change significantly in the interim periods. If there are highly transient processes, e.g., a sudden reactivity insertion event or brief pressure pulses, a too low plot frequency might miss such simulation results of interest, for example a peak primary side pressure.

- The user should ensure that no figure of merit for the uncertainty analysis is significantly affected by output frequency settings. This might include having to define a GCSM signal for `SGPLOT` in order to flexibly define plot time intervals for certain periods of interest.

- Alternatively, the FoM could be calculated via a GCSM model that ensures that, e.g., the maximum value or the average value of a code output (i.e., a process variable) in a certain time period is saved to an GCSM signal.

- Highly transient processes will generally result in small time steps, a typical size would be $\sim 10^{-5}$ s. By setting `ISPLOT` to a suitable value, a minimum resolution of periods with such small time-steps can be achieved. For example, setting `ISPLOT = 100` would lead to a plot data resolution of ~ 1 ms if the average time step size is around $\sim 10^{-5}$ s.

- Conversely, a low number given for `ISPLOT` might invalidate large values of the plot time interval provided via `DTPLOT` or the `SGPLOT` signal. In larger simulation models, the average time step is often ~ 10 ms. Setting `ISPLOT = 10`, a setting of `DTPLOT` = 5 s will be largely redundant[4].

5.5.2 Other output settings

As mentioned, writing data to a file slows down calculations and uses up disk space. For an uncertainty analysis, where only a limited number of FoM are actually analysed in detail, writing a lot of additional data for each simulation should be avoided. The following considerations apply.

- Before starting an uncertainty analysis, a reference (best estimate) calculation should be performed. This should be analysed in detail in order to check that the simulation produces reasonable results and behaves as expected. This phase should be used to improve the input deck and implement and verify any additional modelling needed for the determination of figures of merit. This could include providing GCSM process signals to make code results accessible independent of the nodalisation (if, e.g., the number of CVs is selected as an uncertain parameter). It would also include the

[4] Note that DTPLOT and ISPLOT or SGPLOT and ISPLOT both independently trigger an output of a plot vector. This will increase the number of plot vectors written and can result in smaller time intervals in the plot data than expected based on the CW CONTROL settings.

provision of GCSM signals aggregating code results over multiple model elements (CVs, HCVs) or over time periods.

- Writing main edits to the main output file produces quite a lot of data. Thus, setting `ISPRIN < 0` and `INPPRN < 2` is sensible.

- The output level of the models should be set under CW `OBJECTCON`. Generally, an output level of `IOUTO = 1` should be sufficient. For TFO or HCO, where additional output is needed to determine a figure of merit, this can be activated under the respective PW `TFO` and PW `HCO` as needed. Alternatively, a dedicated process signal could be added under CW `GCSM`. For TFO and HCO objects, that are neither evaluated nor would be used for in-depth analyses, setting the output level to 0 is an option.

- GCSM controller blocks can be removed from the plot data by setting `IPRI = -1`. Under CW `OBJECTCON`, PW `GCSM`, signals of interest can be manually added to the plot data again. Alternatively, signals can be individually removed from the output. For removing process signals from the output, only PW `GCSM` settings can be used.

5.6 Reliability assessment of passive safety systems

One possible approach to substantiate reliability claims for a passive safety system would be to perform an uncertainty analysis with a simulation code like ATHLET and identify a failure region. Neglecting for a moment that failure conditions of a passive system might be associated with very small or very large percentiles of parameter distributions, that these tails might not be well supported by experimental data, and that the code, e.g., ATHLET, might not be fully validated for such model parameter values, there is a purely practical problem if Wilks' formula should be used.

Passive system failure rates are often claimed to be exceptionally low. A typical figure might be 10^{-5}, corresponding to a reliability of 0.99999. As evident from Tab. 5.1, substantiating such high reliability (i.e. probability content) will require a very high number of samples. If at least one sample should be outside of the coverage at a confidence level of 0.9, then at least 388971 samples would be needed. That will not be feasible in most settings. Consequently, application of Wilk's formula for assessments of very high reliability claims will often not be feasible and more targeted approaches will be needed to determine (distribution-free) confidence intervals.

5.7 Additional observations

Physical systems simulated with AC²/ATHLET will not exhibit discontinuities, even if some of the phenomena available in ATHLET can occur on very small time scales. As ATHLET uses a discrete time step and as the plot data are usually a (sparse) sample of all time steps, this might no longer be obvious in the data available for an uncertainty analysis.

Looking at the manifold of (physical variable) code results $A^{(M_p,n)}([0,1]^n \times [t_l, t_h])$ over the whole simulation for the set of uncertain parameter values, this set should be a closed connected set, provided the set of initial results $A^{(M_p,n)}([t_l] \times [0,1]^n)$ was connected, which is often a reasonable assumption. The same applies to the set of FoM $A^{(m,n)}([t_l, t_h] \times [0,1]^n)$, provided these behave as physical variables. However, as the non-linear differential equations in the ATHLET models can (and sometimes will) lead to bifurcations and other qualitative changes in results, the set of FoM for a specific time point $A^{(m,n)}([0,1]^n \times [t_0])$ with t_0 sufficiently distant from the starting time t_l may very well be non-contiguous.

The presence of discontinuous models in ATHLET like, e.g., GCSM, will introduce further discontinuities to the code response. Moreover, even where non-hyperbolicity of ATHLET thermohydraulic models is not an issue, the numerical integration of multi-physics models or code coupling with not fully effective numerical error control can introduce chaotic behaviour of code responses to small parameter variations. This is, e.g., a known feature of core degradation models in ATHLET-CD during highly transient phases of cladding oxidation and fuel relocation, where highly non-linear models govern the response of the simulation. That means that for all code results in a small ϵ-ball around $A^{(m,n)}(t_0)x_i^{(n)}$ with a small constant ϵ, which is larger than the minimum numerical resolution ϵ_r, there is not always a reasonably small constant $K \gg 0$ so that all parameter values $x_j^{(n)}$ are in a (small) $K\epsilon$-ball around $x_i^{(n)}$, i.e.

$$A^{(m,n)}(t_0)x_j^{(n)} \in B_\epsilon\left(A^{(m,n)}(t_0)x_i^{(n)}\right) \nRightarrow x_j^{(n)} \in B_{K\epsilon}\left(x_i^{(n)}\right) \; \forall \; x_j^{(n)} \tag{5.33}$$

In addition, two different sets of uncertain parameters $x_i^{(n)}$ and $x_j^{(n)}$, which are not even close to each other, might lead to the same code response at (and thus potentially after)

a time t_0, i.e. $A^{(M,n)}([t_0,t_h])x_i^{(n)} = A^{(M,n)}([t_0,t_h])x_j^{(n)}$ with $\left|x_i^{(n)} - x_j^{(n)}\right| > \delta \gg 0$. And with regard to the FoM of the uncertainty analysis, the subset of these FoM can easily have the same values for different uncertain parameter values at a time t_0 and then a different value at a later time t_1:

$$A^{(m,n)}(t_0)x_i^{(n)} = A^{(m,n)}(t_0)x_j^{(n)} \wedge A^{(m,n)}(t_1)x_i^{(n)} \neq A^{(m,n)}(t_1)x_j^{(n)} \mid t_1 > t_0 \tag{5.34}$$

In other words, for any small ball $B_\epsilon(f_0^{(m)}(t_0)) \subset (V_C^m)$ around a sample $f_0^{(m)}(t_0)$ at a specific time point with ϵ sufficiently larger than the minimum numerical resolution ϵ_r, the retraction $\left(A^{(m,n)}(t)x^{(n)}\right)^{\leftarrow}\left(B_\epsilon(f_0^{(m)})\right) \subset [0,1]^n \times \{t_0\}$ might consist of several mutually disjoint subsets, each of which are connected, but might not be simply connected. And, moreover, not all subsets have to have the same dimension. Conversely, for any small ball $B_\epsilon(u_0,t_0) \subset [0,1]^n \times \{t_0\}$, the image $\left(A^{(m,n)}(t_0)x^{(n)}\right)(B_\epsilon(u_0,t_0))$ might consist of several mutually disjoint connected sets, which might not be simply connected and have different dimensions[5].

Mathematically, this means that the function $A^{(m,n)}(t)$ does not need to induce a continuous mapping from the space of uncertain parameters to the space of FoM for a specific point in time or a time interval. It may not be continuous mapping globally or even locally, at least for certain subsets of the parameter space. In addition, the function $A^{(m,n)}(t)$ does not need to be bijective, and thus it may not be a homeomorphism /FIS 78/, /JÄN 79/ either globally or even locally, if $n = m$.

While these observations are not a problem for the application of Wilks' formula on code results, additional care is needed when interpreting the confidence intervals and contours obtained in this way and in applying and interpreting any sensitivity indices to the sample. One advantage of the order statistics approach is that sensitivity indices applied to the ordered ranks are generally valid. However, if sensitivity indices are applied to the

[5] Importantly, the system function A might show deterministically chaotic behaviour and we might end up with non-integer dimensions for these sets and/or subsets, if we would allow arbitrary numerical precision. Such a result can very well be a property of the underlying multi-physics two-phase flow system, but it might also be an artefact of numerical integration.

sample of code results, it needs to be checked if relevant premises for the application and interpretation of these indices are fulfilled. This will require insights about the (at least local) behaviour of $A^{(m,n)}(t)$, if it is a continuous function, a monomorphism or a diffeomorphism (which would imply $n = m$) /FOR 77/.

Another topic is making inferences about the underlying uncertain parameter distributions based on the confidence intervals obtained in the uncertainty analysis. Generally, such inferences will not be possible. If, however, it can be demonstrated that $A^{(m,n)}(t)$ (and thus $A^{(m,n)}(t)x^{(n)}$) is a continuous and invertible function (locally on a subset, i.e. generally $n = m$ for those subsets) and if the Jacobian is positive definite, then $\left(A^{(n,n)}(t)\right)^{-1}$ exists /FOR 77/ and preserves the orientation of a set of code values. If these conditions would apply globally (i.e. for the relevant time window and for the whole hyperrectangle of the confidence contour), it would be possible to transform the contour back into the space for the uncertain parameters. This, however, would still not ensure that the set $\left(A^{(n,n)}(t)\right)^{-1}(V_C^m)$ is free of gaps (i.e., simply connected) on a time interval of interest $[t_0, t_1]$, it only has to be a closed connected set. A way of showing that $\left(A^{(n,n)}([t_0, t_1])\right)^{-1}(V_C^m)$ is a simply connected set, would be to show that $A^{(n,n)}(t)$ is also a convex function, at least in the interval $[t_0, t_1]$. In most cases, most of the conditions discussed above are not met, e.g., because $n \neq m$. For the more generic case, we would need to show, e.g., that $U_C^n = \left(A^{(m,n)}(t)x^{(n)}\right)^{\leftarrow}(V_C^m)$ is a simply connected set. And if we want to transfer the boundaries directly, we would have to show that $\partial U_C^n = \partial\left(A^{(m,n)}(t)x^{(n)}\right)^{\leftarrow}(V_C^m)$. Reliably working approaches for actual analyses with, e.g., ATHLET to make such demonstrations should be investigated.

5.8 Update of ATHLET documentation

The ATHLET validation manual has been extended with additional guidance on the application of Wilks' formula, the determination of sample sizes for uncertainty analysis and the treatment of code crashes, first for ATHLET 3.3 /HOL 22/ and then extended again for ATHLET 3.4 /HOL 23/. The additions to the manual cover the issues discussed in more detail in the sections above and summarize these for ATHLET users. In addition, a table for reading off minimum sample sizes as well as tables for estimating upper and lower limiting samples for different percentiles have been provided.

