TUM School of Computation, Information and Technology

Uncertainty Quantification and Machine Learning Surrogate Models for Multi-Scale High-Performance-Computing Plasma Physics Turbulent Transport Simulations

Yehor Yudin, M. Sc. (hons)

Vollständiger Abdruck der von der TUM School of Computation, Information and Technology der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitz:		Prof. Dr. Eric Sonnendrücker
Prüfer der Dissertation	1.	Hon.-Prof. Dr. Frank Jenko
	2.	Prof. Dr. Hans-Joachim Bungartz

Die Dissertation wurde am 16.04.2024 bei der Technischen Universität München eingereicht und durch die TUM School of Computation, Information and Technology am 05.07.2024 angenommen.

DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Doctoral Thesis

Uncertainty Quantification and Machine Learning Surrogate Models for Multi-Scale High-Performance-Computing Plasma Physics Turbulent Transport Simulations

Author:	Yehor Yudin
Supervisor:	Prof. Dr. Frank Jenko
Advisor:	Dr. David Coster
Submission Date:	16.04.2024

Abstract

One of the questions that arises when it comes to analysing mathematical models' trustworthiness and their practical application is the quantification of uncertainties related to the model. This process requires the interpretation of different types of uncertainties, identification of its sources, choice of particular quantitative metrics, design of algorithms to obtain these metrics, and computational means to perform the algorithm. In most cases, such uncertainty analysis is based on extensive parametric studies of the solution of the analysed model, which is computationally prohibitively expensive for many real-world relevant problems.

This work presents an application of uncertainty quantification to a coupled system of equations describing the evolution of plasma in the core of a toroidal device for magnetic confinement of plasma. This work focuses on analysing plasma behaviour and associated mathematical uncertainties in plasma transport due to turbulent processes, which are inherently non-linear and computationally expensive to model, prone to parametric instabilities, and challenging to capture in their parametric dependencies. The discussed turbulence processes are the dominant mechanism of heat and particle transport in fusion plasmas, and the ability to model them quickly and account for uncertainties is crucial for the design of nuclear fusion reactors.

This dissertation suggests a method for quantifying uncertainties in the transport fluxes computed by a turbulence model and uncertainties arising in plasma's core temperature and density profiles. The work presents the software workflows for uncertainty quantification and their application to multi-scale multi-component turbulent transport simulation workflow, which allowed the models' sensitivity to uncertainty to be captured and the most influential quantities to be revealed, and to quantitatively balance the uncertainty estimation accuracy and the required computational cost.

Furthermore, this work demonstrates an application of data-driven machine learning surrogate models to substantially speed up the simulation workflow and uncertainty quantification procedures. It also discusses multiple aspects of surrogate modelling, including the required data, training and validation, control of their fidelity, and active learning.

Kurzfassung

Eine der Fragen, die sich bei der Analyse der Vertrauenswürdigkeit mathematischer Modelle und ihrer praktischen Anwendung stellt, ist die Quantifizierung der mit dem Modell verbundenen Unsicherheiten. Dieser Prozess erfordert die Interpretation verschiedener Arten von Unsicherheiten, die Identifizierung ihrer Quellen, die Wahl bestimmter quantitativer Metriken, die Entwicklung von Algorithmen zur Ermittlung dieser Metriken und rechnerische Mittel zur Durchführung des Algorithmus. In den meisten Fällen basiert eine solche Unsicherheitsanalyse auf umfangreichen parametrischen Studien der Lösung des analysierten Modells, was für viele in der Praxis relevante Probleme unvertretbar rechnenintensiv ist.

In vorliegenden Arbeit wird eine Anwendung der Unsicherheitsquantifizierung auf ein gekoppeltes Gleichungssystem vorgestellt, welches die Entwicklung eines Fusionsplasmas im Kern eines torusförmigen Gefässes zum magnetischen Plasmaeinschluss beschreibt. Diese Arbeit konzentriert sich auf die Analyse des Plasmaverhaltens und der damit verbundenen mathematischen Unsicherheiten beim Plasmatransport aufgrund turbulenter Prozesse, die von Natur aus nichtlinear und rechenintensiv zu modellieren sind, anfällig für parametrische Instabilitäten sind, und für welche es schwierig ist die parametrischen Abhängigkeiten zu erfassen. Die erwähnten Turbulenzprozesse sind der dominierende Mechanismus des Wärme- und Teilchentransports in Fusionsplasmen, und die Fähigkeit, sie schnell zu modellieren und Unsicherheiten zu berücksichtigen, ist für das Design von Kernfusionsreaktoren entscheidend.

Diese Dissertation schlägt eine Methode zur Quantifizierung von Unsicherheiten vor, sowohl von denen im Turbulenzmodell berechneten Transportflüssen, als auch jene die in den Temperatur- und Dichteprofilen des Plasmas auftreten. In der Arbeit werden die Software-Workflows für die Quantifizierung von Unsicherheiten und ihre Anwendung auf Multiskalen-Multikomponenten-Simulationen des turbulenten Transports vorgestellt. Sie ermöglichen es, die Empfindlichkeit der Modelle gegenüber Unsicherheiten zu erfassen und die einflussreichsten Größen aufzudecken sowie ein quantitatives Balance zwischen der Genauigkeit der Unsicherheitsabschätzung und den erforderlichen Rechenkosten zu finden.

Darüber hinaus demonstriert diese Arbeit eine Anwendung von datenbasierten Maschinellen-Lernen-Surrogatmodellen, die zu einer wesentlichen Beschleunigung des Simulationsablaufs und des Verfahrens zur Unsicherheitquantifizierung, und erörtert mehrere Aspekte der Surrogatmodellierung, einschließlich der erforderlichen Daten, des Trainings und der Validierung, der Kontrolle der Genauigkeit und des aktiven Lernens.

List of Publications

Publications

1. Y. Yudin, D. Coster, U. von Toussaint, and F. Jenko. "Epistemic and Aleatoric Uncertainty Quantification and Surrogate Modelling in High-Performance Multiscale Plasma Physics Simulations". en. In: *Computational Science – ICCS 2023*. Ed. by J. Mikyška, C. de Mulatier, M. Paszynski, et al. Cham: Springer Nature Switzerland, 2023, pp. 572–586. ISBN: 978-3-031-36027-5. DOI: `10.1007/978-3-031-36027-5_45`

2. O. O. Luk, J. Lakhlili, Y. Yudin, et al. "Performing Validation, Verification, and Sensitivity Analysis on Multiscale Fusion Plasma Simulations with the VECMA Toolkit". In: *Proceedings of 47th EPS Conference on Plasma Physics*. Geneva: European Physical Society, 2021. URL: `https://hdl.handle.net/21.11116/0000-0009-A9D1-C`

3. Y. Yudin, J. Lakhlili, O. Luk, et al. "Gaussian Process surrogate models for uncertainty quantification in multiscale fusion simulations". In: *Proceedings of International Conference on Computational Science 2021 (conference abstract)*. 2021

Conference presentations

1. Y. Yudin et al. "Epistemic and Aleatoric Uncertainty Quantification and Surrogate Modelling in High-Performance Multiscale Plasma Physics Simulations". In: *International Conference on Computational Science 2023*. Prague, Czech Republic, July 3, 2023

2. Y. Yudin et al. "Uncertainty Quantification for Multiscale Turbulent Transport Simulations". In: *DPG SMuK (Poster)*. Dresden, Germany, Mar. 22, 2023

3. Y. Yudin et al. "Uncertainty Quantification for Multiscale Turbulent Transport Simulations". In: *DPG SMuK (Poster)*. Mainz, Germany (virtual), Mar. 25, 2023

4. Y. Yudin et al. "Gaussian Process Surrogate Models for Uncertainty Quantification in Multiscale Fusion Simulations". In: *International Conference on Computational Science 2021 (Talk)*. Krakow, Poland (virtual), June 18, 2021

5. Y. Yudin et al. "Gaussian Process Surrogate Models for Uncertainty Quantification in Multiscale Turbulent Transport Simulations". In: *DPG SMuK (Poster)*. virtual, Aug. 26, 2021

Acknowledgments

I would like to express my gratitude to those without whom this Ph.D. thesis would not have been possible and to everyone who supported me on the way.

First of all, I want to thank Frank Jenko, my academic supervisor, for the opportunity to join the IPP and work on this project. I would like to thank Hans-Joachim Bungartz for the chance to join the Munich School of Data Science and for being an examination committee member. I also want to thank Eric Sonnendrücker for agreeing to chair my examination committee.

I would like to express my immense gratitude to David Coster, my scientific mentor, for his tireless supervision, advice, discussions, help, collaborative coding and debugging, and relentless and comprehensive support in every aspect from the very beginning of the project.

I want to thank Udo von Toussaint, my scientific council, for his frequent and helpful advice on the broadest range of topics. Furthermore, I want to express gratitude to Jalal Lakhlili, Onnie Luk, Olivier Hoenen, as well as Roland Preuss and Christopher Albert, the people involved in VECMA project at IPP, for their help in the first steps of the project and introduction to the fields of research related to this work.

My thanks go to Lourence Veen and Maarten Sebregts for their tireless support on the MUSCLE3 library and Bartosz Bosak and Tomasz Piontek for their constant help with QCG-PJ library. Also, I thank Wouter Edeling, Derek Groen, and other international members of VECMA and SEAVEA collaboration for the discussions and mutual support in software development and usage.

I thank Michael Bergmann, Kislaya Ravi, Constantin Gahr, Victor Artigues, Robin Greif, and other members of MuDS for frequent conversations on multiple topics and assistance with reviewing.

Thank you, my office mates, Felix Wilms, Leonhard Leppin, Verena Mitterauer, and all others across the last few years, for conversations and help with cross-language experience.

I am very grateful to my TOK colleagues and all the other people at IPP with whom I built connections for sharing knowledge and for their help and company.

Finally and most importantly, I thank my friends and family in Munich, Ukraine, or elsewhere in the world for their constant support and source of resilience during challenging times.

Contents

1. Introduction

1.1. Nuclear Fusion

Nuclear Fusion is a process of merging nuclei, usually two of them, into a larger nucleus. For nuclei of a small atomic number, the product nucleus mass is smaller than the sum of the initial nuclei mass, and the surplus binding energy is usually released in the form of gamma radiation or transferred to fast remainder particles. Being able to utilise this process as an industrial source of energy has enormous potential as it would require only fuel, which is abundant and has low technological hazard risk. Nuclear fusion happens when the kinetic energy of reacting particles is sufficient to overcome the electrostatic forces. In ensembles of particles, this process is described via cross-section σ measure of probability for such a reaction to occur, which is a function of the temperature. In practice, such energies are much higher than those binding the electrons and nuclei, meaning nuclear fusion reactions *en masse* would happen in a medium of fully ionised gas or plasma. Such processes are happening inside stars and are a dominant source of radiation of stars and hence energy coming to planets like Earth. Even though the dominant type of fusion reaction in the Sun is a proton-proton chain leading to the merging of several ^1H nuclei into a ^4He nuclei, the reactions with highest cross-section possible are one between Deuterium and Tritium. Hence, this $D-T$ reaction is of the highest interest to tackle by human technology.