6 International Activities with reference to Uncertainty Analyses and Cross-Cutting Activities

To follow the current state of knowledge about uncertainty analyses and quantification and to present the results of our work, participation in conferences and international projects or activities was an important activity in the project.

The participation in the following conferences had a significant benefit for the methodology development and application within work package 1, which is described in chapter 1:

Courses in 2021 about Data Analysis and Bayesian Methods

- Data Science and beyond: Data Assimilation with elements of machine learning (Summer school Utrecht) – 2ECTS
- Data Assimilation – intensive joint course of DARC/NCEO/ECMWF
- Bayesian Statistics 2ECTS with exam (The 4th IPSA – HSE Summer School for Methods of Political & Social Research)
    - This course aimed to offer a brief, informal introduction to the theory and application of Bayesian statistical methods. It began by covering the fundamentals of Bayesian statistics, such as conditional probability, Bayes' theorem, and prior and posterior distributions. The course explored various methods for estimating and evaluating Bayesian models, with a focus on MCMC techniques in the context of generalized linear models. Key Bayesian approaches to model selection, including Bayes factors, DIC, and cross-validation methods were also discussed. Furthermore, an introduction to Bayesian model averaging (BMA), a powerful tool for addressing model specification uncertainty, has been addressed.

CNS-DIET

The DIET conference provides a forum for nuclear industry professionals, academics, and regulators to exchange views, ideas and information relating to the innovative technologies that are already having an impact, as well as those just over the horizon.

2022 - CNS-DIET Online Conference 2022: During the conference, papers were presented for different aspects ("Towards Net Zero Future", "Disruptive Technologies"., etc) with Machine Learning and Artificial Intelligence.

2023 - CNS-DIET Online Conference on “State of the art in Machine Learning and Artificial Intelligence”: the plenary and technical sessions of the conference covered a wide range of cutting-edge topics, including data analysis and innovative methodologies, artificial intelligence and machine learning, regulatory frameworks, big data, digital twins, virtual reality, AI and robotics, nanotechnology, fusion, and adaptive manufacturing. These sessions highlighted advancements and emerging trends across various fields.

BEPU – Best Estimate Plus Uncertainty Conference 2024

The Conference was organized in two main tracks: one devoted to Industrial Applications and Licensing (Track-1), and the other to Research and Development (Track-2). GRS actively participated with a presentation (/TIB 24/), and in the FONESYS and SILENCE panel discussion: “Status and perspectives of codes and experimentation in nuclear thermal hydraulics following FONESYS and SILENCE networks” (/NIN 24/).

Additionally, a detailed literature study was performed for the development of improved methodology for model uncertainties quantification. The main findings were:

- Bayesian methods are regarded as more suitable than frequentist method for inverted uncertainty quantification on the basis of comparison with measured data
- Quality of the experimental data base used for quantification and definition of the error metric for comparison of simulation and measured data is usually more important than the features of the applied probabilistic method
- A crucial problem of model uncertainties quantification is its application to the large scales. The upscaling of the quantified model uncertainties is still an open issue

Concerning the international cooperation in the field of inverted uncertainty quantification of system codes thermal-hydraulic models, GRS participates in the OECD/NEA project ATRIUM which was endorsed at the end of 2021 and started in 2022, and FONESYS, which is a voluntary network of thermal-hydraulic code developers.

In the following two sub-chapters the GRS activities within these two projects are summarized.

6.1 ATRIUM

The OECD/NEA project ATRIUM (Application Tests for Realisation of Inverse Uncertainty quantification and validation Methodologies in thermal-hydraulics) is a continuation of the already finished OECD/NEA projects BEMUSE (/NEA 11/), PREMIUM (/NEA 17/) and SAPIUM (/NEA 18/). The main scope is to demonstrate the applicability of the methodology developed in SAPIUM with the following objectives /NEA 23/

- Verify the applicability of the best-practices;
- Identify possible new issues;
- Summarise the lessons learnt from the different participants and possibly update the recommendations based on the results of the activity.

To achieve these objectives two phases of a benchmark exercise are foreseen /NEA 23/:

- In the first exercise, the experimental database is composed of Separate Effect Tests (SETs) which allows isolating the physical phenomenon of interest, simplifying the IUQ process. The selected phenomenon is the critical flow at the break, which is of significant relevance during a LOCA.
- In the second exercise, several influential phenomena are involved at the same time in the experiments, which may then be called Combined Effect Tests (CETs). In this category, the post-CHF heat transfer phenomena will be studied.

6.1.1 Participation in the ATRIUM IUQ benchmark Phase 1

In the following section, the GRS contribution to the first phase of the benchmark exercise is presented. The final evaluation and comparison to other codes applications is not finalised at the end of the present project and needs to be reported by the task coordinator.

6.1.1.1 Selection and assessment of simulation model: element 3 of SAPIUM

6.1.1.1.1 Selection of simulation model

The critical flow model in ATHLET is based on two sets of balance equations:

- Set of 4 balance equations for liquid dominant system – low void fraction, saturated vapour, subcooled, saturated or superheated liquid
- Set of 4 balance equations for vapour dominant system – large void fraction, saturated liquid, superheated, saturated or subcooled vapour

The sets of balance equations are closed by an evaporation model, a phase slip model simulating different acceleration of the phases, and a wall friction model. The constitutive models are mechanistic and have been proved for a wide range of parameters. The empirical parameter in the evaporation model is the number of nucleation sites. It is a parameter depending on the simulated fluid (and based on theory also on boundary conditions like, e.g., pressure). Its value was kept constant for all simulations with water. The critical discharge model of ATHLET was tested also for cryogenic hydrogen, where only the number of nucleation sites was modified. All other models remained the same as for water simulations. It is a strong indication for the full range performance of the model.

6.1.1.1.2 Spatial discretization

For the determination of critical discharge flow rates, a set of steady-state balance equation is solved. The model equations are integrated by the numerical solver spatially along the nozzle. The nozzle geometry is defined as nozzle cross-section in the function of nozzle length coordinate. The shape of the nozzle is approximated in ATHLET by the part of cosines curve or constant diameter pipe. It means the modelled shape of the inlet part of the nozzle can be different from the real one. It is negligible for the long nozzles with throat of constant diameter. For short nozzles such an approximation contributes to uncertainty of the simulation (the cross-section areas given in the excel file for fidelity of the converging part of the nozzle 2 seem to be in disagreement with the shape of the nozzle as defined in the specification). The expansion after the nozzle throat and the diverging parts of the nozzle are not simulated with ATHLET critical discharge flow model.

The model equations are differential equations. They are integrated spatially by the ODE solver DIFSYS using the finite difference method with local rational extrapolation. The solver uses variable integration step automatically adjusted to the solution variables development. The integration steps can vary strongly depending on geometry but also depending on thermal-hydraulic parameters. The observed number of integration steps varied from 20 up to over 100.

6.1.1.1.3 Sensitivity analyses

Sensitivity analysis concerning wall roughness has been performed. The wall friction model in ATHLET uses the Darcy-Weisbach friction factor λ. In the analyses the factor was assumed to be 0.01. For highly turbulent flows this value is equivalent to a roughness of about 10^{-5} m. Sensitivity analyses were performed using λ = 0.02. It is a larger value equivalent to about 10^{-4} m wall roughness. Such a large value has been taken, because for the Sozzi & Sutherland experiment generally too high critical flow rates were calculated. Using a larger λ value resulted in smaller predicted critical flows. The variation according to the different λ was below 1% for short nozzle up to about 10% for long nozzles. The smaller critical flow rates mean better agreement with the experiment but λ = 0.02 is a very large value for small discharge nozzles. Finally, for the quantification of model uncertainty the λ = 0.01 has been taken. It was recognized as more probable for discharge nozzles. Moreover, taking λ = 0.01 was regarded as consideration of experimental uncertainty leading to the increase of quantified uncertainty range of the model.

6.1.1.1.4 Uncertain input parameter

Uncertainty of the critical discharge model is simulated using a multiplication factor for critical flow rates. Selection of the global parameter for representation of the model uncertainty was motivated by:

- The measurement of critical flow rates is accurate compared to the accuracy of measurement of parameters related to constituent models, like evaporation rates or relative velocity/void fraction
- Application of a global parameter in this case is simpler and more accurate. The cumulate measurement error of parameters related to specific models would be really large, whereas the accuracy of the critical discharge rates is quite high.

- The overall accuracy of the critical discharge model has been proved as good, so there was no necessity to search for a model contributing to the overall simulation error, as the evaluation of errors of particular models would be worse than the global error.
- It is known that in the determination of critical discharge rates, the flashing model is the most sensitive but just experimental estimation of the evaporation rate is particularly inaccurate.

6.1.1.2 Model input uncertainty quantification

6.1.1.2.1 Aggregation of information from experiments

For the quantification of model uncertainties, the experiments of Sozzi & Sutherland and Moby Dick were selected by development and assessment of the experimental data base. The number of tests in the Sozzi & Sutherland experiment was much larger than in the Moby Dick experiment. Since analyses of adequacy of the experimental data base showed that the accuracy of the Moby Dick experiments was better, it was decided to reduce the number of the considered Sozzi & Sutherland tests. Instead of introducing different weights for the tests, the same number of tests (12 runs) for each nozzle used in the Sozzi & Sutherland experiment was considered. The tests were selected to get a representative sample covering the experimental investigated field.

6.1.1.2.2 Uncertainty quantification

In case of the Sozzi & Sutherland tests, the test series with throat length 7.5”, while in case of the Super Moby Dick the test series with expansion, were not used for the quantification, since they were chosen for the validation phase (see below).

The uncertainty quantification was based on the G_exp/G_cal value determined for each test. All data points of the Sozzi & Sutherland tests (12 for each nozzle) were taken into account with a weight of one. In case of the Super Moby Dick (27 for SMD divergent configuration) two variants have been considered with weights of one or two. The latter (w = 2) was investigated to express the higher confidence in the SMD results due to its well defined one-phase inlet conditions.

Based on the data points, a fitting procedure has been carried out to find the best fitting distribution. In this process, the search was not only based on statistical analysis, but also on engineering considerations taking physical feasibility into account, thus limiting the possible distributions.

The fitting has been carried out with the help of Python. The quality of the fit has been assessed using the Residual Sum of Squares (RSS) and Kolmogorov-Smirnov (KS) tests. Bootstrapping has been utilised to judge the stability of the fitted distribution. The decision then was made based on the combination of these tests, choosing the distribution performing well on all three.