Since this type of reaction is happening in the plasma medium, the challenge of mastering nuclear fusion is creating and controlling plasmas with the required conditions. One of the ways to achieve this is magnetically confined fusion based on trapping plasma into a toroidally configured magnetic field in order to separate plasma from the external medium [9]. Due to the basic topological properties of solenoidal vector fields like the magnetic field, the most straightforward configuration of such a field to form a closed trap would be toroidal.

1.1.1. Tokamak

One of the types of such devices used to confine the plasma, currently in the purpose of performing various experiments on plasma physics and exploring possibilities for the creation of a practical nuclear fusion reactor, is called a *tokamak*, which bears its name after Russian abbreviations of “тороидальная магнитная камера с катушками” and “тороидальная камера с аксиальным магнитным полем” meaning “a toroidal magnetic chamber with coils” or “a toroidal chamber with an axial magnetic field”. The other type is a *stellarator*, named after the Latin “stēlla” for “star”.

The latter type creates a magnetic field required for plasma confinement with only external coils of complex shape, which usually leads to toroidally asymmetric fields. Conversely, the tokamaks create nearly perfectly axisymmetric fields and, unlike stellarators, induce toroidal currents inside the plasma.

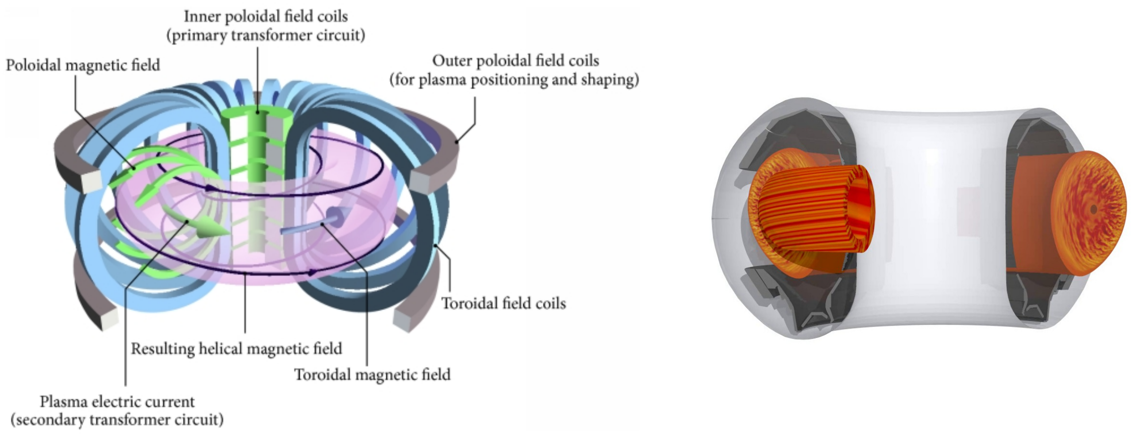
Inner poloidal field coils
(primary transformer circuit)
Poloidal magnetic field
Outer poloidal field coils
(for plasma positioning and shaping)
Toroidal field coils
Resulting helical magnetic field
Toroidal magnetic field
Plasma electric current
(secondary transformer circuit)

Figure 1.1.: On the left schematic description of a tokamak, with principle coils and magnetic field components depicted, credited to [10]. On the right: an image of a simulation of turbulence in ASDEX Upgrade tokamak performed by gyrokinetic code GENE [11].

Such a configuration determines a particular type of coordinate system used for plasma modelling. One of the ways to define a point inside such a toroidal geometry is via a coordinate system based on three scalar coordinates: radial coordinate, toroidal angle, and poloidal angle. Each coordinate can be defined based on one of the multiple principles that usually includes a concrete magnetic field distribution inside a torus. Here, some critical properties indifferent to all of the coordinate choices are that most of the processes inside such a toroidal device are symmetric with respect to the toroidal angle. Furthermore, the toroidal motion of the plasma is the fastest component of its dynamics.

One of the principal features of such a device is that it creates a helically winding toroidal magnetic field, as shown in figure 1.1 with the poloidal component being created by the toroidal currents induced in the plasma itself. Such a magnetic field structure assures the forces necessary to confine plasma inside the toroidal region [9].

The most essential scalar quantities to describe the quality of such a magnetical confinement device are the maximal particle density n, plasma temperature T, and the confinement time τ_E. The latter parameter estimates the time required for all energy to leave the plasma. It is usually defined as a ratio of plasma energy $W \approx 3VnT$ and the heating power P required to keep in steady state $\tau_E = W/P$. The product of these three quantities $nT\tau_E$ is one of the figures of merit describing the quality of the fusion plasma and is called the *triple product*.

This work considers the tokamak type of magnetic confinement devices and model, specifically the *ASDEX Upgrade (AUG)* device from IPP, Garching, Germany [12]. However, the presented methodology can be applied to other tokamaks, including the ones in the construction or planning process, like ITER, and, to some degree and with substantial modification, to stellarators.

1.1.2. Plasma Modelling and Magnetised Plasma Turbulence

Plasma can be described mathematically in a number of ways. Since the description of the dynamics of a single particle, which a thermonuclear device may contain the order of 10^{23}, is out of the question for any approach, the most precise first-principle description of a plasma is via a *kinetic model*. This framework describes the distribution function f of an ensemble of particles over the phase space $\mathbb{X} \times \mathbb{V}$, the product $3D$ configurational space and $3D$ velocity space, and the conservation of the distribution function of time leads to Boltzmann equation. The presence of an electromagnetic field and charged particles requires the inclusion of Lorentz force, which leads to the *Vlasov equation*

$$\frac{\partial f_\alpha}{\partial t} + \boldsymbol{v}_\alpha \cdot \nabla_{\boldsymbol{x}} f_\alpha + \frac{Z_\alpha e}{m_\alpha}\left(\boldsymbol{E} + \frac{\boldsymbol{v}_\alpha}{c} \times \boldsymbol{B}\right) \cdot \nabla_{\boldsymbol{v}} f_\alpha = 0 \tag{1.1}$$

where α is species index, Z_α is effective charge of species, e.g., -1 for electrons and $+1$ for deutrons. Coupling this equation with the Maxwell equations for the field $\boldsymbol{E}$ and $\boldsymbol{B}$ leads to the equations that comprehensively describe plasma as an ensemble of particles, each with 6 *degrees of freedom (DoFs)*.

The next type of model in the hierarchy of complexity is *gyrokinetics*, which reduces the dimensionality of the velocity space to $2D$ by integrating over the smallest scale of dynamics, which is the gyration of particles in the magnetic field, yielding effectively a $5D$ model. The other family of models that describe the velocity space of finite-dimensional structure are *fluid* models. They describe only a finite number of moments of the distribution function in velocity space and do not capture kinetic effects in plasma. A combination of these two approaches, namely reducing the equations to avoid gyrations resolution and expressing equations in terms of moments, leads to the *gyrofluid* model of plasma [13]. This type of model is the model of choice for describing plasma turbulence in this work.

Furthermore, a description of plasma as a conducting fluid, characterised by a finite number of moments interacting with the field via Lorentz force, leads to the plasma model of *Magnetohydrodynamics (MHD)*.

As in neutral fluid, described by *Navier-Stokes (NS)* equations, a phenomenon of turbulence occurs in plasmas, bearing both similarities and differences to fluid turbulence. Common properties include non-linear interactions of multiple scales, the requirement of injection of energy into the system on certain scales, dissipation of energy on smaller scales, and the emergence of an energy cascade across scales. Furthermore, given a strong background magnetic field, resulting in high anisotropy of the system, fusion plasma turbulence bears resemblance to $2D$ fluid one, which happens in thin layers, for example, in the Earth's atmosphere. The kinetic effects and non-local interaction, however, introduce dissimilarities in different dissipation scales and mechanisms and a variety of instability classes not inherent to fluids [13, 14].

The free energy sources required to drive turbulence in fusion plasmas are mainly temperature and density gradients. One of the types of plasma turbulence significant for heat and particle transport in the core of fusion plasmas is *Ion Temperature Gradient (ITG)* turbulence. This type of turbulence is of particular relevance for the chosen cases to model, and as its main drive is the ion temperature gradient ∇T_i, this quantity is the main independent parameter of choice in this work.

Due to the limitations of the analytic results of most of the high-fidelity plasma models, including gyrokinetic and gyrofluid ones, the study and predictions of plasmas require numerical solutions for those models. This is done through various spatial and time discretisation methods, like *Finite Element methods (FEM)* or *Fourier Spectral* methods, and via a variety of techniques to solve the resulting linear systems and integrate equations in time. Those are implemented via highly performant parallelised program codes suited to be run on *High-Performance-Computing (HPC)* systems and require $\gg \mathcal{O}(10^3)$ hours of computations on a core (core-hours, CPUh) for cases relevant to experimental reactors.

An important way to characterise plasma behaviour is to consider the transport of the main conserved quantities. One usually considers the transport of the particles, or particle density n, moment, and heat. In this work, heat transport is the process of choice to study. From the conservation equation point of view, one can define the transport of a quantity q via the divergence of its flux, in the differential form expressed as $\frac{\partial q}{\partial t} + \nabla \cdot \boldsymbol{Q} = S$. The S denotes sources and sinks of the quantity, which are inconsistent with a considered closed system model; in the context of fusion plasmas, it could be energy loss due to radiation or injection of energy via a heating system. Further vector analysis allows for distinguishing different processes responsible for transport or other separate channels. For the heat flux, we can distinguish the diffusive end convective parts $\boldsymbol{Q} = -\chi n \nabla T + \boldsymbol{v} n T$, with a diffusive part being characterised by diffusion coefficient $D = \chi n$ and convective bulk transport with convective velocity $\boldsymbol{v}_{\text{conv}}$.

It is observed by experiment and supported by the theory that in the core of fusion transport, most of the transport is performed via a microscopic turbulent process, which is primarily effectively attributed to diffusive channels. Due to high nonlinearity and inherently stochastic dynamics, turbulence and effective turbulent transport have various associated uncertainties requiring their quantification.

1.2. Uncertainty Quantification

As natural sciences aim to uncover the truth about reality, it is essential to allow science to distinguish what is known from what is not. However, due to the fact that all knowledge is partial, as well as due to imperfections in the way knowledge is acquired, the boundary between known truth and the absence of understanding is blurred; thus, generalisation or extrapolation of knowledge requires a notion of uncertainty. Out of two primary ways of acquiring knowledge in natural science, namely experimental and theoretical, the former one, while being ultimately more immediately connected to the studied reality, has developed ways to understand how uncertain the discovered facts are in two main ways: one based on estimating how imprecise the measurement mechanism is, and another based on evaluating the effects of a finite sample of observations, meaning estimating the statistical errors. The latter way, namely the theoretical one, while naturally more directed towards producing well-generalisable knowledge, still needs to introduce the notion of uncertainty. The most elaborated way to produce theoretical knowledge is by creating mathematical models of the studied real processes or phenomena, which in physics are usually formulated as systems of ordinary or *partial differential equations (PDEs)*. These systems of equations have to be solved in order to be able to say anything about the studied systems, which, in general cases, could be done only via methods of numerical mathematics.