As a result, the following distribution has been deduced:

- W_{SMD} = 1 Gamma distribution with shape=2.317, scale=0.0928, location=0.7389 parameters (Fig. 6.1)

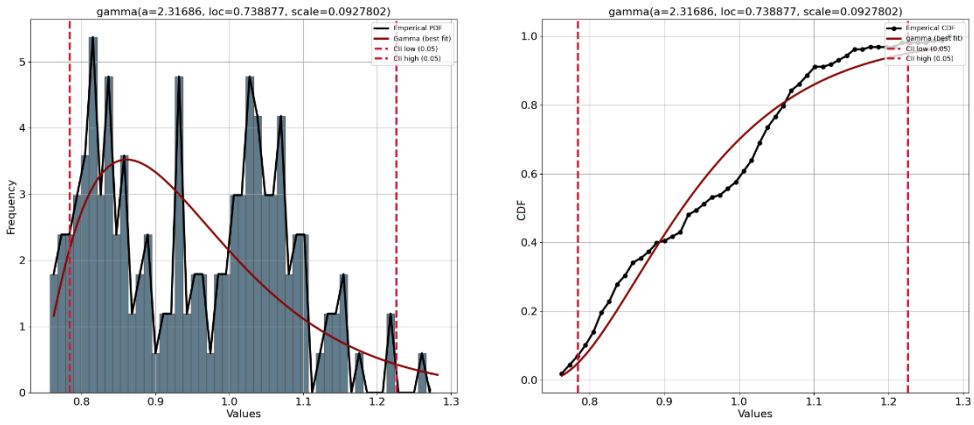

Fig. 6.1 Frequency and CDF as function of W_{SMD} = 1 Gamma distribution with shape=2.317, scale=0.0928, location=0.7389

- W_{SMD} = 2 Gamma distribution with shape=1857.41, scale=0.002814, location=-4.258 parameters (Fig. 6.2)

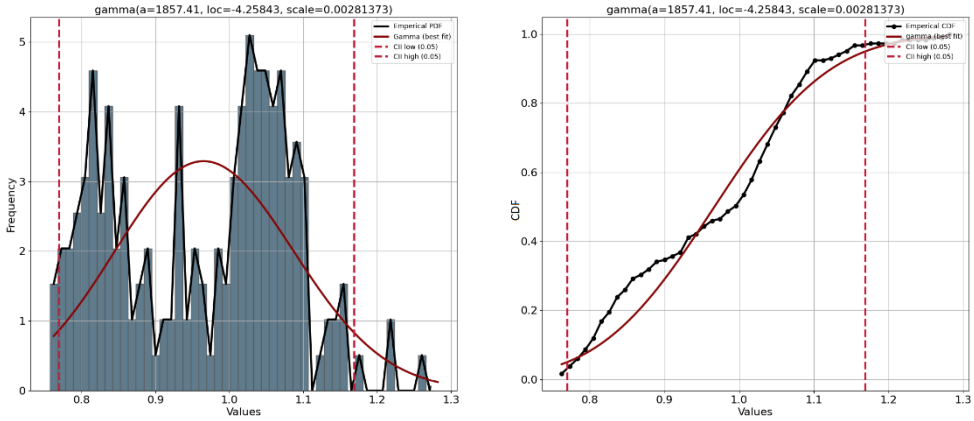

Fig. 6.2 Frequency and CDF as function of W_{SMD} = 2 Gamma distribution with shape=1857.41, scale=0.002814, location=-4.258

In both cases the cut-off for the distribution support was determined to be [0.75, 1.3].

6.1.1.3 Validation

For the validation of quantified model uncertainty, the Moby Dick tests with expansion and Sozzi & Sutherland tests with throat length 7.5” were used. Since the determination of critical discharge rates is based on the steady-state model it is sufficient to perform the validation also using steady-state experiments. Finally, the Marviken tests considered by development of the experimental data base were not used. Validation of the quantified uncertainty on transient application could be of interest but concerning the additional uncertainties of the Marviken test simulation the results of such validation would be less accurate than results of validation based only on the steady-state experiments.

In Fig. 6.3 the data points of the validation set are shown alongside with the fitted distributions for both w_{SMD}=1/2. In both cases the experimental points lie within the possible range determined by the fitted distributions and fall in the range of more probable values.

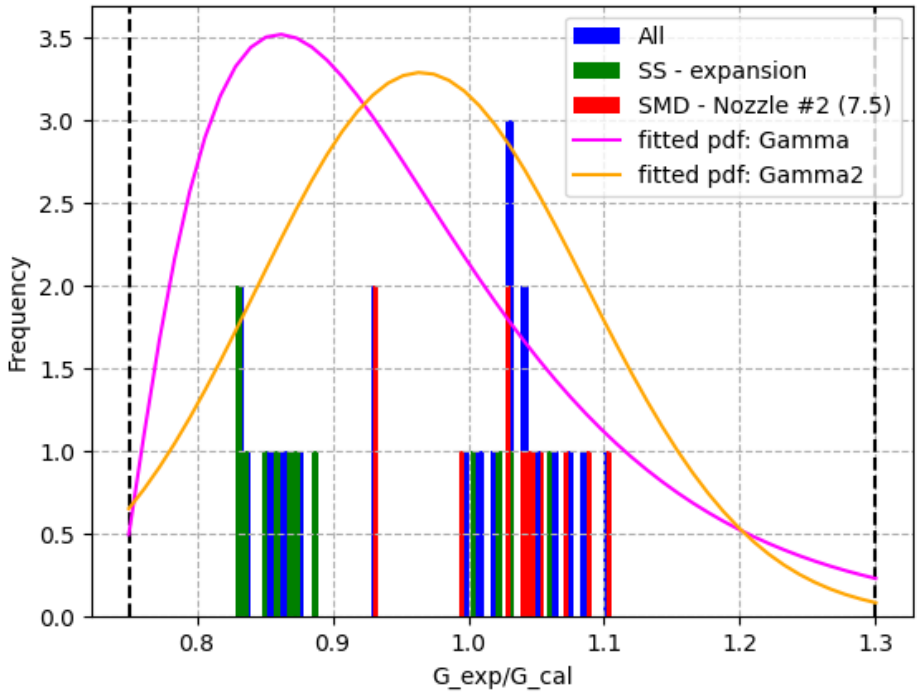

Fig. 6.3 Validation of derived model uncertainty

A joint publication focusing on the adequacy analysis (representiveness and completeness) of the experimental data base intended to be used for IUQ has been published /BAC 24/. Another joint publication focusing on the first exercise in the ATRIUM project (the quantification of the input model uncertainties related to the critical flow) is under preparation.

6.2 FONESYS Benchmark

In the frame of this project, considering the GRS experience in this field, presentation on inverse uncertainty quantification was invited and presented during a FONESYS meeting in 2021. GRS participates also in the benchmark on IUQ defined within the FONESYS network. This benchmark is related to uncertainties quantification of thermal-hydraulic models of system codes. During the first part of the benchmark, separate phenomena are to be identified within the combined effect tests and on this basis model uncertainties are to be quantified analogous to the separate effect tests. The measurements proposed by the FONESYS group for quantification were analysed by GRS and results were presented excluded from the quantification data base, as not sufficiently qualified for the models to be used in inverse uncertainty quantification.

Additionally, a benchmark was initiated about the reverse natural circulation in the PKL facility based on the experiment PKL III F1.2 to evaluate the impact of pressure drops on the reverse natural circulation. Some selected preliminary results of the benchmark which is coordinated by University of Pisa are presented in the following. The final comparison report is under development (/CHA 24/).

6.2.1 Inverse Uncertainty Quantification

An activity based on the THTF and RBHT test facilities was proposed and carried out. During these activities, GRS actively participated in discussions, contributed to advancing the community's knowledge, and remained up to date with international advancements concerning IUQ at partner institutes. Additionally, GRS presented its approach for fitting a global uncertainty factor for critical mass flow models in ATHLET. However, due to resource constraints, a full IUQ study based on the THTF and RBHT test facilities could not be conducted.

6.2.2 PKL reverse natural circulation benchmark

6.2.2.1 Objectives of PKL RNC Benchmark

Flow reversal (FR) conditions, which are related to direct flow under nominal operating conditions, can occur in several fault scenarios that are part of the design basis conditions (DBC) envelope. Such conditions can be observed in the entire primary circuit or in selected parts of the primary circuit. The reverse natural circulation (RNC) scenario implies pressure losses at low flow velocities (low Re), which are sensitive and subject to large prediction errors. In the RNC benchmark, a postulated modification was made to the PKL configuration, shifting the power from the core to the downcomer. As RNC was not investigated experimentally in the PKL test facility, the user effect of the choice of form loss coefficient would be included in the results. /CHA 24/

The objective of the RNC benchmark is to examine the impact of deficiencies in pressure loss modelling in a scenario of practical relevance with qualified nodalization for direct NC (PKL test F1.2) and to elucidate the influence of local pressure losses on the prediction of complex accident scenarios. The anticipated outcome of this endeavor was a repository of calculation results furnished by the participants. /CHA 24/

6.2.2.2 Results of PKL RNC Benchmark

The virtual experiment started with a single-phase reverse natural circulation and transitioned to a two-phase reverse natural circulation as the mass inventory was gradually reduced. The set-up was characterized by the occurrence of different natural circulation conditions. In general, the main phases are described by the following regimes:

- Single phase reverse natural circulation
- Two-phase reverse natural circulation

The initial conditions of the facility are characterized by stable natural circulation in subcooled conditions. The partial closure of the steam valve of the SG results in an increase in pressure in the secondary side and temperature in the primary side. The mass flow rates are approximately 1.25 kg/s in each loop, as calculated by ATHLET (Fig. 6.4).

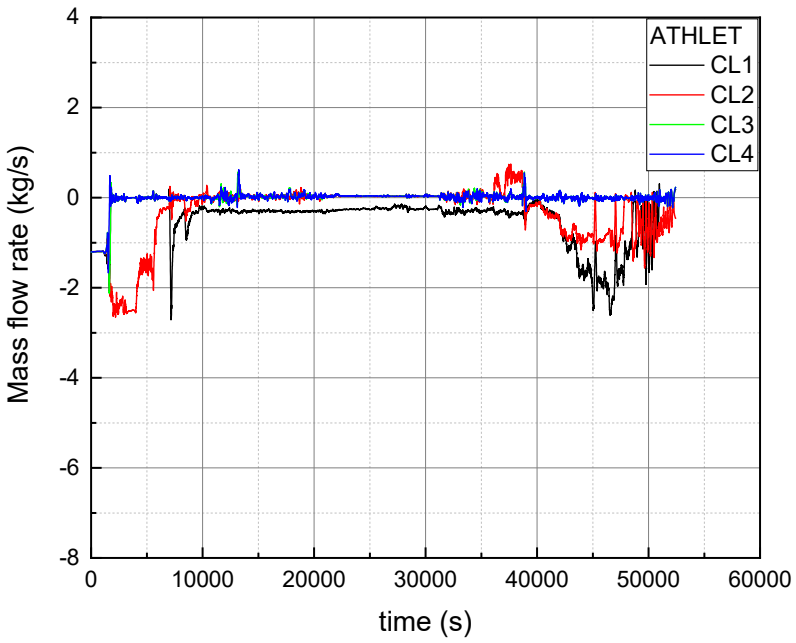

Fig. 6.4 CL mass flow by ATHLET (/CHA 24/)

During two-phase reverse natural circulation, asymmetrical behaviour in the four loop is predicted in the ATHLET simulation. The mass flow in loop 2 increased up to 2.5 kg/s by the end of single-phase RNC, while the mass flow in other three loops decreased to 0 kg/s. At 7000 s, the two-phase RNC stopped in loop 2 and started in loop 1. Loop 3

and 4 have no mass flow during the two-phase RNC (Fig. 6.4). Fig. 6.5 shows the void fraction in the different CL as function of time. After 7000 s mainly steam is predicted in most loops.

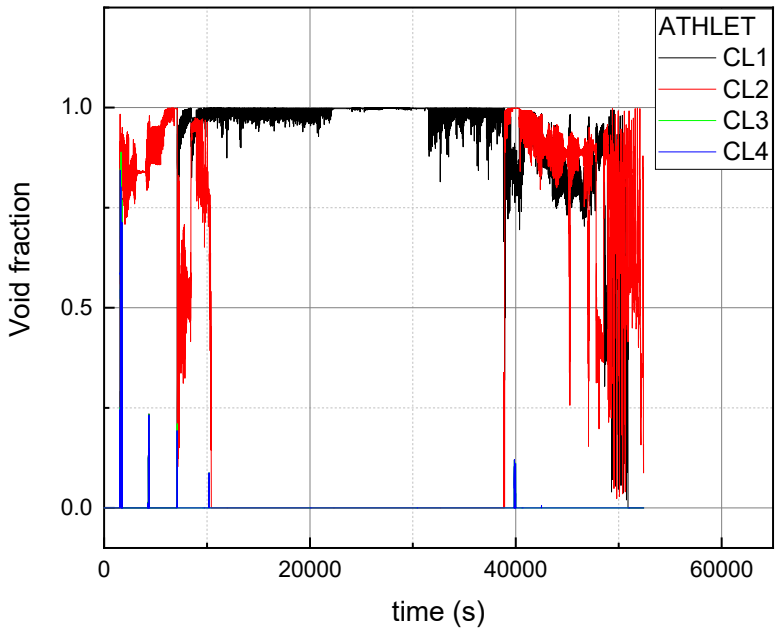

Fig. 6.5 CL void fraction by ATHLET (/CHA 24/)

The main objective of this exercise was the prediction of pressure losses as function of mass flow. Afterwards, a short summary of the ATHLET findings is given.

In single-phase flow, the results calculated by ATHLET (Fig. 6.6, Fig. 6.7, Fig. 6.8 and Fig. 6.9) showed a linear relationship between pressure drop and mass flow rate. There is no obvious fitting relationship between pressure drop and mass flow in two-phase flow.

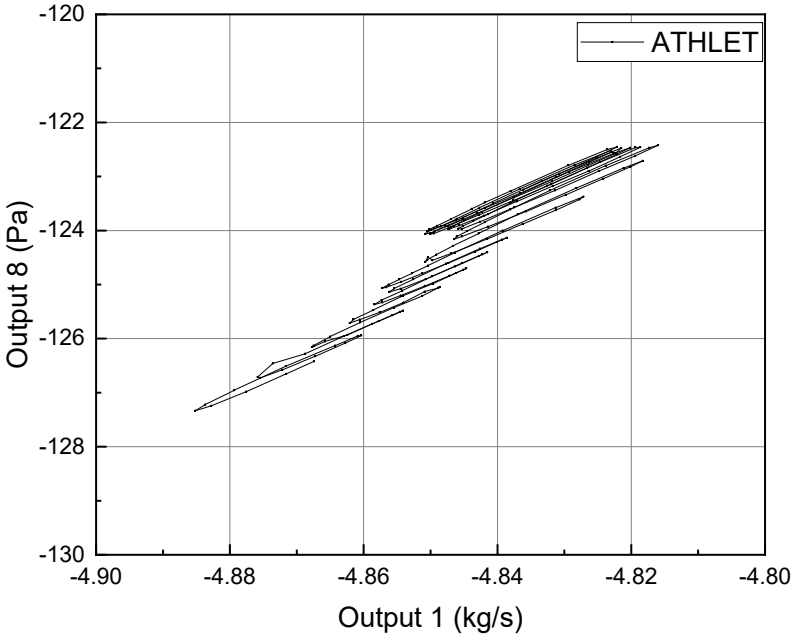

Fig. 6.6 Single-phase RNC, DP1 as a function of core mass flow by ATHLET (/CHA 24/)

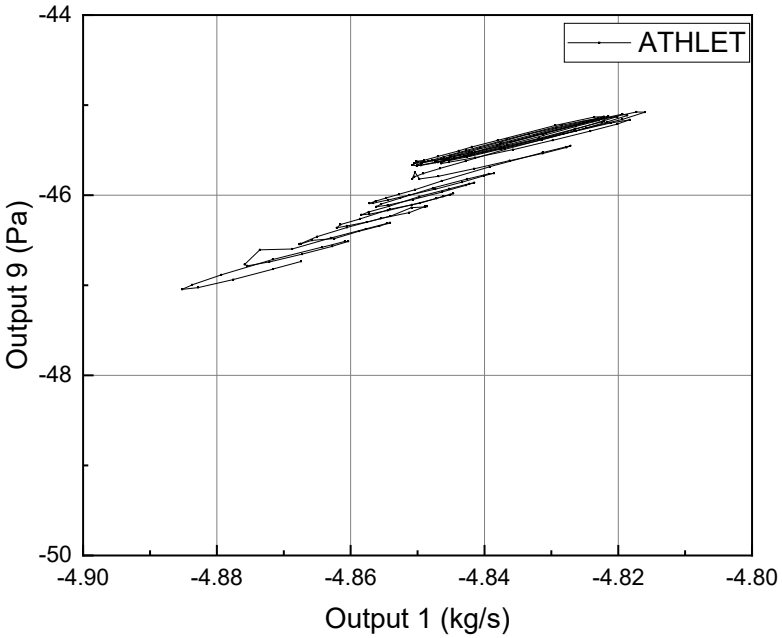

Fig. 6.7 Single-phase RNC, DP2 as a function of core mass flow by ATHLET (/CHA 24/)

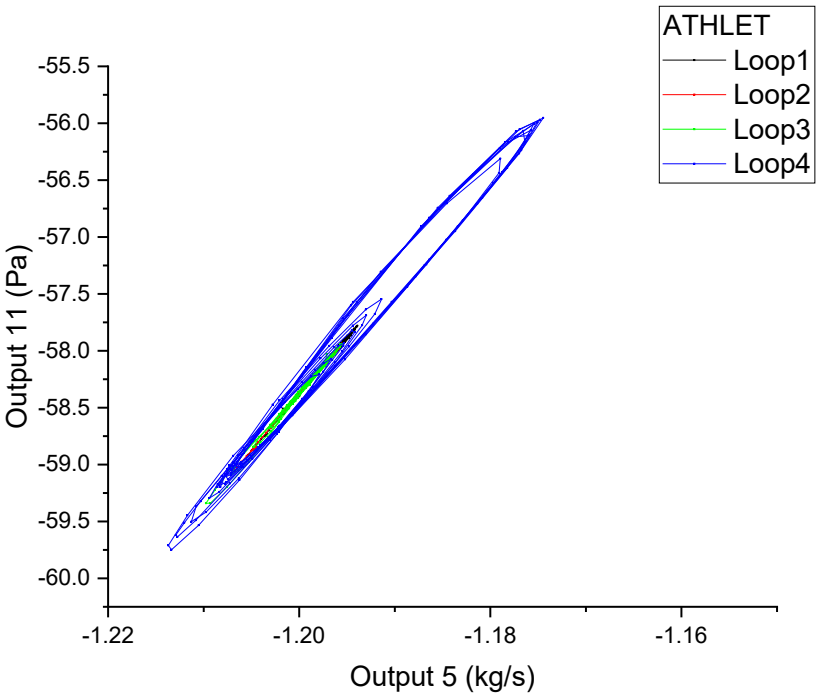

Fig. 6.8 Single-phase RNC, DP4 as a function of mass flow by ATHLET (/CHA 24/)

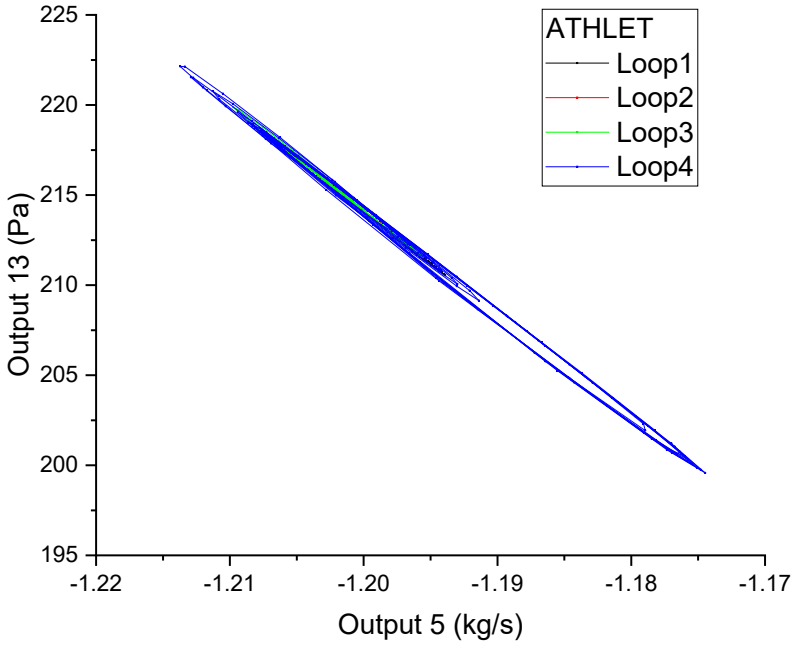

Fig. 6.9 Single-phase RNC, DP6 as a function of mass flow by ATHLET (/CHA 24/)

6.2.2.3 Benchmark Conclusions

The virtual RNC benchmark considered four simulations by four different codes. The system codes include RELAP5/MOD3.3, ATHLET, CATHARE and LOCUST 1.2. All calculations presented steady-state single-phase reverse natural circulation and two-phase reverse natural circulation for a period exceeding 20,000 s. The calculations conducted using RELAP5, CATHARE and LOCUST1.2 were terminated prior to the replenishment stages of the primary mass inventory. Asymmetrical loop behaviour was observed in all calculations during two-phase flow. (/CHA 24/)

In single-phase flow, a linear relationship is observed between pressure drop and mass flow rate, as calculated by ATHLET and CATHARE. In contrast, the pressure drop calculated by LOCUST1.2 exhibits a step distribution as a function of mass flow rate. No discernible correlation exists between pressure drop and mass flow in two-phase flow. (/CHA 24/)

7 Conclusion and Outlook

The objective of the project RS1597 was the development of a methodology for the inverse uncertainty quantification for AC²/ATHLET. Besides the methodology development, suitable Separate Effect Tests (SET) and Combined Effect Tests (CET) for this task were analysed. The project included also cross-cutting issues like updating of the ATHLET documentation for the GRS method for uncertainty analysis as well as participation in international activities with respect to (inverse) uncertainty quantification.

In the framework of the project several aspects of carrying out an inverse uncertainty analysis have been addressed. As the question of parameter choice of interest (parameters related to the reflooding phenomena) has been set at the beginning of the project, the first step of the SAPIUM guidelines have been already fulfilled. As this phenomenon served as the basis for several international project previously, the set of experiments has already been tried and been deemed adequate and appropriate for the quantification process. This was established during the preparation phase in order to be able to focus our resources on the practical statistical step of the inverse uncertainty quantification: derivation of the probability distribution function (being in a closed or numerical/empirical form) for multiple uncertain input parameter based on integral and combined effect tests.

For this purpose, several data analysis methods within the Bayesian framework have been investigated. Due to the difficulty in determining a likelihood function: either being mathematically challenging/not possible or in order to be able to derive it having to introduce several constraints on the distribution of results (mathematically as well as relying on physical/modelling information that requires expert judgement) it has been decided to move to a so-called likelihood free method. These methods still use the likelihood function for the inference, but instead of determining it in some way (analytically or numerically) it is merely approximated.

The approach chosen and implemented for our purposes is the so-called ABC (Approximate Bayesian Computation) method. This method circumvents the derivation of the likelihood function by introducing a rejection scheme based on an appropriately chosen distance metric comparing the simulated values with the observational data, or their properties (mean, skewness, etc.). As our goal was to derive a method that can easily be applied and adopted to other problems, we aimed to eliminate expert judgement as much as possible. This not only led to the application of the ABC method but also to the decision of not using meta-models, but the actual code (ATHLET) itself. This does not

mean that in the future the derived method could not be applied to meta-models, which would save computational time, but ensured that we did not make assumptions in our approach that are only valid for the given problem and can be difficult or impossible to transfer to other problems.