There are many ways for a solution of a given mathematical model to be imprecise or uncertain in the description of the actual reality. In general, the best mathematical tool to understand such uncertainty is probability theory, which requires introducing the notion that particular situations or outcomes could only be understood as random, and a probability measure should be assigned to them. The goal of *uncertainty quantification (UQ)* is to provide quantitive statements about probabilities of quantities related to the mathematical models and their solutions when such quantities are understood as random variables [15, 16].

Applying uncertainty quantification to mathematical models in natural sciences allows, among other things, establishing a common statistical understanding of the quantities between their observations and computation and, for example, estimating various risks associated with the models under uncertainty [17].

1.2.1. Epsitemic and Aleatoric Uncertainty

It makes sense to attribute uncertainty to different types for a better conceptual and algorithmic characterisation of uncertainties in mathematical models. An important categorisation of uncertainties deals with their sources, relationship to knowledge beyond the model itself, and how a person can influence the model uncertainty with this knowledge. The two types of uncertainties presented are *epistemic* and *aleatoric*, and, as it will be discussed later in this work, applying such a distinction to practical uncertainty quantification has practical benefits.

Epistemic uncertainty

The type of uncertainty most discussed within various scientific areas is epistemic uncertainty, which is concerned with a lack of information. These uncertainty sources could include a lack of knowledge about the real system described by the model, a lack of knowledge of a particular model form, or, what is discussed most often, a lack of information on the specific values of model parameters. In the latter case, one usually deals with a specific estimate of some numerical parameters and hence would like to question how such uncertainties in a parameter would influence the uncertainties in quantities of interest of the model solution. This type of uncertainty derives its name from $\epsilon\pi\iota\sigma\tau\eta\mu\eta$, a Greek word for knowledge or understanding.

Often it is difficult to characterise or formalise any uncertainties arising from a lack of knowledge about the model. The simplest type of epistemic uncertainties that could be incorporated into a mathematical model are the ones associated with the uncertainties of the model's parameters. In such case, an uncertain numerical parameter could be treated as a random variable, or in a more general case, a random process or random field, and its uncertainties can be characterised with a *probability density function (PDF)* or properties of the respective random mathematical objects.

The practical question would be characterise how such a parametric uncertainty influences and interacts with an uncertainty of the model's dependent *quantities of interest (QoIs)*. This requires modelling them as random variables and characterising their PDFs, often as a non-trivial function of independent random input variables.

Aleatoric uncertainty

One of the types of uncertainties in the solution for mathematical models dealing with the inherent properties of the models is the aleatoric uncertainty. It is, in the first place, related to the dynamic stochastic behaviour of the model and the quantities of interest for which it is being solved. The designated name for such type of uncertainty is derived from *alea*, Latin for dice, as it is considered that the model's behaviour is truly random and is characterised by the fact that no more information about the model and its parameters can reduce this uncertainty. This parameter-independency distinguishes this type of uncertainty from the epistemic one, primarily dealing with the lack of information, and also leads to another synonym, the *irreducible* uncertainty.

Without changing the model and its solver, it is impossible to eliminate variability in the quantity of interest, even having full precise knowledge of the parameters' values of the model. Examples of such uncertainty are the chaotic behaviour of some nonlinear dynamic systems and the turbulent behaviour of fluids or plasmas. In some cases, aleatoric uncertainty could also incorporate the uncertainty that comes from particular approaches for the model solution, like introducing a random perturbation into some particular known solutions, which could come from a random number generator. However, properties of the solution, like uncertainty due to the solver's approximation error or, in the case of a *Monte Carlo (MC)* class solver, the random noise of the solution, are usually not considered as a part of aleatoric uncertainty *per se* and are subject to the numerical analysis of the solver.

Most of the methods of aleatoric uncertainty analysis incorporate the following two principles. The first principle is the separation of epistemic uncertainties. It assumes that a particular model, its parametric form, and parameter values are given, and what is left is to analyse the stochastic properties of the model given all the knowledge about it.

The second principle is the statistical treatment of the output quantities of interest. That assumes that the behaviour of such quantity could be split into a mean or trend and the stochastic noise on top of it. Given a sample representing an ensemble of runs, one could analyse the properties of such a stochastic noise.

The result of such an analysis is a probabilistic model of some QoI as a stochastic process, given its deterministic model for mean trend value and characterisation of irreducible noise, for example, in terms of its covariance function. The understanding of physical dynamics as a stochastic process requires ergodic properties of phase space, mixing, and the existence of an equilibrium to relate the chaotic dynamical distributions with a notion of the uncertainty of a state [18].

1.2.2. Intrusive and Nonintrusive Methods

A categorisation of uncertainty quantification methods by how they engage the studied mathematical model in uncertainty analysis often distinguishes *intrusive* and *nonintrusive* methods.

A *nonintrusive* method assumes that a model is a black box capable of producing an answer, in terms of values of certain QoIs, for given values of a set of some independent input parameters. Hence, here, one can introduce a model solution function, and solving a model could be understood as an evaluation of a function value. Often, this evaluation is considered deterministic, while in the UQ framework, inputs and outputs are considered random variables (or vectors or functions). Thus, a nonintrusive UQ algorithm is supposed to capture information about the uncertainty of a random QoI in terms of its PDF or some of its statistics, having only access to deterministic function evaluations. In such a case, estimating some statistics about some model parameters requires some scheme involving evaluating model solution function for multiple independent variables values, and such a scheme could be designed for methods including numerical integration to calculate statistics as integrals or as some statistical sampling [17, 19].

On the contrary, *intrusive* methods are based on the modification of the process of the model solution itself, understanding the solution of the problem as random variables, vectors, or fields whose properties are captured via a finite set of DoFs and expressed via an orthogonal basis of the underlying Hilbert space of possible solutions [17].

Such methods increase the problem's dimensionality, which could be understood in terms of adding additional dimensions of possible values that a particular QoI can take and, in terms of increasing total number of DoFs to describe the solution, now a probabilistic one. However, due to the primary utilisation of the orthogonality of the employed bases, the total increase in complexity of the problem is significantly lower than that of nonintrusive methods.

However, the most significant disadvantage of an intrusive method is that the existing numerical framework to solve a deterministic problem no longer applies, and its program implementations or computational codes can no longer be used. Applying any intrusive UQ method requires reformulating the problem, usually in a weak form, and expanding an existing system of equations into an even larger one that accounts for a new probabilistic basis.

1.2.3. Forward and Backward Methods in Relation to Bayesian Statistics

Two major problems could be formulated within the uncertainty quantification when considering that arguments and values of a function are random variables.

The first one is associated with a situation when one has some qualitative and quantitative consideration of the independent parameters of the model and can prescribe some concrete probabilistic properties to those parameters. The question posed in such a situation is how and in which way such input parameter uncertainties would influence the uncertainties in the solution of the model, particularly in quantities of interest. Such a problem usually bears the name of *forward uncertainty quantification problem*, or *uncertainty propagation*. One of the

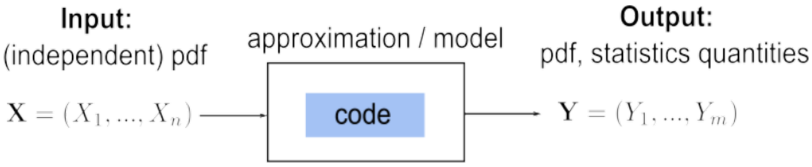
Input:
(independent) pdf
$\mathbf{X} = (X_1, ..., X_n)$
approximation / model
code
Output:
pdf, statistics quantities
$\mathbf{Y} = (Y_1, ..., Y_m)$

Figure 1.2.: A schematics of an uncertainty propagation problem, requiring characterisation of a prior distribution of independent variables, and a code to solve the problem of study deterministically, to quantify uncertainties in the quantities of interest.

key ideas for solving such a problem is to formulate it so that the uncertainties of the solution QoIs can be expressed and defined by a finite sample of problem solutions in its deterministic formulation. In this way, output uncertainties would require a finite number of solutions to the problem for different values of input parameters. As shown in the figure 1.2, this requires obtaining a sample of runs of codes by solving the problem.

The other problem is related to the characterisation of the model parameters and their uncertainties, given a set of observations describing the model solution, which in its turn could also come with uncertainties. This problem is called *inverse uncertainty quantification problem*. Given the probabilistic description of the model parameters, it is usually solved in the framework of *Bayesian calibration* or *inversion*.

In the Bayesian inversion, a forward model is used to formulate a *likelihood* model describing the probability distribution of function values for a given value of argument value $p(y|x)$. This is similar to describing the likelihood of some dependent variable y given some determined value of an observed variable x. Forward propagation of uncertainty would answer a question about what the *marginal* distribution of the dependent QoI $p(y) = \int_X p(y|x)p(x)dx$ for a given distribution of independent variable $p(x)$ is. Such a model could be based on statistics of function evaluations $y_i = f(x_i)$ and expressed, for example, as a *Gaussian Process Regression (GPR)* model [20]. The inverse problem would require formulating a conditional *posterior* for the argument given an existing *prior* and likelihood for function value:

$$p(x|y) = \frac{p(y|x)p(x)}{\int_X p(y|x)p(x)dx} \tag{1.2}$$

This could be done both as a new explicit model and as a part of an analytical expression for future computations, partially due to the difficulty of calculating prefactor $\int_X p(y|x)p(x)dx$ as an integral over a large high-dimensional input parameter space. With the posterior model for the argument, one can reconstruct its marginal distribution for a given distribution of function value as $p(x) = \int_Y p(x|y)p(y)dy$. The schematic picture of the forward and backward uncertainty quantification in Bayesian formulation is shown in figure 1.3. The domain model is used to express the likelihood of a dependent variable given for any independent variable values $p(y|x)$. The forward uncertainty problem for a given prior uncertainty of inputs $p_{\text{pr}}(x)$ is then solved through marginalization to get $p(y)$. The inverse uncertainty problem needs an expression of posterior distribution $p(x|y)$, which uses the forward model or its likelihood. Then one can reconstruct input posterior $p(x)$ for a given observable distribution

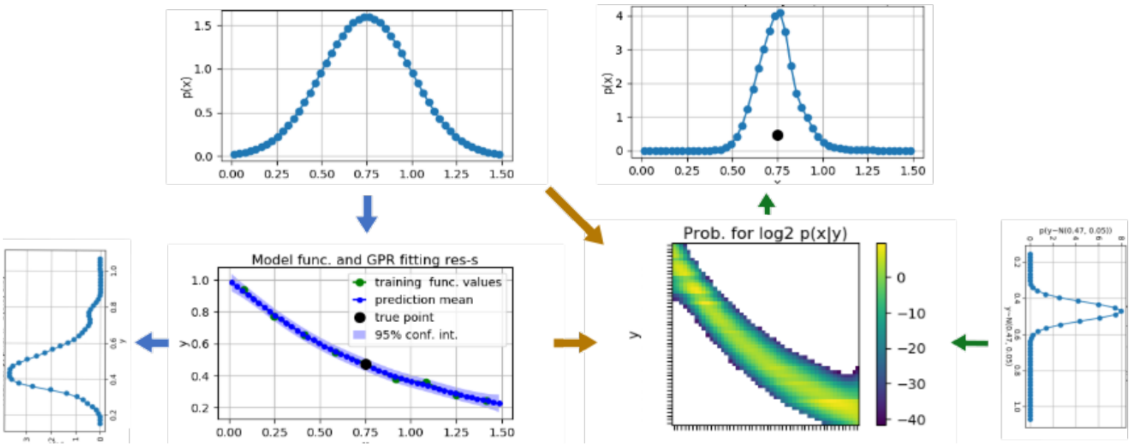

Figure 1.3.: A schematics of Bayesian approach to uncertainty quantification. The blue arrows show that one can use a distribution of an uncertain input $p(x)$ and forward likelihood model $p(y|x)$ to find out marginal output distribution $p(y) = \int_X p(y|x)p(x)dx$ and solve a forward problem. The orange arrows demonstrate the formulation of a posterior of input $p(x|y)$ using the likelihood and input prior. The orange arrows show how to solve an inverse problem and find an input uncertainty for a given output uncertainty. Here, we present a possible scenario of quantifying uncertainty in a particle or heat flux for an uncertain transport function and an uncertain plasma temperature or density value on an arbitrary scale. Furthermore, a Bayesian reconstruction of temperature or density value is outlined.

of dependent variables $p_{\text{ob}}(y)$.