Several implementation approaches have been looked into, and in the end, Python has been selected due to its flexibility, adaptability, extensive available libraries, and for its available easy connection to other GRS tools (MCDET, SUSA) addressing aspects of (forward) uncertainty quantification. The method had been implemented and tested on small examples before moving to adapt it to ATHLET simulations. Here a stepwise approach has been followed extending the observational matrix and the corresponding simulation gradually in order to be able to address any arising problem in due time. Unfortunately, due to the limited reliability of the HPC resources used, the actual simulation phase has taken longer than expected, limiting the progress we could make in this project.

The last steps of the approach (verification and validation) would involve a forward uncertainty analysis applying the derived distributions of the uncertain input parameters. As a first check a simple enveloping check has been carried out. This can be carried out both visually and mathematically. For the latter, a suitable multidimensional distance function needs to be defined and its results need to be checked against an appropriately chosen limit for acceptance.

For the validation of AC²/ATHLET using SET and CET, in the frame of the current project reflooding tests of the test series FEBA, FLECHT and PERICLES were investigated. The results of one of these tests were also used for the inverse uncertainty quantification during the methodology development process. The results of these test series show that the qualitative behaviour can be predicted by ATHLET for all three test cases, but there are some systematic deviations over time during the reflooding phase for some initial and boundary conditions.

The ATHLET validation manual has been extended with additional guidance on the application of Wilks' formula, the determination of sample sizes for uncertainty analysis and the treatment of code crashes, first for ATHLET 3.3 and then extended again for ATHLET 3.4. The additions to the manual cover the issues discussed in more detail in this report and summarize these for ATHLET users. In addition, a table for reading off

minimum sample sizes as well as tables for estimating upper and lower limiting samples for different percentiles have been provided.

In the frame of international activities GRS participates mainly in two projects resp. networks OECD/NEA ATRIUM and FONESYS. Both projects deal – at least partially for FONESYS – with inverse uncertainty quantification. In OECD/NEA ATRIUM a benchmark for inverse uncertainty quantification is the main objective. GRS participated in the first phase, which is still ongoing at the end of this project, and the final report is in preparation. In the frame of FONESYS GRS participated in a virtual reverse natural circulation benchmark on the basis of the PKL F1.2 experiment. One main objective was the prediction of pressure losses for such a scenario. The results show that no discernible correlation exists between pressure drop and mass flow in two-phase flow.

7.1 Outlook

Based on the current status of modelling for the inverse uncertainty quantification used for AC²/ATHLET applications, some topics for improvement could be identified:

Investigating different ABC metrics

The choice of summary statistics as well as distance metric is crucial. Investigating different options could deliver useful information about the applicability of the method and could help develop guidelines concerning the application of the developed approach. Another aspect that should be considered is the relaxing of the acceptance criteria (ϵ) in order to manage the CPU requirements of the approach.

Expanding the experimental data base

Currently, one experiment is considered (FEBA216) with only one time series. This could and should be expanded including not only to several time series of this test, but also to several tests. Furthermore, including more than one experiment (PERICLES; FLECHT; ACHILLES etc.) would be beneficial to counter systematic bias in the FEBA test data. Here another point of the hierarchical approach can be investigated: including all data at once, or deriving a chain of posteriors each improving the previous one as prior while adding information sequentially to the process. The latter might also need an outside iteration loop, but there still might be effectiveness gains.

Saving chain-information mid analysis

At the moment no intermediate storage of chain information takes place, which requires no disturbances of the calculational chain. implementing such storage could address the issues of the fragility of the chains to disturbances, e.g., coming from the HPC resource by making an already started chain restartable without having to repeat already performed calculations. This however has to be implemented outside the present framework and also would require the implementation of the ABC method with only basic Python packages.

Including modelling bias

Currently, the method only takes into consideration the experimental uncertainty and not the possibility of a model bias. This would have been the next step and it would be important to investigate its effects and possible inclusion in the quantification process. At the same time this would further complicate the identifiability problem. We only have simulation results and experimental values. Quantifying several u.i.p. at the same time generally can have several (local) optimums. Which one is the overall optimum and how the experienced uncertainty in results can be allocated to the specific u.i.p are difficult questions. It is possible to step outside of a set of specific u.i.p.s and apply only parts of this set, possibly combined with other u.i.p.s.

Another aspect could be considering nested loops: first fixing either bias or experimental uncertainty, then moving to the next one.

Better resource allocation

Both on the HPC server as well as on the input processing and I/O side from ATHLET optimalisation could be done to on one side reduce the resource intensity of the approach and on the other side to provide stable CPU capacity to perform the analysis. Approaches for user defined output with the option to turn off other data saving could significantly reduce the time. Reserving exclusive resources on the HPC cluster is already feasible, nevertheless it has drawbacks. Initial allocation takes longer and therefore testing before live calculations would be more difficult. Furthermore, if the chains do not converge within the set limits, the analysis might have to be repeated. This could be addressed by saving chain information before convergence.

The question of the likelihood function

Revisiting the consideration of constraining the prior and posterior pdfs in order to easier derive the likelihood could be beneficial. Likelihood free methods are quite powerful, but generally more resource intensive. There is a trade-off here between generality (general application of the method to fundamentally different problems) of the approach and resource requirements to derive the pdfs in a specific question.

Surrogate modelling

To make it more feasible for more complex models an initial step of creating a meta-model seems necessary. Carrying out the vast number of simulations needed to properly explore the posterior distribution is only feasible if the run time is negligible. In this study a run averaged at about 5 minutes of runtime. Nevertheless, as experience showed this already causes problems, especially taking into account the unreliability of the resources.

As mentioned above the basis of SET and CET should be enlarged to apply the methodology on more than one experiments and set the inverse uncertainty quantification on a wider basis. Beside reflooding experiments as used in this report other test series with additional phenomena should be investigated.

By participating in working groups, international projects and networks accompanying experiments GRS could influence and contribute to determining the definition of new experiments, the detailed specification of experiments, the selection of the quantities to be measured and the scope and detail of the documentation of experiments. In this way, findings and open questions from the development of the research programme, from the validation of experiments on other scales and from the application of the computer programmes could be translated into concrete requirements for the performance of experiments also in the light of inverse uncertainty quantification. Furthermore, the discussion of the test results in the working groups, international projects and networks and their safety-related evaluation provided an early overview of the significance of the tests for AC^2 validation and inverse uncertainty quantification. The presentation of own analyses in the accompanying working groups, projects or networks enables the exchange of experience with the experimenters and with analysts who use other advanced computational programmes.

References

/ABU 19/ Abu Saleem, R. A., Kozlowski, T.: Effect of mesh refinement on the estimation of model input parameters using Inverse Uncertainty Quantification. Annals of Nuclear Energy, Vol. 132, pp. 271–276, DOI 10.1016/j.anucene.2019.04.044, 2019.

/ABU 19/ Abu Saleem, R. A., Kozlowski, T.: Estimation of probability Density Functions for model input parameters using inverse uncertainty quantification with bias terms. Annals of Nuclear Energy, Vol. 133, pp. 1–8, DOI 10.1016/j.anucene.2019.05.005, 2019.

/ADA 10/ Adams, A., Eldred, M., Dalbey, K., Bohnhoff, W., Adams, B., Swiler, L., Hough, P., Gay, D., Eddy, J., Haskell, K.: DAKOTA : a multilevel parallel object-oriented framework for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis. Version 5.0, user's manual. Sandia National Laboratories (SNL), SAND2010-2183, DOI 10.2172/991842, 2010.

/AND 13/ Andrew Collette: Python and HDF5. Ed.: O'Reilly, 2013.

/AUS 21/ Austregesilo, H., Schöffel, P. J., von der Cron, D., Weyermann, F., Wielenberg, A., Wong, K. W.: ATHLET 3.3 User's Manual. Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) gGmbH, GRS-P-1/Vol. 1 Rev. 9, November 2021.

/AUS 22/ Austregesilo, H., Schöffel, P. J., von der Cron, D., Weyermann, F., Wielenberg, A., Wong, K. W.: ATHLET 3.3.1 User's Manual, GRS-P-1/Vol. 1 Rev. 10. Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) gGmbH, October 2022.

/BAC 20/ Baccou, J., Zhang, J., Fillion, P., Damblin, G., Petruzzi, A., Mendizábal, R., Reventós, F., Skorek, T., Couplet, M., Iooss, B., Oh, D.-Y., Takeda, T., Sandberg, N.: SAPIUM: A Generic Framework for a Practical and Transparent Quantification of Thermal-Hydraulic Code Model Input Uncertainty. Nuclear Science and Engineering, Vol. 194, No. 8-9, pp. 721–736, DOI 10.1080/00295639.2020.1759310, 2020.

/BAC 20/ Baccou, J., Zhang, J., Fillion, P., Damblin, G., Petruzzi, A., Mendizábal, R., Reventos, F., Skorek, T., Couplet, M., Iooss, B., Oh, D.-Y., Takeda, T.: SAPIUM: Development of a Systematic Approach for Input Uncertainty Quantification of the Physical Models in Thermal-Hydraulic Codes, Good Practices Guidance Report. OECD Nuclear Energy Agency (NEA), NEA/CSNI/R(2020)16, NEA/CSNI/R(2020)16, 2020.

/BAC 24/ Baccou, J., Glantz, T., Ghione, A., Sargentini, L., Fillion, P., Damblin, G., Sueur, R., Iooss, B., Fang, J., Liu, J., Yang, C., Zheng, Y., Ui, A., Saito, M., et al.: A systematic approach for the adequacy analysis of a set of experimental databases: Application in the framework of the ATRIUM activity. Nuclear Engineering and Design, Vol. 421, pp. 113035, DOI 10.1016/j.nucengdes.2024.113035, 2024.

/BAY 63/ Bayes: LII. An essay towards solving a problem in the doctrine of chances. By the late Rev. Mr. Bayes, F. R. S. communicated by Mr. Price, in a letter to John Canton, A. M. F. R. S. Philosophical Transactions of the Royal Society of London, Vol. 53, pp. 370–418, DOI 10.1098/rstl.1763.0053, 1763.

/BLE 17/ Blei, D. M., Kucukelbir, A., McAuliffe, J. D.: Variational Inference: A Review for Statisticians. Journal of the American Statistical Association, Vol. 112, No. 518, pp. 859–877, DOI 10.1080/01621459.2017.1285773, 2017.

/BMUV 22/ Sicherheitsanforderungen an Kernkraftwerke vom 22. November 2012 (SiAnf) zuletzt geändert 25. February 2022 (BAnz AT 15.03.2022 B3).

/BOY 89/ Boyack, B., Duffey, R., Griffith, P., Lellouche, G., Levy, S., Rohatgi, U., Wilson, G., Wulff, W., Zuber, N.: Quantifying Reactor Safety Margins, Application of Code Scaling, Applicability, and Uncertainty Evaluation Methodology to a Large-Break, Loss-of-Coolant Accident. EG&G Idaho, Inc., NUREG/CR-, No. 5249, 31 December 1989.

/BRO 11/ Brooks, S., Gelman, A., Jones, G., Meng, X.-L. (Eds.): Handbook of Markov Chain Monte Carlo. ISBN 9780429138508, DOI 10.1201/b10905, Chapman and Hall/CRC: New York, 2011.

/CAC 10/ Cacuci, D. G., Ionescu-Bujor, M.: Best-Estimate Model Calibration and Prediction through Experimental Data Assimilation—I: Mathematical Framework. Nuclear Science and Engineering, Vol. 165, No. 1, pp. 18–44, DOI 10.13182/NSE09-37B, 2010.

/CAC 10/ Cacuci, D. G., Ionescu-Bujor, M.: Sensitivity and Uncertainty Analysis, Data Assimilation, and Predictive Best-Estimate Model Calibration. Vol. 3, pp. 1913–2051, DOI 10.1007/978-0-387-98149-9_17, 2010.

/CHA 24/ Chai, Q.: Virtual RNC benchmark Comparison Report, Draft 1. Ed.: University of Pisa, I., SI-NET-01-2023, February 2024.

/CRÉ 01/ Crécy, A. de: Determination of the uncertainties of the constitutive relationships in the CATHARE 2 code. In: American Nuclear Society (ANS) (Ed.): Proceedings ANS International Meeting on Mathematical Methods for Nuclear Applications 2001, September 2001,. M&C 2001, Salt Lake City, Utah, USA, 9 - 13 September 2001, ISBN 9780894486616, American Nuclear Society: La Grange Park, Ill., 2001.

/CRÉ 04/ Crécy, A. de, Bazin, P.: Quantification of the uncertainties of the physical models of CATHARE 2. In: American Nuclear Society (ANS) (Ed.): Proceedings of the International Meeting on Updates in Best Estimate Methods in Nuclear Installations Safety Analysis. BE-2004, Washington, D.C., 14 - 18 November 2004, ISBN 978-0894486814, American Nuclear Society: La Grange Park, Ill., 2004.

/CRÉ 08/ Crécy, A. de, Bazin, P., Glaeser, H., Skorek, T., Joucla, J., Probst, P., Fujioka, K., Chung, B. D., Oh, D.-Y., Kyncl, M., Pernica, R., Macek, J., Meca, R., Macian, R., et al.: Uncertainty and sensitivity analysis of the LOFT L2-5 test: Results of the BEMUSE programme. Nuclear Engineering and Design, Vol. 238, No. 12, pp. 3561–3578, DOI 10.1016/j.nucengdes.2008.06.004, 2008.

/CRÉ 12/ Crécy, A. de: Circe: A methodology to quantify the uncertainty of the physical models of a code. Commissariat à l'Énergie Atomique et aux Énergies Alternatives (CEA), STMF/LGLS, 2012.

/CSNI 16/ Committee on the Safety of Nuclear Installations (CSNI): PREMIUM, a benchmark on the quantification of the uncertainty of the physical models in the system thermal-hydraulic codes: methodologies and data review. OECD Nuclear Energy Agency (NEA), NEA/CSNI/, R(2016)9, 132 p., 2016.

/DAM 18/ Damblin, G., Gaillard, P.: A Bayesian framework for quantifying the uncertainty of physical models integrated into thermal-hydraulic computer codes. In: American Nuclear Society (ANS) (Ed.): Proceedings of ANS Best Estimate Plus Uncertainty International Conference (BEPU 2018). BEPU 2018, Lucca, Italy, 13 - 19 May 2018, Paper no. 96, Lucca, Italy, 2018, 2018.

/DAM 20/ Damblin, G., Gaillard, P.: Bayesian inference and non-linear extensions of the CIRCE method for quantifying the uncertainty of closure relationships integrated into thermal-hydraulic system codes. Nuclear Engineering and Design, Vol. 359, DOI 10.1016/j.nucengdes.2019.110391, 2020.

/DAU 95/ D'Auria, F., Debrecin, N., Galassi, G. M.: Outline of the Uncertainty Methodology Based on Accuracy Extrapolation. Nuclear Technology, Vol. 109, No. 1, pp. 21–38, DOI 10.13182/NT109-21, 1995.

/DIG 84/ Diggle, P. J., Gratton, R. J.: Monte Carlo Methods of Inference for Implicit Statistical Models. Journal of the Royal Statistical Society: Series B (Statistical Methodology), Vol. 46, No. 2, pp. 193–212, DOI 10.1111/j.2517-6161.1984.tb01290.x, 1984.

/DIN 03/ Dinh, T. N., Nourgaliev, R., Theofanous, T. G.: Unterstanding the Ill-Posed Two-Fluid Model. In: American Nuclear Society (ANS) (Ed.): NURETH-10, The Tenth International Topical Meeting on Nuclear Reactor Thermal Hydraulics, October 5 - 11, 2003, Seoul, Korea. Korean Nuclear Society (KNS), Seoul, 5 - 11 October 2003, ISBN 8988852117: Dajeon, 2003.

/DOU 01/ Doucet, A., Freitas, N. de, Gordon, N. (Eds.): Sequential Monte Carlo methods in practice. Statistics for engineering and information science, 581 p., ISBN 0387951466, DOI 47093, Springer: New York, Berlin, Heidelberg, 2001.

/ERI 01/ Eric Jones, Travis Oliphant, Pearu Peterson, and others: SciPy: Open Source Scientific Tools for Python. Available from http://www.scipy.org/, as at 2001-.

/ERI 10/ Eric Gazoni, Charlie Clark: Openpyxl: A Python library to read/write Excel 2010 xlsx/xlsm files. 2010.

/FEA 12/ Fearnhead, P., Prangle, D.: Constructing Summary Statistics for Approximate Bayesian Computation: Semi-Automatic Approximate Bayesian Computation. Journal of the Royal Statistical Society: Series B (Statistical Methodology), Vol. 74, No. 3, pp. 419–474, DOI 10.1111/j.1467-9868.2011.01010.x, 2012.

/FIS 78/ Fischer, G.: Analytische Geometrie. Rororo-Vieweg Grundkurs Mathematik, Vol. 35, 199 p., ISBN 3499270358, Rowohlt Taschenbuch Verl.: Reinbek bei Hamburg, 1978.

/FOR 77/ Forster, O.: Analysis 2, Differentialrechnung im Rn Gewöhnliche Differentialgleichungen. Rororo-Vieweg Grundkurs Mathematik, Vol. 31, 163 p., ISBN 3499270315, Rowohlt: Reinbek, 1977.

/FOR 09/ Forrester, A. I. J., Keane, A. J.: Recent advances in surrogate-based optimization. Progress in Aerospace Sciences, Vol. 45, No. 1-3, pp. 50–79, DOI 10.1016/j.paerosci.2008.11.001, 2009.

/GEM 84/ Geman, S., Geman, D.: Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. IEEE transactions on pattern analysis and machine intelligence, Vol. 6, No. 6, pp. 721–741, DOI 10.1109/TPAMI.1984.4767596, 30 November 1984.

/GEW 92/ Geweke, J.: Evaluating the Accuracy of Sampling-Based Approaches to the Calculation of Posterior Moments. In: Bernardo, J. M., Berger, J. O., Dawid, P., Smith, A. F. M. (Eds.): Bayesian Statistics 4. pp. 169–194, ISBN 9780198522669, DOI 10.1093/oso/9780198522669.003.0010, Oxford University PressOxford, 1992.

/GIT 21/ Git Community: git --everything-is-local. Available from https://git-scm.com/, retrieved on 15 March 2021.

/GLA 08/ Glaeser, H.: GRS Method for Uncertainty and Sensitivity Evaluation of Code Results and Applications. Science and Technology of Nuclear Installations, Vol. 2008, pp. 1–7, DOI 10.1155/2008/798901, 2008.

/GOR 93/ Gordon, N. J., Salmond, D. J., Smith, A. F. M.: Novel approach to nonlinear/non-Gaussian Bayesian state estimation. IEE Proceedings F (Radar and Signal Processing), Vol. 140, No. 2, pp. 107, DOI 10.1049/ip-f-2.1993.0015, April 1993.

/GOT 85/ Gottula, R. C., Condie, K. G., Sundaram, R. K., Neti, S., Chen, J. C., Nelson, R. A.: Forced convective, non-equilibrium, post-CHF heat transfer experiment data and correlations comparison report. EG&G Idaho, Inc., NUREG, CR-3193, March 1985.

/GOU 93/ Gourieroux, C., Monfort, A., Renault, E.: Indirect inference. Journal of Applied Econometrics, Vol. 8, S1, pp. S85-S118, DOI 10.1002/jae.3950080507, 1993.

/GRA 81/ Gradstejn, I. S., Ryzik, I. M.: Tables of series, products, and integrals. ISBN 978-3871443503, Deutsch: Thun, 1981.

/HAR 20/ Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., et al.: Array programming with NumPy. Nature, Vol. 585, No. 7825, pp. 357–362, DOI 10.1038/s41586-020-2649-2, 2020.

/HAS 70/ Hastings, W. K.: Monte Carlo sampling methods using Markov chains and their applications. Biometrika, Vol. 57, No. 1, pp. 97–109, DOI 10.1093/biomet/57.1.97, April 1970.

/HEI 81/ Heidelberger, P., Welch, P. D.: A spectral method for confidence interval generation and run length control in simulations. Communications of the ACM, Vol. 24, No. 4, pp. 233–245, DOI 10.1145/358598.358630, 1981.

/HEI 83/ Heidelberger, P., Welch, P. D.: Simulation Run Length Control in the Presence of an Initial Transient. Operations Research, Vol. 31, No. 6, pp. 1109–1144, DOI 10.1287/opre.31.6.1109, 1983.

/HEO 14/ Heo Jaeseok, LEE, S.-W., Kim, K. D.: IMPLEMENTATION OF DATA ASSIMILATION METHODOLOGY FOR PHYSICAL MODEL UNCERTAINTY EVALUATION USING POST-CHF EXPERIMENTAL DATA. Nuclear Engineering and Technology, Vol. 46, No. 5, pp. 619–632, DOI 10.5516/NET.02.2013.083, 2014.

/HEO 15/ Heo, J., Kim, K. D.: PAPIRUS, a parallel computing framework for sensitivity analysis, uncertainty propagation, and estimation of parameter distribution. Nuclear Engineering and Design, Vol. 292, pp. 237–247, DOI 10.1016/j.nucengdes.2015.07.002, 2015.

/HOF 85/ Hofer, E., Krzykacz-Hausmann, B., Ehrhardt, J., Fischer, F., Crick, M. J., Kelly, G. N.: Uncertainty and Sensitivity Analysis of Accident consequence Sub-models. In: Electric Power Research Institute (EPRI): Proc. ANS/ENS International Topical Meeting on Probabilistic Safety Methods and Applications. San Francisco, 14 February - 1 March 1985, Vol. 2, pp. 130.1–130.17, 1985.

/HOF 11/ Hoffman, M. D., Andrew Gelman: The No-U-turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. Ed.: Journal of Machine Learning, 1593-1623, 15. ed., 2011.

/HOL 22/ Hollands, T., Austregesilo, H., Buchholz, S., Di Nora, V. A., Dünne, N., Eckert, D., Junk, M., Schöffel, P. J., Wack, J., Wielenberg, A.: ATHLET 3.3.1 Validation. GRS-P1/Vol.3 Rev. 7, November 2022.

/HOL 23/ Hollands, T., Buchholz, S., Di Nora, V. A., Dünne, N., Eckert, D., Junk, M., Krüssenberg, A., Rezchikova, A., Schöffel, P. J., Wielenberg, A.: ATHLET 3.4.0 Validation. GRS-P-1/Vol. 3 Rev. 8, November 2023.

/HU 16/ Hu, G., Kozlowski, T.: Inverse uncertainty quantification of trace physical model parameters using BFBT benchmark data. Annals of Nuclear Energy, Vol. 96, pp. 197–203, DOI 10.1016/j.anucene.2016.05.021, 2016.