1.3. Sensitivity Analysis

In most situations, one is interested in mathematical models mostly in terms of how some quantity of interest y of its solution depends on the independent variable $\boldsymbol{x} = (x_1, \dots, x_n)$. For such a reduced view, one should question how strongly this dependency $y = f(\boldsymbol{x})$ is influenced by each component of $\boldsymbol{x}$. The goal of the *sensitivity analysis (SA)* field is to answer this question by giving quantitative estimates of this influence [21].

One of the ways to analyse such sensitivities is *local* one. It answers the question of how a small perturbation of $\boldsymbol{x}$ around a particular point influences $\boldsymbol{y}$, usually utilising the model's parametric gradients $\nabla_{\boldsymbol{x}} f(\boldsymbol{x})$. The other type of sensitivity analysis is *global*, which considers a variation of input parameter $\boldsymbol{x}$ over a whole region of the domain and gives an estimate of sensitivity in the average sense.

For example, a *derivative-based* method could assume that the independent variables have a particular prior distribution $\boldsymbol{x} \sim p(\boldsymbol{x})$ and measure a global sensitivity as an integral of the models' derivative over this prior distribution $\int_X \nabla_{\boldsymbol{x}} f(\boldsymbol{x}) p(\boldsymbol{x}) dx$. However, the model's gradients to its parameters are often not analytically accessible. They have to be either reconstructed using the model's *automatic differentiation* or approximated with an intermediate of a data-based proxy model. Another type of method, *variance-based* ones, targets measuring the variance of the quantities of interest $\mathbb{V}[y]$ for given input priors.

Taking into account the multivariate nature of the dependency $y = f(x_1, \ldots, x_n)$, one would like to expand into an additive combination of functions, each dependent only on the subset of components $\mathcal{I} \subset \{1, \ldots, n\}$. Applying this type of expansion to the variance of the quantity of interest is the basis of *analysis of variance (ANOVA)* methods.

Apart from giving some general information about the model, sensitivity analysis methods are often coupled to *dimensionality reduction* methods, utilising the information on sensitivity to simplify the $y = f(\boldsymbol{x})$ dependency by decreasing the effective dimensionality of $\boldsymbol{x} \rightarrow \boldsymbol{x}_{\text{red}}$.

This work uses predominantly different types of Sobol indices, popular in ANOVA, as sensitivity metrics.

1.4. Surrogate Modelling

A class of practical tasks, interesting for many areas of science and engineering, may include multiple function evaluations, each of which, in practice, requires solving an expensive numerical problem, often obtaining a numerical solution for a system of PDEs. Such a class includes optimisation problems and uncertainty quantification. In these cases, an external loop solving the corresponding problem requires many iterations, each performing an expensive simulation. This class of problems is often denoted as *multi-query problem*, referring to querying some opaque, black-box process from the perspective of the outer loop.

Depending on the complexity, regularity, and dimensionality of the problem, applying such an algorithm may require $O(10^2 - 10^5)$ function evaluation, or even more, each often being a relatively challenging task to perform from the perspective of modern HPC systems.

One of the approaches applied for enabling the utilisation of such algorithms and solving such multi-query problems is *surrogate modelling*. It assumes that there is a possibility to extract enough information from a finite and realistic number of function evaluations, hence simulation runs, and to regress parametric dependencies of all relevant quantities of interest with sufficient accuracy over a specific parametric region of interest to form a cheap-to-evaluate proxy *surrogate model*, such that replacement of original function evaluation by usage of such cheap surrogate would lead to accurate enough solution of a multi-query problem.

1.5. Machine Learning

Machine Learning (ML) is a broad field that studies and designs algorithms capable of solving problems based on extracting knowledge from data and by iterative improvement.

By way of how an algorithm assimilates data, ML methods are usually categorised into classes of supervised, unsupervised (including semi-supervised and self-supervised subtypes), and reinforcement learning.

The main types of problems that Machine Learning algorithms can solve are *classification*, *regression*, as well as *generation*, e.g., capturing a PDF of some data and sampling from it.

The regression ML models trained on simulation data can serve as fast and efficient surrogate models introduced in section 1.4.

With the growth of the amount of data accumulated in the experimental and observational fields and computations, as well as with growing computing capabilities employed for theoretical calculations, the field of machine learning is gaining particular traction [22].

1.5.1. Problems and Methods

Supervised algorithms are based on the idea that an algorithm to solve a problem can be created based on a finite sample of problem input independent data and the problem's numerical solutions. In other words, the algorithm should generalise a specific function based on a set of function arguments and values, where the function could have arbitrary properties, including being very non-linear, non-regular, or having a domain of extremely high dimensionality.

Unsupervised learning, on the contrary, assumes that there are no given values of such a function, and the algorithm is supposed to instead identify certain features or patterns of data of a specific type and is sampled from a particular distribution. This type of algorithm solves, among others, problems of *clustering* samples into groups by some similarity criteria but without knowing specific labels or compressing or encoding data by finding patterns and representations in small dimensional latent spaces.

Reinforcement Learning (RL) algorithms assume a rather specific formulation of the problem to solve, which is based on learning the best strategies or policies for performing particular tasks. This class of algorithms assumes that there is an agent that can perform actions that change the state of its environment and that agent is able to receive rewarding or penalising signals from the environment. Under such conditions, the algorithm aims to teach the agent strategies to maximise some cumulative reward that describes how well the task is solved.

The *classification* task requires an algorithm to determine which class a sample belongs to based on the sample's features. A machine learning algorithm to perform such a task would require a data set of samples, each presented by some numerical features and a label describing its class.

The *regression* problem assumes that the value of the function, or label of the data, could be non-categorical, usually a vector $y \in \mathbb{R}^n$ and the task of the algorithm is to approximate the value of the function for previously unseen values of the arguments $\boldsymbol{x} \in \mathbb{R}^N$. This method can be applied to use finite data to capture dependencies in physics and other fields for quantities that are expensive to compute.

One of the other prominent tasks solved by the Machine Learning model is *generative modelling*, which is to create new data samples that match some properties of the learned data, or, in other words, to capture PDF of the learned data and to sample from it.

One of the critical features of all the Machine Learning algorithms is that since they are based on certain data following a certain PDF and covering a finite range of possible argument values, they are inherently ill-fitted for extrapolation and describing data coming outside of the training data distribution. This issue has to be considered in any practical application of Machine Learning algorithms, and some of the ways to validate the severity of extrapolation errors and possible remedies suitable for the concrete cases are discussed in this work.

1.6. Thesis Overview

The next chapter 2, this work will overview the methods applied, including the particular ways to quantify uncertainty, both aleatoric and epistemic, and the algorithms to do that, the approach of surrogate modelling, and the different types of surrogate models applied.

Chapter 3 will discuss the physical model studied in this work, the aspects of its computer implementation, and the aspects of multi-scale multi-component modelling of turbulent transport.

In chapter 4, the work will present the practical results obtained through the application of the methods to analyse uncertainties of the solution of turbulent codes as well as how the suggested approaches enhanced the computational efficiency of the algorithm. This chapter will show the analysis of uncertainties in fluctuating model solution QoIs, how to leverage sequential analysis to save compute while getting statistical information for the solution, and how to get and analyse statistics related to parametric uncertainties. Also, how to effectively get data, train and utilise data-based surrogate models in multi-component simulations, and ensure they maintain the fidelity of the physics they describe via adaptive retraining or active learning methods. Finally, we will demonstrate how to capture the parametric and irreducible uncertainties that arise due to turbulence in tokamak core plasma simulations and how to get this uncertainty, saving the compute resources efficiently.

Chapter 5 will discuss the application of surrogate modelling to the coupled multi-scale multi-component transport simulations. It will demonstrate the procedure of physics-informed data-based model retraining and a method to capture uncertainties of global parameters due to uncertainties in microscopic turbulence.

Finally, the last chapter 6 will summarise the work done and discuss the open questions and possible directions for future research.

2. Methodology

This chapter introduces the methods to quantify the uncertainties in the numerical simulations of plasma turbulent transport, targeting uncertainties of two types. The first is aleatoric irreducible uncertainty of fluctuating physical quantities, treated as a stochastic process. Algorithms to estimate these properties for time series are presented. The other type is epistemic parametric uncertainties, which have to be quantified via statistical and numerical treatment of the model's independent and dependent quantities as random variables. Moreover, this chapter describes methods to create and use fast data-based Machine Learning surrogate models using simulation data.

2.1. Multiscale Modelling

Often, a mathematical model describes real-world phenomena that happen simultaneously on different scales in space and time. In such cases, a property of the model on one scale is defined by the properties of the processes on other scales. Examples of such a situation are when the transport properties of a particular medium are defined by phenomena on much smaller scales that might include microscale turbulence or interactions of individual particles composing a physical medium.

The complete solution of such a model would require resolving the smallest and fastest, hence more detailed and computationally expensive, model features spanning the largest scale of the problem. However, in many situations, the similarity and homogeneity of the phenomena on the smallest scales and large discrepancy of the characteristical dimensionality of the scales allow us to assume a certain scale separation and to reuse information about smaller scales, which furthermore influences larger scales only in terms of their averaged properties.

For the time *scale separation*, from the point of view of a microscale model, the macroscopic processes are quasi-stationary $\frac{\partial y_M}{\partial t} = 0$. Conversely, from the point of view of the macroscale model, microscale processes provide an immediate response $y_m \neq y_m(t)$. For the spatial separation, the micromodel depends only on local point-wise quantities associated with the macroscale model. And visa versa, the microscopic model influences only the local parameters of the macroscopic model. To be benign for model formulation and computational cost, an additional regularity property of both macroscopic and microscopic models should accompany such a scale separation. In order to have a coarse spatial resolution, where each DoF of the macroscopic model corresponds to an independent microscopic, hence expensive, model, the quantities of the macroscopic model should be spatially well-correlated, and the macroscopic model should not have large time derivatives. In the coupling context, the microscopic model's QoI cannot be too sensitive to macroscopic parameters, which could be expressed in terms of sensitivity indices or amplification factors for uncertainties (or, in a general case, in terms of Lyapunov coefficients or Lipshitz continuity of microscopic QoIs as a function of macroscopic quantities).