/HUN 07/ Hunter, J. D.: Matplotlib: A 2D Graphics Environment. Computing in Science and Engineering, Vol. 9, No. 3, pp. 90–95, DOI 10.1109/MCSE.2007.55, 2007.

/IAEA 08/ International Atomic Energy Agency (IAEA): Best estimate safety analysis for nuclear power plants, Uncertainty evaluation. Safety reports series, Vol. 52, STI/PUB/1306, 199 p., Internat. Atomic Energy Agency: Vienna, 2008.

/IAEA 16/ International Atomic Energy Agency (IAEA): Safety Assessment for Facilities and Activities, General Safety Requirements. IAEA Safety Standards Series, GSR Part 4 (Rev. 1), 163 p., ISBN 978-92-0-109115-4, IAEA: Vienna, 2016.

/IAEA 19/ International Atomic Energy Agency (IAEA): Deterministic Safety Analysis for Nuclear Power Plants, Specific Safety Guide. IAEA Safety Standards Series, SSG-2 (Rev. 1), ISBN 978-92-0-102119-9, IAEA, 2019.

/IHL 84/ Ihle, P., Rust, K.: FEBA - Flooding Experiments with Blocked Arrays, Evaluation Report, KfK 3657. Kernforschungszentrum Karlsruhe (FZK), March 1984.

/IHL 84/ Ihle, P., Rust, K.: FEBA - Flooding Experiments with Blocked Arrays, Data Report 1, KfK 3658. Kernforschungszentrum Karlsruhe (FZK), March 1984.

/JÄN 79/ Jänich, K.: Lineare Algebra, Ein Skriptum für das 1. Semester. Hochschultext, 236 p., ISBN 354009458X, Springer: Berlin, Heidelberg, 1979.

/JON 98/ Jones, D. R., Schonlau, M., Welch, W. J.: Efficient Global Optimization of Expensive Black-Box Functions. Journal of Global Optimization, Vol. 13, No. 4, pp. 455–492, DOI 10.1023/A:1008306431147, 1998.

/JOU 08/ JOUCLA, J., Probst, P.: DIPE: Determination of Input Parameters Uncertainties Methodology Applied to CATHARE V2.5_1. Journal of Power and Energy Systems, Vol. 2, No. 1, pp. 409–420, DOI 10.1299/jpes.2.409, 2008.

/KIN 13/ Kingma, D., Welling, M.: Auto-Encoding Variational Bayes. ICLR. Universiteit van Amsterdam, 2013.

/KLO 20/ Kloos, M.: SUSA Version 4.2, User's Guide and Tutorial. GRS-P-5, Vol. 1, Rev. 6, October 2020.

/KOV 14/ Kovtonyuk, A.: DEVELOPMENT OF METHODOLOGY FOR EVALUATION OF UNCERTAINTIES OF SYSTEM THERMAL-HYDRAULIC CODES' INPUT PARAMETERS. Dissertation, 265 p., Scuola di Dottorato in Ingegneria "Leonardo da Vinci", Università di Pisa (UNIPI): Pisa, Italy, 2014.

/KOV 17/ Kovtonyuk, A., Lutsanych, S., Moretti, F., D'Auria, F.: Development and assessment of a method for evaluating uncertainty of input parameters. Nuclear Engineering and Design, Vol. 321, pp. 219–229, DOI 10.1016/j.nucengdes.2016.08.021, 2017.

/KRU 15/ Kruschke, J. K.: Doing Bayesian data analysis, A tutorial with R, JAGS, and Stan. 2. ed., 759 p., ISBN 9780124058880, Elsevier Academic Press: Amsterdam, Boston, Heidelberg, London, New York, Oxford, Paris, San Diego, San Francisco, Singapore, Sydney, Tokyo, 2015.

/KUM 19/ Kumar, R., Carroll, C., Hartikainen, A., Martin, O.: ArviZ a unified library for exploratory analysis of Bayesian models in Python. Ed.: Journal of Open Source Software: The Open Journal, 2019.

/LER 19/ Lerchl, G., Austregesilo, H., Langenfeld, A., Schöffel, P. J., von der Cron, D., Weyermann, F.: ATHLET 3.2 User's Manual. Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) gGmbH (GRS), GRS-P-1/Vol. 1 Rev. 8, February 2019.

/LOF 80/ Loftus, M. J., Hochreiter, L., Conway, C. E., Dodge, C. E., Tong, A., Rosal, E. R., Valkovic, M. M., Wong, S.: PWR FLECHT SEASET Unblocked Bundle, Forced and Gravity Reflood Task, Data Report, Volume I. Westinghouse Electric Company LLC (WEC): Washington, DC, June 1980.

/MAR 11/ Marin, J.-M., Pudlo, P., Robert, C. P., Ryder, R.: Approximate Bayesian Computational methods. 5 January 2011.

/MCK 10/ McKinney, W.: Data Structures for Statistical Computing in Python. In: 9th Python in Science Conference. SciPy 2010, Austin, Texas, USA, 28 June - 3 July 2010, pp. 56–61, DOI 10.25080/Majora-92bf1922-00a, 2010.

/NEA 11/ OECD Nuclear Energy Agency (NEA): BEMUSE Phase VI Report: Status report on the area, classification of the methods, conclusions and recommendations. NEA/CSNI/R(2011)4, 28 March 2011.

/NEA 17/ OECD Nuclear Energy Agency (NEA): Post-BEMUSE Reflood Model Input Uncertainty Methods (PREMIUM) Benchmark, Final Report. August 2017.

/NEA 18/ OECD Nuclear Energy Agency (NEA): SAPIUM: Development of a Systematic Approach for Input Uncertainty Quantification of the Physical Models in Thermal-Hydraulic Codes, Good Practice Guidance Report. NEA/CSNI/R(2019)xx, 2018.

/NEA 23/ OECD Nuclear Energy Agency (NEA): Application Tests for Realisation of Inverse Uncertainty quantification and validation Methodologies in thermal-hydraulics - ATRIUM. Available from https://www.oecd-nea.org/jcms/pl_83562, as at 2023.

/NIN 24/ NINE (Ed.): Best Estimate Plus Uncertainty International Conference (BEPU 2024). Real Collegio, Lucca, Tuscany, Italy, 2024.

/NIST 24/ National Institute of Standards and Technology (NIST): 8 Incomplete Gamma and Related Function, 8.17 Related Functions, Version 1.2.0. Digital Library of Mathematical Functions, as at 5 June 2024, available from https://dlmf.nist.gov/8.17, 2024.

/NRC 19/ U.S. Nuclear Regulatory Commission (NRC): NRC Regulations Part 50 Domestic Licensing of Production and Utilization Facilities. Code of Federal Regulations, 10 CFR 50, as at 23 September 2019, available from https://www.nrc.gov/reading-rm/doc-collections/cfr/part050/full-text.html, retrieved on 7 October 2019.

/PAN 18/ Panicker, N., Passalacqua, A., Fox, R. O.: On the hyperbolicity of the two-fluid model for gas–liquid bubbly flows. Applied Mathematical Modelling, Vol. 57, pp. 432–447, DOI 10.1016/j.apm.2018.01.011, 2018.

/PER 10/ Perez, M., Reventos, J., Batet, L., Pericas, R., Toth, I., Bazin, P., Crecy, A. de, Germain, P., Borisov, S., Glaeser, H., Skorek, T., Joucla, J., Probst, P., Ui, A., et al.: Main Results of phase IV BEMUSE Project: Simulation of LB LOCA in a NPP, Science and Technology of Nuclear Installations. 10 p., 2010.

/PET 08/ Petruzzi, A.: DEVELOPMENT AND APPLICATION OF METHODOLOGIES FOR SENSITIVITY ANALYSIS AND UNCERTAINTY EVALUATION OF THE RESULTS OF THE BEST ESTIMATE SYSTEM CODES APPLIED IN NUCLEAR TECHNOLOGY. Dissertation, 364 p., Scuola di Dottorato in Ingegneria “Leonardo da Vinci”, Università di Pisa (UNIPI): Pisa, Italy, December 2008.

/PET 10/ Petruzzi, A., Cacuci, D. G., D’Auria, F.: Best-Estimate Model Calibration and Prediction through Experimental Data Assimilation—II: Application to a Blowdown Benchmark Experiment. Nuclear Science and Engineering, Vol. 165, No. 1, pp. 45–100, DOI 10.13182/NSE09-37C, 2010.

/PET 11/ Petruzzi, A., Kovtonyuk, A., Raucci, M., De Luca, D., Veronese, F., D'Auria, F.: A Procedure for Characterizing the Range of Input Uncertainty Parameters by the Use of the FFTBM. In: OECD/CSNI: Proceedings of OECD/CSNI Workshop Best Estimate Methods and Uncertainty Evaluations. OECD/CSNI Workshop Best Estimate Methods and Uncertainty Evaluations, Barcelona, Spain, 16 - 18 November 2011, NEA/CSNI/R(2013)8/PART2, pp. 61–77, 2011.

/PET 14/ Petruzzi, A., D’Auria, F.: Uncertainties in Predictions by System Thermal-Hydraulic Codes: The CASUALIDAD Method. In: American Society of Mechanical Engineers (ASME) (Ed.): 22nd International Conference on Nuclear Engineering, Volume 2B: Thermal Hydraulics. ICONE-22, Prague, Czech Republic, 7 - 11 July 2014, ISBN 978-0-7918-4591-2, DOI 10.1115/ICONE22-31274, American Society of Mechanical Engineers, 2014.

/PET 19/ Petruzzi, A.: The CASUALIDAD Method for Uncertainty Evaluation of Best-Estimate System Thermal-Hydraulic Calculations. Nuclear Technology, Vol. 205, No. 12, pp. 1554–1566, DOI 10.1080/00295450.2019.1632092, 4 February 2019.

/PLU 23/ Plummer, M. (Ed.): JAGS - Just Another Gibbs Sampler. Available from https://mcmc-jags.sourceforge.io/, as at 23 November 2023.

/POR 19/ Porter, N. W.: Wilks' formula applied to computational tools: A practical discussion and verification. Annals of Nuclear Energy, Vol. 133, pp. 129–137, DOI 10.1016/j.anucene.2019.05.012, 2019.

/PYT 23/ PyTensor Development Team.: PyTensor: A Python library for fast computation. 2023.

/PYT 23/ Python Software Foundation.: Python Language Reference, version 3.12. Available from https://docs.python.org/3/, as at 2023.

/RAF 92/ Raftery, A. E., Lewis, S. M.: [Practical Markov Chain Monte Carlo]: Comment: One Long Run with Diagnostics: Implementation Strategies for Markov Chain Monte Carlo. Statistical Science, Vol. 7, No. 4, DOI 10.1214/ss/1177011143, November 1992.

/RAF 95/ Raftery, A., Lewis, S. M.: The Number of Iterations, Convergence Diagnostics and Generic Metropolis Algorithms. Ed.: Practical Markov Chain Monte Carlo, University of Washington, 1995.

/RAJ 14/ Rajesh Ranganath, Sean Gerrish, David Blei: Black Box Variational Inference. In: Society for Artificial Intelligence and Statistics: 17th International Conference on Artificial Intelligence and Statistics. AISTATS 2014, Reykjavik, Island, 22 - 25 April 2014, pp. 814–822, 2014.

/RAZ 12/ Razavi, S., Tolson, B. A., Burn, D. H.: Review of surrogate modeling in water resources. Water Resources Research, Vol. 48, No. 7, DOI 10.1029/2011WR011527, July 2012.

/RCO 23/ R Core Team: R: A Language and Environment for Statistical Computing. Ed.: R Foundation for Statistical Computing: Vienna, Austria, 2023.

/ROS 77/ Rosal, E. R., Conway, C. E., Krepinevich, M. C.: FLECHT low flooding rate skewed test series data report. Westinghouse Electric Corporation, WCAP-9108: Pittsburgh, PA, USA, May 1977.

/RUB 84/ Rubin, D. B.: Bayesianly Justifiable and Relevant Frequency Calculations for the Applied Statistician. The Annals of Statistics, Vol. 12, No. 4, pp. 1151–1172, DOI 10.1214/aos/1176346785, 1984.

/SAC 69/ Sachs, L.: Statistische Auswertungsmethoden. 2. ed., 677 p., ISBN 978-3540046950, Springer: Berlin, Heidelberg, New York, 1969.

/SAL 16/ Salvatier, J., Wiecki, T. V., Fonnesbeck, C.: Probabilistic programming in Python using PyMC3. PeerJ Computer Science, Vol. 2, pp. e55, DOI 10.7717/peerj-cs.55, 2016.

/SCH 24/ Schöffel, P. J., Di Nora, V. A., Eckert, D., Junk, M., Cron, D. von der, Weyermann, F., Wielenberg, A.: ATHLET 3.4.1 User's Manual. Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) gGmbH, GRS-P-1/Vol 1 Rev. 12, May 2024.

/SHR 16/ Shrestha, R., Kozlowski, T.: Inverse uncertainty quantification of input model parameters for thermal-hydraulics simulations using expectation–maximization under Bayesian framework. Journal of Applied Statistics, Vol. 43, No. 6, pp. 1011–1026, DOI 10.1080/02664763.2015.1089220, 2016.

/SIM 01/ Simpson, T. W., Poplinski, J. D., Koch, P. N., Allen, J. K.: Metamodels for Computer-based Engineering Design: Survey and recommendations. Engineering with Computers, Vol. 17, No. 2, pp. 129–150, DOI 10.1007/PL00007198, 2001.

/SKO 12/ Skorek, T.: Description of FEBA Test Facility and FEBA/SEFLEX Experimental Program, OECD/NEA PREMIUM Benchmark. Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) gGmbH (GRS), Technical Note: Garching b. München, October 2012.

/SKO 13/ Skorek, T., Crécy, A. de: PREMIUM - Benchmark on the quantification of the uncertainty of the physical models in the system thermal-hydraulic codes. In: OECD/CSNI (Ed.): Proceedings of OECD/CSNI Workshop 2013. OECD/CSNI Workshop 2013, NEA/CSNI/R(2013)8, 2013.

/SKO 16/ Skorek, T., Tiborcz, L.: Validation of ATHLET system code – scaling effects by simulation of reflooding experiments. In: Korean Nuclear Society (KNS): 11th International Topical Meeting on Nuclear Thermal-hydraulics, Operation and Safety, Proceedings. NUTHOS-11, Gyeongju, Korea, 9 - 13 October 2016, 2016.

/SKO 17/ Skorek, T.: Input uncertainties in uncertainty analyses of system codes: Quantification of physical model uncertainties on the basis of CET (combined effect tests). Nuclear Engineering and Design, Vol. 321, pp. 301–317, DOI 10.1016/j.nucengdes.2016.10.028, 2017.

/SKO 18/ Skorek, T.: V&V and Uncertainty Quantification of Code Models. In: American Nuclear Society (ANS) (Ed.): Proceedings of ANS Best Estimate Plus Uncertainty International Conference (BEPU 2018). BEPU 2018, Lucca, Italy, 13 - 19 May 2018, Paper no 110, Lucca, Italy 2018, 2018.

/SKO 19/ Skorek, T.: Verification and Validation and Uncertainty Quantification of Code Models. Nuclear Technology, Vol. 205, No. 12, pp. 1540–1553, DOI 10.1080/00295450.2019.1580532, 2019.

/SKO 22/ Skorek, T.: SAPIUM: Good Practices for Thermal-hydraulic Codes Model Input Uncertainty Quantification, Open issues and predictive capability assessment. In: American Nuclear Society (ANS) (Ed.): 19th International Topical Meeting on Nuclear Reactor Thermal Hydraulics (NURETH-19). NURETH-19, Brussels, Belgium (virtual), 6 - 11 March 2022, ISBN 9789076971261, 2022.

/STA 24/ Stan Development Team: Stan User’s Guide, Version 2.36. Accessed 2024.

/SUN 13/ Sunnåker, M., Busetto, A. G., Numminen, E., Corander, J., Foll, M., Dessimoz, C.: Approximate Bayesian computation. PLoS computational biology, Vol. 9, No. 1, pp. e1002803, DOI 10.1371/journal.pcbi.1002803, 10 January 2013.

/TEN 24/ Tensorflow: Probability. Available from https://www.tensorflow.org/probability, as at 2024.

/THE 22/ The MathWorks Inc.: MATLAB version: 9.13.0 (R2022b). Natick, Massachusetts, United States, 2022.

/TIB 15/ Tiborcz, L.: Validation of the Reflooding Model in the Code ATHLET. Diploma Thesis, INSTITUTE OF NUCLEAR TECHNIQUES, Budapest University of Technology and Economics: Budapest (Hungary), 2015.

/TIB 24/ Tiborcz, L., Beck, S.: GRS Results and Lessons Learned during the EC-MUSA Project. In: NINE (Ed.): Best Estimate Plus Uncertainty International Conference (BEPU 2024). Real Collegio, Lucca, Tuscany, Italy, pp. Paper 340, 2024.

/TUK 47/ Tukey, J. W.: Non-Parametric Estimation II. Statistically Equivalent Block and Tolerance Regions - The Continuous Case. The Annals of Mathematical Statistics, Vol. 18, No. 4, pp. 529–539, 1947.

/UNA 11/ Unal, C., Williams, B., Hemez, F., Atamturktur, S. H., McClure, P.: Improved best estimate plus uncertainty methodology, including advanced validation concepts, to license evolving nuclear reactors. Nuclear Engineering and Design, Vol. 241, No. 5, pp. 1813–1833, DOI 10.1016/j.nucengdes.2011.01.048, 2011.

/VAN 95/ van Rossum, G., Drake, Fred L., Jr.: Python reference manual. Ed.: Centrum voor Wiskunde en Informatica Amsterdam, 1995.

/VIN 07/ Vinai, P., Macian-Juan, R., Chawla, R.: A statistical methodology for quantification of uncertainty in best estimate code physical models. Annals of Nuclear Energy, Vol. 34, No. 8, pp. 628–640, DOI 10.1016/j.anucene.2007.03.003, 2007.

/VOJ 82/ Vojtek, I.: Untersuchung der Wärmeübertragungsverhältnisse in der Hochdruckphase eines Kühlmittelverluststörfalls mit mittlerem und grossem Bruchquerschnitt. Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) gGmbH, GRS-A-Bericht, GRS-A-709: Garching, March 1982.

/WAL 43/ Wald, A.: An Extension of Wilks' Method for Setting Tolerance Limits. The Annals of Mathematical Statistics, Vol. 14, No. 1, pp. 45–55, DOI 10.1214/aoms/1177731491, 1943.

/WAS 21/ Waskom, M.: seaborn: statistical data visualization. Journal of Open Source Software, Vol. 6, No. 60, pp. 3021, DOI 10.21105/joss.03021, 2021.

/WEY 24/ Weyermann, F., Eschricht, D., Wielenberg, A., Steinhoff, T., Schöffel, P. J., Spengler, C., Lovasz, L.: AC² 2023.2 User Manual. Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) gGmbH, GRS-P-15/Vol. 1 Rev. 2, 80 p., May 2024.

/WIE 19/ Wielenberg, A., Lovasz, L., Pandazis, P., Papukchiev, A., Tiborcz, L., Schöffel, P. J., Spengler, C., Sonnenkalb, M., Schaffrath, A.: Recent improvements in the system code package AC2 2019 for the safety analysis of nuclear reactors. Nuclear Engineering and Design, No. 354, pp. 110211, DOI 10.1016/j.nucengdes.2019.110211, 2019.

/WIL 41/ Wilks, S. S.: Determination of Sample Sizes for Setting Tolerance Limits. The Annals of Mathematical Statistics, Vol. 12, No. 1, pp. 91–96, DOI 10.1214/aoms/1177731788, 1941.

/WIL 42/ Wilks, S. S.: Statistical Prediction with Special Reference to the Problem of Tolerance Limits. The Annals of Mathematical Statistics, Vol. 13, No. 4, pp. 400–409, DOI 10.1214/aoms/1177731537, 1942.

/WIL 48/ Wilks, S. S.: Order Statistics. Bulletin of the American Mathematical Society, Vol. 54, No. 1, pp. 6–50, 1948.

/WOO 10/ Wood, S. N.: Statistical inference for noisy nonlinear ecological dynamic systems. Nature, Vol. 466, No. 7310, pp. 1102–1104, DOI 10.1038/nature09319, 2010.

/WU 18/ Wu, X., Kozlowski, T., Meidani, H., Shirvan, K.: Inverse uncertainty quantification using the modular Bayesian approach based on Gaussian process, Part 1: Theory. Nuclear Engineering and Design, Vol. 335, pp. 339–355, DOI 10.1016/j.nucengdes.2018.06.004, 2018.

/WU 18/ Wu, X., Kozlowski, T., Meidani, H., Shirvan, K.: Inverse uncertainty quantification using the modular Bayesian approach based on Gaussian Process, Part 2: Application to TRACE. Nuclear Engineering and Design, Vol. 335, pp. 417–431, DOI 10.1016/j.nucengdes.2018.06.003, 2018.

/WU 18/ Wu, X., Kozlowski, T., Meidani, H.: Kriging-based inverse uncertainty quantification of nuclear fuel performance code BISON fission gas release model using time series measurement data. Reliability Engineering & System Safety, Vol. 169, pp. 422–436, DOI 10.1016/j.ress.2017.09.029, January 2018.

/WU 19/ Wu, X., Shirvan, K., Kozlowski, T.: Demonstration of the Relationship between Sensitivity and Identifiability for Inverse Uncertainty Quantification. Journal of Computational Physics, No. 396, pp. 12–30, 2019.

/WU 21/ Wu, X., Xie, Z., Alsafadi, F., Kozlowski, T.: A comprehensive survey of inverse uncertainty quantification of physical model parameters in nuclear system thermal–hydraulics codes. Nuclear Engineering and Design, Vol. 384, pp. 111460, DOI 10.1016/j.nucengdes.2021.111460, 2021.

/ZHA 19/ Zhang, J., Dethioux, A., Kovtonyuk, A., Schneidesch, C.: Development of a Pragmatic Approach to Model Input Uncertainty Quantification for BEPU Applications. Nuclear Technology, Vol. 205, No. 1-2, pp. 140–152, DOI 10.1080/00295450.2018.1516055, 2019.

List of Figures

List of Tables

Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) gGmbH

Schwertnergasse 1
50667 Köln
Telefon +49 221 2068-0
Telefax +49 221 2068-888

Boltzmannstraße 14
85748 Garching b. München
Telefon +49 89 32004-0
Telefax +49 89 32004-300

Kurfürstendamm 200
10719 Berlin
Telefon +49 30 88589-0
Telefax +49 30 88589-111

Theodor-Heuss-Straße 4
38122 Braunschweig
Telefon +49 531 8012-0
Telefax +49 531 8012-200

www.grs.de

ISBN 978-3-910548-78-7