In such a favourable case when the scale separation assumptions could be effectively applied, one could reformulate the problem in such a way that, at a time, only a subset of a problem should be solved, and the parameters of this subset that depend on the rest of the problem could be assumed fixed. In such a case, every scale of the model is described by a sub-model, with every sub-model coupled to others in terms of parametric dependency. Solution of the entire model would require an iterative solution of each sub-model with subsequent calculation and substitution of the parameter values to the dependent model, be it formulated in terms of forward time integration of some non-stationary process or iterative process of convergence to a specific solution self-consistent with a given global model.

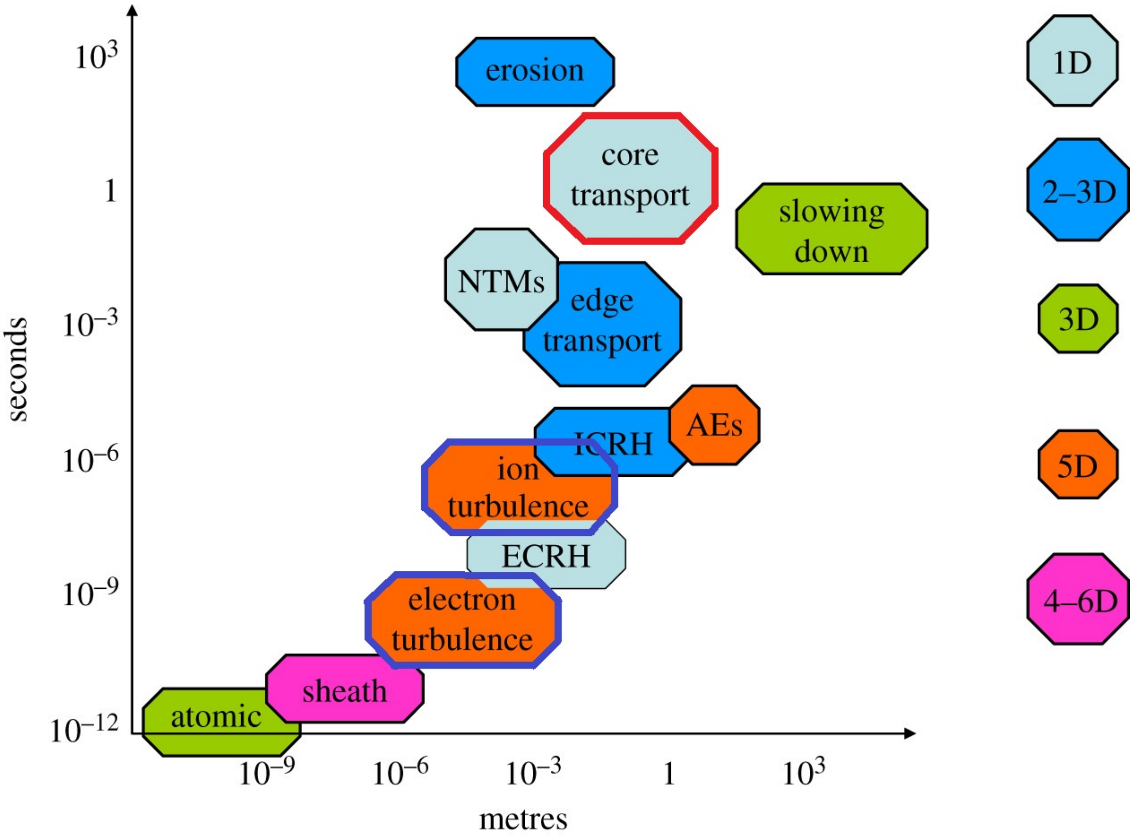

Figure 2.1.: Spatial and temporal scales of processes relevant for a tokamak plasma discharge and their physical dimensionality. This work focuses on core transport and ion turbulence processes, which have a discrepancy of $2-3$ orders of magnitude in space and ~ 6 orders in time.

In plasma physics, processes on multiple spatial and temporal scales occur within a single discharge. As shown in figure 2.1, the relevant scales in the time range from 10^{-9} to 10^{3} s, from picoseconds on atomic scales to hundreds of seconds for some wall material processes. For the spatial scale, the processes vary from 10^{-10} to 10^{3} m, from nanometers for single particle interactions to several meters describing the entire devices. In order to model the plasma of a discharge consistently, one has to accommodate this scale disparity of the processes, usually each represented by its own model. This discrepancy, often more than 6 orders of magnitude, leads to the need for methods to couple different models appropriately, both in mathematical and programmatical senses, and to find an ability to solve microscopic models fast.

In the last two decades, significant progress has been made in modelling multiscale turbulent transport in fusion plasmas. Following the primary approach in the work, we consider a multi-component approach to model this process, where turbulence, transport, and other subprocesses are each solved by its code.

A number of works were done in developing and studying workflows employing transport, turbulence, and equilibrium components, including ones across the MAPPER, COMPAT, and VECMA projects [23, 24, 25] and other projects [26, 27].

2.1.1. Multiscale Modelling Simulation Language

There are several methods of describing parametric dependencies of different model and the related pipeline for their solution, most of them describing it in terms of a modified graph. One such description approach is *Multiscale Modelling Simulation Language (MMSL)* [28].

This method describes a computational workflow as a directed graph where each vertex is a particular mathematical model, and the edges are communication conduits between the models. A model usually receives data on its independent parameters and sends out data on the QoI of its solution. In this approach, the graph's vertices are components, representing codes that solve a particular problem. The edges of the graphs, respectively, are the conduits connecting the ports of the components, describing the exchange of information on the partial solution.

Such a graph description allows for multiple patterns of model interdependencies and interactions with respect to the organisation of the models' time integration. Furthermore, it allows for a clear description of computational workflows via YAML-based configuration files. This so-called YMMSL description may also include all the information required for the program implementing the workflow, like the meta-information on the model parameters, the models' code implementations, the computational resources they need, and other configuration parameters.

This work uses the MUSCLE3 library to implement such multi-component multiscale computational workflows, updating previous implementations in the MUSCLE2 library and adding capabilities to utilise surrogate and different codes to solve the turbulence problem [29, 30, 31, 32, 28]. This library translates a YMMSL workflow description into an executable instance, runs the program components of workflows, manages the required resources, and arranges the communication between coupled codes via TCP/IP sockets.

2.2. Time Traces Analysis

This work deals with physical quantities evolving in time and exposing stochastic properties due to underlying processes' high non-linearity and parametric sensitivity. The study of aleatoric uncertainties focuses on analysing properties of such fluctuating quantities. Those are computed via time-discrete numerical codes and are considered on the level of separate scalar quantities. Thus, to study the aleatoric uncertainties in plasma turbulence, this work uses the time trace analysis methods as a tool of choice, which are discussed in this section.

2.2.1. Theory. Stochastic Processes

In a broad sense, *stochastic process* $\{X_t\}$ is a set of random variables, each indexed by an element t of some set T. Each random variable X in a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ is a function $X : \Omega \to \mathbb{R}$ measurable over is sigma-algebra $\mathcal{F}$ with a domain of sample space Ω, value in space state $\mathbb{R}$, and a probability measure $\mathbb{P} : \mathcal{F} \to [0, 1]$. In this work, we consider a narrower type of stochastic process where the indexing set T has a relationship of total order on itself. In such a case, the index for random variables is physically understood as a time reading. We consider two types of processes, a *continuous* process with indexing set isomorphic to an open interval on $\mathbb{R}$, e.g., $(0, 1)$ and a *discrete* process indexed with a set isomorphic to an interval on $\mathbb{N}$. The resulting stochastic process is used to describe a change of a scalar quantity in some process with time [33].

In this work, we mainly consider *stationary* stochastic processes, meaning that the properties of the process do not change in time, or more precisely, the joint distribution of any subset of $\{X_t\}$ does not change under translation by an arbitrary $\tau : t \to t + \tau$.

One of the important properties of a stationary stochastic process is its *autocorrelation function (ACF)* $\rho(\tau)$ describing how two of the elements of a stochastic process with indices τ distance apart are correlated $\rho_{xx}(\tau) = \mathrm{Cov}(x_t, x_{t+\tau})/\sigma_{x_t}\sigma_{x_{t+\tau}}$. For the realisation of a continuous process, it can be estimated as an expected value for a product of value with itself with a lag $\rho_{xx}(\tau) = \mathbb{E}[(X_t - \mu_{x_t})(X_{t+\tau} - \mu_{x_{t+\tau}})]/\sigma_{x_t}\sigma_{x_{t+\tau}}$, normalised to have a zero mean and unitary standard deviation. For an ergodic stationary process, it can be estimated via convolution of the values against itself with a lag $\rho_{xx}(\tau) = \lim_{T\to\infty} \frac{1}{T}\int_0^T x^*(t)x(t+\tau)dt$. Equivalently, one can compute it as a product in Fourier space $\rho_{xx}(\tau) = \mathscr{F}^{-1}\{\mathscr{F}^*\{x(t)\} \cdot \mathscr{F}\{x(t)\}\}$. In practice, computation of ACF with Fourier transform via an algorithm like FFT [34] would require padding the sample with zeros until the size is equal to the power of two [35]. Furthermore, the evaluation of ACF is sensitive to the sample size.

An advantageous type of analysis for such one-dimensional stochastic processes if *Fourier* analysis, based on the Fourier transformation of a series $\mathscr{F}(f) = \int_{-\infty}^{\infty} x(t)e^{-i2\pi ft}dt$, a convolution of the function with basic harmonics of frequency f.

In this work, the analysis of the spectra of all the heat flux time series revealed that the frequency falls off quadratically f^{-2}, which corresponds to Brownian noise behaviour and fits with the spectrum of a Wiener process or a random walk. A representative power spectrum in time of a radially localised ion heat flux value calculated by GEM is shown in figure 2.2.

Here, we also use a conception of *autocorrelation time (ACT)* defined as such an argument of

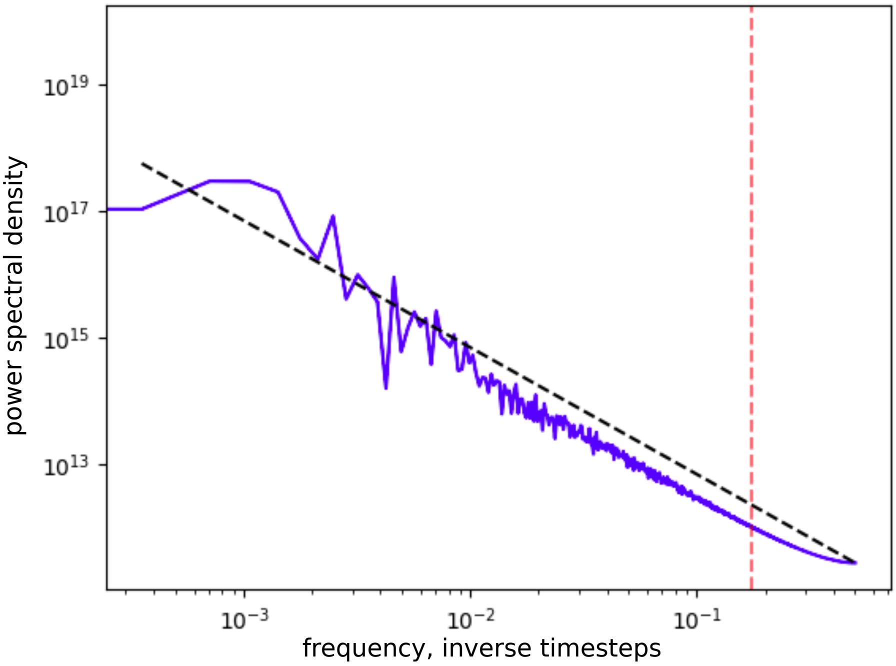

Figure 2.2.: Fourier power spectrum for outward radial ion heat flux for an AUG discharge at $\rho_{\mathrm{tor}}^{\mathrm{norm}} \approx 0.7$ from GEM. The $\log f = -2$ slope (dashed black) and 99% cumulative power cut-off (dashed red) are indicated.

ACF τ^* for which its value falls off in e times. Generally, the autocorrelation function is non-monotonic and can increase once the lag τ reaches some high-period oscillation component of the time series.

When analysing stationary series, the essential statistics (the functions of a sample) are often the main moments of the PDF of the process, as one is interested in the effective value of some quantity and its variability.

In this work, the basis of the analysis is calculating the mean of the time series $\mu[x] = \frac{1}{n}\sum_{x \in \mathbf{X}} x$ as well as the error of its estimate, *standard error of the mean (SEM)*, which the standard deviation $\sigma[x] = \sqrt{\frac{1}{n-1}\sum_{x \in \mathbf{X}} (x - \mu[x])^2}$ normalised by the square root of the sample size $\mathrm{SEM}[x] = \sigma[x]/\sqrt{n_s}$.

Unlike the analytical considerations for the continuous stochastic processes, dealing with the data from practical problems, be it real-world observations or numerical experiments, mostly requires analysing *discrete* time series. In this work, for example, we consider values of specific quantities calculated with computer codes implementing a numerical method solving a PDE, and these methods are naturally discrete in time. Even though this quantity is meant to present a continuous physical one, what is accessible to analysis is only a finite ordered set of numbers. This means that one may only obtain statistical estimates of the abovementioned characteristics. This requires numerical algorithms and introduces finite-sample statistical errors. One of the features of such discrete time series is that one cannot capture any effects associated with frequencies more than half of the sampling frequency $f \geq f_N = f_s/2$, called the *Nyquist* frequency [35]

The situation when approximates continuous processes that are dominated by frequencies much larger than the sampling frequency $f \gg f_N$ introduces another type of consideration. In this work, as in many other physical applications, the time steps between the calculation of two consecutive values of QoI (heat fluxes) are much smaller than the period of fluctuation of quantities of interest for the highest observed frequency, meaning that the time series describing the evolution of the QoI does not lose any information that the given model can provide. On the other side, taking the QoI value on every time step would represent the time series in a way hardly suitable for statistical analysis as any small feature that we would like to understand as a sample from a variety of possible stochastic behaviours of QoI would be represented by a multitude of highly correlated readings. This leads to a need to deliberately lose some information on QoI's time dependency in order not to overrepresent any features and to downsample the readings of QoI. In this work, the downsampling technique employed is based on taking a single sample for a number of consecutive readings n_w equal to the time series autocorrelation time $\mathrm{ACT}[q]$ in time steps, where the new reading is equal to an average value of all reading in such an ACT window $q_i^{\mathrm{eff}} = \frac{1}{n} \sum_{j=i \cdot \mathrm{ACT}[q]}^{(i+1) \cdot \mathrm{ACT}[q]} q_j$.

2.2.2. Data and Algorithm

The quantities that could be directly computed from the (gyro-) fluid model simulations that are of most interest for the transport model describing processes on a larger time scale and in terms of $1D$ profiles are time-averaged particle and energy fluxes, typically across flux surfaces defined by a particular radial coordinate of a tokamak plasma.

In this work, we consider two particle species, electron and main ion, nucleus of Deuterium, and the respective heat fluxes measured or calculated for a flux tube located at a particular value of $\rho_{\mathrm{tor}}^{\mathrm{norm}}$ (normalised toroidal magnetic flux ϕ) and changing with time $Q_{e,i}(\rho_{\mathrm{tor}}^{\mathrm{norm}}, t)$.

We use the $(\cdot)_e$ subscript for quantities associated with electrons, the $(\cdot)_i$ subscript for the ones related to main ion species, and the $(\cdot)_{e,i}$ subscript when talking about things that are general for both species.

Given the nature of the behaviour of the selected quantities of interest with time and our goal to characterise aleatoric uncertainties, we treat them as a time series with the following analysis steps.

Since the overall goal is to describe aleatoric uncertainty in terms of a particular noise's statistical properties on top of a quantity's mean time-wise dynamics, we would like to work with stationary time series.

The first step in preparing data for such time series is to eliminate the non-stationary part of the readings. In the context of turbulence simulations, it is part of the reading from the initial steps during which the turbulence develops until it reaches a saturated state.

The first type of statistic that is of interest in analysing such time traces and for subsequent usage in modelling is the sample mean value of the readings. This mean value is to be used to calculate an effective value of the fluctuating quantity; here, effective means is the one relevant for the resolution of the other larger scales.

In the context of turbulence and turbulent flows, even estimating such a natural first-order statistic as a mean could entail some issues, first and foremost in the high statistical noise for such an estimate [36, 37].

An example of the ion heat transport flux values growing and fluctuating with time computed with code GEM, together with the estimated statistics, could be found in figure 2.3.

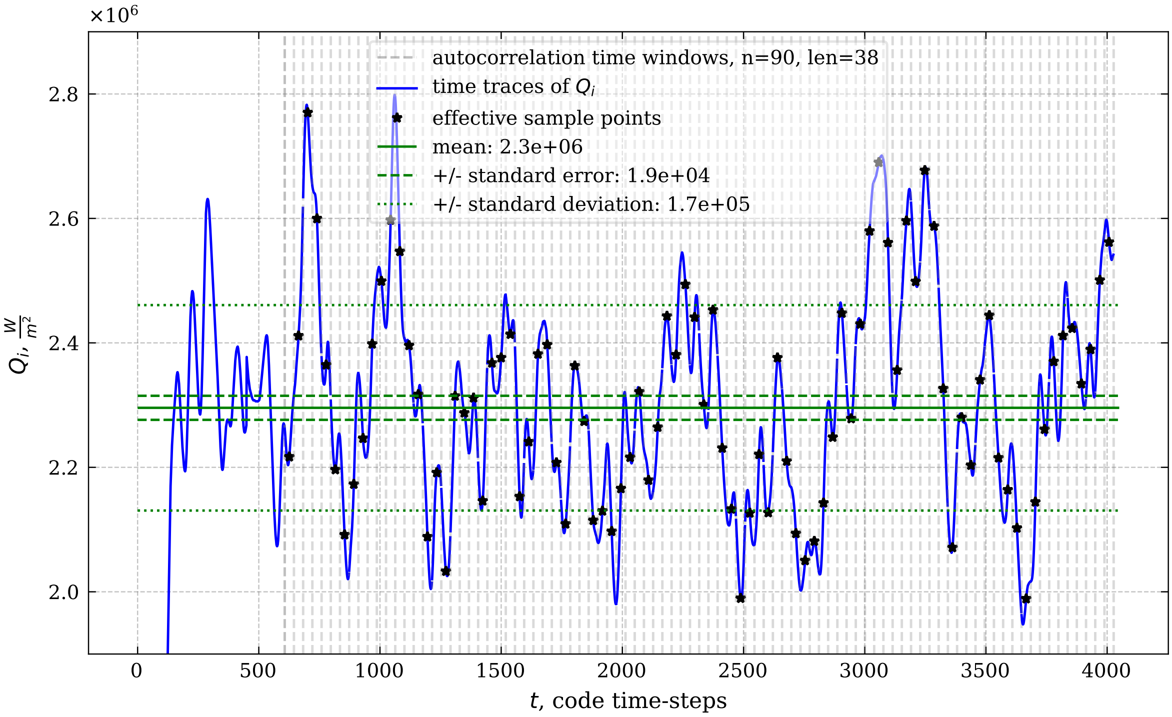

Figure 2.3.: Ion heat flux $Q_i(t)$ at $\rho_{\text{tor}}^{\text{norm}} = 0.7$ computed by a GEM flux tube simulation as a function of time. The initial growth phase readings are discarded before analysis. The autocorrelation time widows, effective averaged readings, mean, standard deviation, and standard error of the mean estimates are indicated according to the plot legend.

Even though the variance σ^2 is a more natural quantitive characteristic of the uncertainty related to a stochastic process, the additional consideration of standard error SEM of the mean helps analyse the statistical noise. Furthermore, as it is more dependent on the sample size, it could be used to provide judgment on the length of the time series, hence the duration of the turbulence simulations, necessary to capture the information on the effective quantities of interest behaviour.

In this work, we suggest a sequential procedure to analyse the convergence of the SEM of turbulent transport fluxes in the course of the time integration of the turbulence equations that allow us to judge whether the collected sample in time is sufficient to estimate the effective level of flux values.

The procedure assumes adding new readings associated with the time evolution of quantities of interest Y_i to the total sample of readings every t_n time steps of the integration. Once the sample is expanded, the autocorrelation time of time series ACT_i is recalculated, and a new effective sample is downsampled using average values of every time window of the length equal to the ACT. Using the new effective sample, the mean, standard deviation, and standard error of the mean are recalculated. The new values of these three statistics are then compared with those computed on previous procedure steps, and the relative change is then compared to the pre-selected threshold values. When the relative change of the mean and the standard error of its estimate drops below the chosen thresholds, the sample is considered sufficient, and the simulation of turbulence for given kinetic characteristics is stopped. Such an algorithm allows us to control the fidelity of the estimate of the transport fluxes by a flexible choice of the simulation duration.

The outline of the procedure to calculate the SEM of a model solution QoI using its time trace $y(t)$ in a sequential manner during the simulation is presented via the following steps in listing 2.2.2, which are also visually summarised in figure 2.4.

The procedure for analysing the QoI time traces $y(t)$, representing a model solution for a single parametric point, constitutes of the following steps:

1. For the time-traces $y(t)$ of length t_n representing a model solution in a scalar quantity of interest at time steps $\{t\}$, we select a part in the saturated phase:
   a) Define the $y(t)$ ramp-up phase duration $t_{\text{r.u.}}$: here, in practice, for the long term, we chose an initial 15% of readings
   b) Discard the readings from the ramp-up phase
2. Downsample the readings:
   a) Calculate the Auto-Correlation Time: $t_{\text{A}} = \text{ACT}[y(t)] = \min t^* :$ $\frac{1}{(t_n - t^*)} \sum_{t=t^*}^{t_n} (y(t) - \bar{y}) \cdot (y(t - t^*) - \bar{y}) < \frac{1}{t_n} \sum_{t=t_1}^{t_n} y^2(t) \cdot e^{-1}$
   b) Split time series in saturated phase into $n_{\text{eff}} = \lfloor \frac{t_n}{t_{\text{A}}} \rfloor$ windows
   c) For every autocorrelation time window of length t_{A} choose a downsampled reading as a mean value $y_i^{\text{eff}} = \frac{1}{t_{\text{A}}} \sum_{j=i \cdot t_{\text{A}}}^{(i+1) \cdot t_{\text{A}}} y_j$ for an effective time step $t_i^{\text{eff}} = \frac{1}{t_{\text{A}}} \sum_{j=i \cdot t_{\text{A}}}^{(i+1) \cdot t_{\text{A}}} j$
   d) Collect downsampled readings into a new set $\boldsymbol{Y}^{\text{eff}} = \{y_i^{\text{eff}}\}$
3. Test the stationarity of the resulting time series:
   a) Here: compute an ordinary least-squares linear multivariate regression model of downsampled QoI readings over time
   b) Apply Normal Equations to find coefficients: $\hat{\boldsymbol{\theta}} = \left(\boldsymbol{X}^\top \boldsymbol{X}\right)^{-1} \boldsymbol{X}^\top \boldsymbol{Y}$ where $\boldsymbol{X}$ consists of effective time steps t_i^{eff}

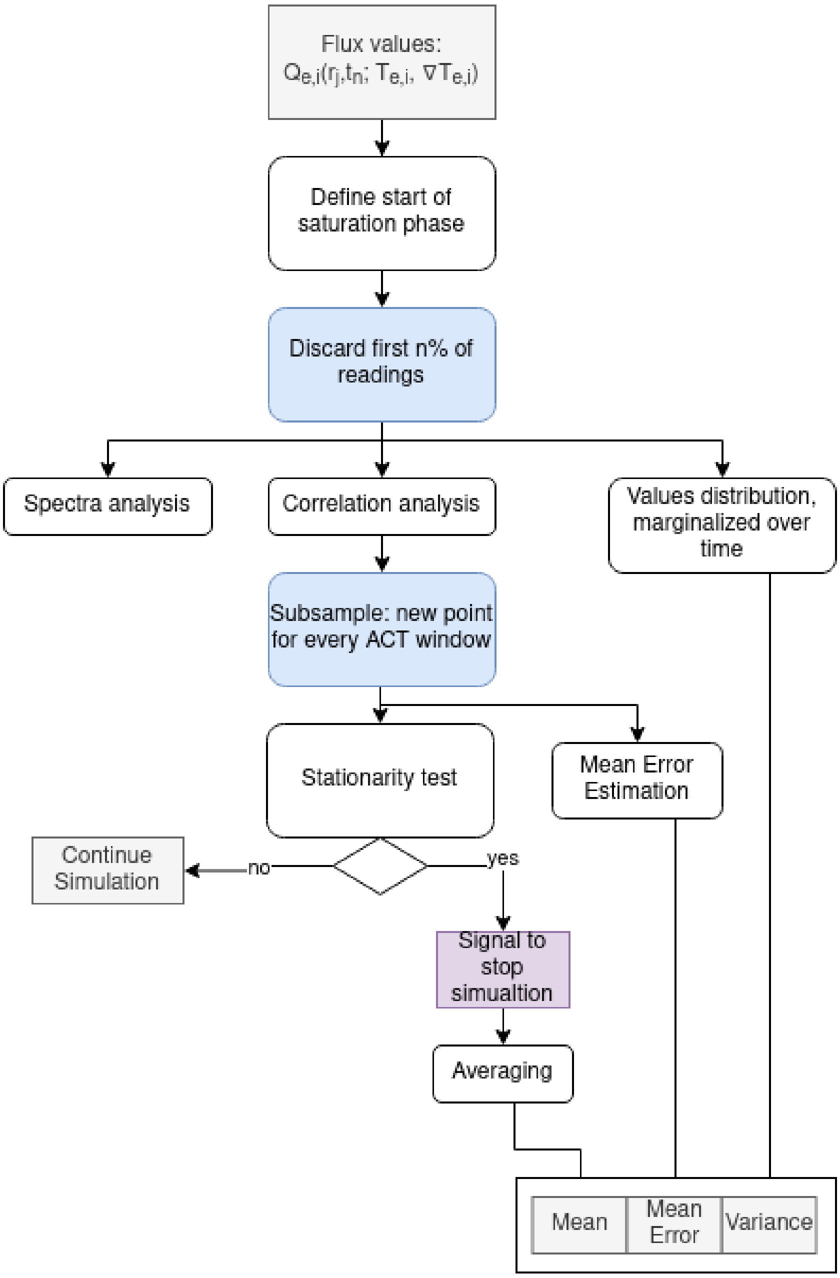
Flux values:
$Q_{e,i}(r_j,t_n; T_{e,i}, \nabla T_{e,i})$
Define start of saturation phase
Discard first n% of readings
Spectra analysis
Correlation analysis
Values distribution, marginalized over time
Subsample: new point for every ACT window
Stationarity test
Mean Error Estimation
Continue Simulation
no
yes
Signal to stop simualtion
Averaging
Mean
Mean Error
Variance

Figure 2.4.: A flowchart describing the steps to process the flux values time series computed by turbulence code at a particular radial location.

c) Test if the linear regression coefficients are below a chosen relative tolerance: $\hat{\boldsymbol{\theta}} < \boldsymbol{\epsilon}_{\text{tol}}$

   i. If the coefficients are too large, continue simulating another t_{run} time steps

   ii. If the coefficients are small enough, stop the simulation and proceed to the statistics calculation

4. Compute the essential statistics estimates:
   a) Mean: $\mu[y] = \frac{1}{n_{\text{eff}}} \sum_{y \in \mathbf{Y}^{\text{eff}}} y$
   b) Standard deviation: $\sigma[y] = \left(\frac{1}{n_{\text{eff}}-1} \sum_{y \in \mathbf{Y}^{\text{eff}}} (y - \mu[y])^2 \right)^{-\frac{1}{2}}$
   c) Standard Error of the Mean: $\text{SEM}[y] = \sigma[y] / \sqrt{n_{\text{eff}}}$

However, the particular threshold quantities for the convergence criteria require fine-tuning for particular simulation regimes, which in practice means producing several longer preliminary exploratory runs to decide on the convergence criteria. Furthermore, consideration of multiple QoIs with qualitatively similar behaviour within a single run requires to have a way of combining various criteria $\mathbf{C} = \{\text{crit}_i\}$. For that, one may employ the majority voting principle, the unanimous voting principle (logical `AND` for all criteria), or criteria for linear combinations of QoI or other functions of QoI.

The described procedure is applied to all GEM simulations in the study of its parametric dependency, which allows us to estimate the error of the effective transport level, quantify the model's aleatoric uncertainty, and decide on the sufficient duration of turbulence simulation.

2.2.3. Literature Review

The area of aleatoric uncertainty quantification is relatively new as a single concept. However, many problems it deals with are presented in other areas of computational science. These problems include considering systems exposing chaotic behaviour and fluctuating quantities of interest.

Specific properties of the studied system should support treating time traces of fluctuating quantities as stochastic processes. Some argue that such a description is suitable primarily for non-linear systems in a quasistationary state in the vicinity of an attractor and exposing ergodicity of the dynamics [18, 38].

One of the works considers an ensemble of replicated simulations for the same fixed values of independent parameters to compute purely aleatoric uncertainty in binding energies computed by molecular dynamics (MD) simulations [39]. Other works approach the aleatoric uncertainty in the context of filtering algorithms that split trend behaviour from the noisy dynamics [40].

The primary type of aleatoric uncertainty considered in this work is the uncertainty related to fluctuating QoIs in the solution of numerical models. Given the practicality of the analysis of chosen scalar quantities and the discreteness of numerical quantities in terms of time, this leads to a particular interest in studying time series. The consideration of uncertainties in the time series is usually done by describing such time series as a stochastic process, usually a stationary one, which raises questions about its quantification with properties of two kinds. One kind is the properties of a stochastic process from its theoretical perspective, which includes the estimation of its ACF, variance, and other moments. This type of quantity gives insight and characterisation of the properties of the mathematical model that produced the time series and the physics behind the described process. The other type is purely statistical errors in estimating the parameters of such stochastic properties. These quantities describe the effects of the finite sample size and give an idea of the numerical error such a model can produce.

The problem of characterising the time series in such a manner arises in multiple fields of applied Computational Science, including molecular dynamics, computational chemistry, computational fluid dynamics, climate modelling, and plasma physics. One of the works in plasma physics highlights the statistical understanding of the error estimates for mean values of fluctuating quantities and the importance of the dynamic control of the sample size to produce such time series [36]. Another work in Computational Fluid Dynamics (CFD) concerns the computational aspects of time series analysis of turbulent flow, especially the memory footprint of autocorrelation computation for a large number of DoFs for which the series is analysed, and suggests a way to perform such an analysis using temporary local updates [37]. In the area of molecular dynamics, the problem of finding the time point when the system has reached equilibrium so that simulation can be stopped arises, and one can tackle it via a range of approaches based on sequential analysis of standard error estimates [41].

2.3. Uncertainty Quantification. Forward problem

In this work, we perform uncertainty propagation, also called the solution of the forward uncertainty quantification problem, for a computational model describing turbulent transport in plasmas. Such a problem assumes that the model's independent variables are random and require quantifying properties of the model's quantities of interest as random variables. This means numerical characterisation of its distribution functions, which we perform by describing its statistics. The computation of statistics requires numerical integration over the distribution of independent variables, for which multiple methods were developed. This section presents spectral projection quadrature methods, namely the Polynomial Chaos Expansion (PCE), the collocation-based methods, and Monte Carlo (MC) numerical integration.

2.3.1. Polynomial Chaos Expansion

One of the most widely used and well-known methods for expressing uncertainties of a random function is *Polynomial Chaos Expansion (PCE)* [17, 16].

It is based on the idea that the PDF of the function value y can be expressed via an expansion using an orthonormal polynomial basis $P_i(x)$ each depending on the subset of function argument components

$$Y \approx \hat{Y}(X) = \sum_{i=0}^{N} c_i P_i(X) \tag{2.1}$$

where N is the highest order of polynomials, defining the truncation of the expansion, and c_i are the coefficients of the expansion.

For a given function $Y = f_X(x)$, the expansion coefficients c_i and the points to evaluate at x_k could be determined as a *quadrature* scheme based on a Spectral Projection method, exploiting the orthogonality of the basis polynomials and using normalisation factors H_i for polynomials P_i:

$$\begin{aligned} c_i &= \frac{1}{H_i}\int_\Omega Y(x)P_i(x)p(x)dx \approx \frac{1}{H_i}\sum_{k=1}^{N} Y(x_k)P_i(x_k)\omega_k \\ H_i &= \int_\Omega P_i^2(x)p(x)dx \end{aligned} \tag{2.2}$$

Quadrature scheme

For each of polynomial classes, for every degree p of a univariate polynomial $p_d(x)$ there is a set of $p+1$ points ξ_i values of a polynomial at which $p_d(\xi_i)$ fully define the polynomial.

In this case, we considering approximations of type $Q(f) = \sum_{i=1}^{N} w_i f(x_i)$.

An essential element of the expansion is the argument values to evaluate the function at, which are called *abscissas*, *nodes*, or *points* of the quadrature. The other element of such an expansion is multiplicative coefficients in front of the function values, which are called respectively quadrature *weights*.

In this work, we employ *Gaussian* quadrature that optimises the order of accuracy of the scheme and, having positive weights, is numerically stable. This quadrature uses weights in the form of values of polynomials, and using n abscissas can exactly express polynomials up to degree $2n-1$. This scheme uses zeros of the orthonormal basis polynomials as the nodes.

Basis choice

In order to implement the approximation optimally, in terms of its accuracy increment with an addition of a DoF, the basis of the expansion is chosen to be orthogonal. For the polynomials used in the Polynomial Chaos Expansion (PCE) class of expansions, one such possible basis is a family of polynomials to form such a Gaussian quadrature, described later in this section. They are orthonormal in the sense of a product in a weighted Hilbert space $\langle p_i(x)p_j(x)\rangle_p = \int_\Omega p_i(x)p_j(x)p(x)dx$.

Polynomial classes and Wiener-Askey scheme

For different types of distributions of independent variables, one should choose a corresponding quadrature scheme with the appropriate basis and quadrature nodes.

The Hilbert space to which the expansion belongs is chosen to be weighted with a PDF of the independent variables $p(\boldsymbol{x})$, chosen as inputs of the analysis, and serving as prior distributions in the Bayesian analysis of uncertainties. In this way, we are speaking about choosing an appropriate basis in $L^2(\mathbb{R},\gamma,\mathbb{R})$ where γ is chosen to be the PDF of the independent variable $\gamma = p_x(x)$ of the expansion of the function $f(x) : \mathbb{R} \to \mathbb{R}$. For example, for the normally distributed random variable $\xi : \xi \sim \gamma = \mathcal{N}(0,1)$, such basis is composed of *Hermite* polynomials $\mathrm{He}_n(\xi)$ orthonormal with respect to the product with the Normal measure $\int_\Xi \mathrm{He}_n(\xi)\mathrm{He}_m(\xi)d\gamma(\xi) = n!\delta_{mn}$.

Subsequently, the expansion has the form $f(\xi) = \sum_{n\in\mathbb{N}_0} u_n \mathrm{He}_n(\xi)$ with the coefficients defined by the projections on the chosen Hermite basis:

$$u_n = \frac{\langle f, \mathrm{He}_n\rangle_{L^2(\gamma)}}{||\mathrm{He}_n||^2_{L^2_\gamma}} = \int_{-\infty}^{\infty} f(\xi)\mathrm{He}_n(\xi)e^{-\xi^2/2}d\xi \tag{2.3}$$

Such a PC expansion leads to simple expressions for the expected value and variance of the value of random function $U = f(\xi)$; namely, it is 0^{th} term of the expansion, and the weighted sum of the squared expansion terms:

$$\begin{aligned}\mathbb{E}[U] &= u_0\\ \mathbb{V}[U] &= \sum_{n\in\mathbb{N}} n!u_n^2\end{aligned} \tag{2.4}$$

This leads to a choice of a particular polynomial class based on the input PDF with a mapping of these PDF on polynomial type known as *Wiener-Askey* scheme [42]. For every class of PDF of the independent uncertain input variable, there is a type of polynomial and respective quadrature suitable for such a polynomial expression of uncertainties in the function value. This correspondence is described in the table 2.1.

Table 2.1.: Wiener-Askey scheme table for correspondence of parameter PDF classes and polynomial quadrature schemes [42].

Polynomial	Support	Distribution Name	Density	Normalisation
Hermite	$(-\infty,\infty)$	Gaussian	e^{-x^2}	$\sqrt{\pi}2^n n!$
Laguerre	$[0,\infty]$	Gamma	$x^\alpha e^{-x}$	$\frac{\Gamma(n+\alpha++1)}{n!}$
Jacobi	$[a,b]$	Beta	$(1-x)^\alpha(1+x)^\beta$	$\frac{2^{\alpha+\beta+1\Gamma(n+\alpha+1)\Gamma(n+\beta+1)}}{n!(2n+\alpha+\beta+1)\Gamma(n+\alpha+\beta+1)}$
Legendre	$[a,b]$	Uniform	1	$\frac{2}{2n+1}$
Charlier	$\mathbb{N}_0$	Poisson	$e^{-\alpha}\alpha^x/x!$	$\alpha^{-n}n!$
Meixner	$\mathbb{N}_0$	Negative Binomial	$(\beta)_x c^x/x!$	$\frac{n!c^{-n}}{(\beta)_n(1-c)^\beta}$
Hahn	$\{0,1,\ldots,N\}$	Hypergeometric	$\frac{(\alpha+1)_x(\beta+1)_{N-x}}{x!(n-x)!}$	$\frac{-1^n(n+\alpha+\beta+1)_{N+1}(\beta+1)_n n!}{(2n+\alpha+\beta+1)(\alpha_1)_n(-N)_n N!}$
Krawtchouk	$\{0,1,\ldots,N\}$	Binomial	$\begin{pmatrix} n \\ x \end{pmatrix} p^x(1-p)^x$	$\left(\frac{1-p}{p}\right)^n \begin{pmatrix} N \\ n \end{pmatrix}$

Finite truncations

In practice, the number of components of the expansion, meaning the maximum degree of the polynomial used for the approximation, should be chosen as finite. For the application, where a range of different maximum degrees of the polynomials could be tested, one should truncate by an elbow-like change in the increment of the approximation.

The approximation error in terms of its $L_2(\gamma)$ norm for $U - U^k$ for the expansion $U \approx U^K = \sum_{k=0}^K u_k \mathrm{He}_k(\Xi)$ tends to zero in mean square $\lim_{k\to\infty}(U^k - U) = 0$.

Curse of dimensionality

Usually, such an expansion is used for a function of multiple variables; hence, polynomials are multivariate functions with different arguments contributing in different degrees.

The PCE expansion in the form presented earlier could be generalised for multivariate functions trivially if it is performed for independent, in the probabilistic sense, of its argument components. Such a modified expansion, often called *generalised Polynomial Chaos (gPC)*, uses a set of mutually orthogonal polynomials to span the product of underlying spaces $L^2(\Theta, \mu, \mathbb{R})$ for $\Theta = \bigotimes_{1 \leq i \leq d} \Theta_i$ and $\mu = \bigotimes_{1 \leq i \leq d} \mu_i$

Analysing the contribution of every component, out of d total argument components, and their combinations up to some total degree p would be respectively combinatorially complex in the number of required function evaluations $N = (d+p)!/(p! \cdot d!)$. Such phenomenon, colloquially known as *curse of dimensionality*, leads to prohibitively high computational costs for any parametric studies in large dimensionalities and leads to a need to drastically reduce the compute by some of the means, that include dimensionality reduction techniques or adaptive type of sampling.

There are multiple modifications that allow to partially alleviate the issue of the course of the dimensionality. Among those are *sparse* quadrature formulae like Smolyak sparse grids that use a set of nested $1D$ quadrature rules to construct a node and weight choice to use only a sparse subset of the basis functions [43, 44, 45]. The other methods that work around the high dimensionality are *Monte Carlo (MC)* [46] method based on statistics of random variable evaluations and the Law of Large Numbers (LLN), which will be discussed in a later subsection.

2.3.2. Stochastic Collocation

In some cases, finding an approximate (uncertain) solution for a system of equations is required, given a set of (parameter) points for which the equation should be satisfied exactly. Such points then are called *collocation* points, and the respective method is called *collocation* approach [47, 17]. It differs from the previously described method in that the points are pre-described; hence, the subset of the solution space that is used is also pre-described. Such a set of collocation points defines the low-dimensional object, usually polynomial, used to approximate the solution, and the dimensionality of this object is proportional to the number of collocation points.

Often, such an approximate polynomial has to interpolate the solution; hence, polynomials of the *Lagrange* family are frequently used. For a given set of points $\Theta = \{\theta_n\}$ such a polynomial will have the form of:

$$\tilde{U}(\theta) = \sum_{\theta_n \in \Theta} U(\theta_n) \prod_{\theta_k \in \Theta, \theta_k \neq \theta_n} \frac{\theta - \theta_k}{\theta_n - \theta_k} \tag{2.5}$$

Such an interpolation could be prone to multiple issues, including the *Runge phenomenon*, and still, the collocation method could benefit from control of choosing the particular collocation points like *Chebyshev nodes* [48].

2.3.3. Monte Carlo

A famous and widely-used family of methods for numerical integration is *Monte Carlo (MC)*. It is based on the (pseudo-) random approach and the formulation of the integration problem so that the sought quantity $f(X)$ is an estimate of an expected value of some random variable [46, 17]. Then, one could employ the Law of Large Numbers (LLN) to calculate it as $\mathbb{E}_{X\sim\mu}[f(X)] \approx S_n(f) = \frac{1}{n}\sum_{i=1}^{n} f(X^{(i)})$. With such an approach, the integration process is based on random sampling from a PDF μ.

One of the main advantages of this class of methods is that they do not suffer from the course of dimensionality. For most cases, the error convergence of MC methods does not depend on the dimensionality of the problem and has a relatively slow rate of $\epsilon \sim n_{\text{samples}}^{-1/2}$. This arises due to Chebyshev inequality relating the error of the mean estimate with its variance $P(|S_n - \mathbb{E}[f(X)]| \geq t) \leq \frac{\mathbb{V}[f(X)]}{nt^2}$.

However, the integration of multivariate functions $f : \mathbb{R}^k \to \mathbb{R}$ requires an independent variation of every component of the argument vector $\boldsymbol{x} \in \mathbb{R}$. Furthermore, some statistics, including Sobol indices, which describe variance fractions due to the impact of subsets of input components and which shall be introduced in a later subsection, require integration over non-hyperrectangular domains, which in principle could lead to combinatorial or exponential complexity with the dimensionality k for MC integration. Due to the partition of unity property, however, the calculation of Sobol indices (which also entails the calculation of total variance and the mean) can be done in linear in dimensionality k complexity with a practical algorithm of *Saltelli* sampling [49, 50] requiring only $n(k+2)$ function evaluations.

More advanced versions of MC algorithms utilize knowledge of the independent parameters of the model to re-weight the sampling probability accordingly. This includes a *CDF inversion* or *inverse transform* method, which produces samples as $x_i \sim \text{CDF}^{-1}(U[0,1])$ [51]. This requires Cumulative Distribution Function $\text{CDF}(x)$ to be well-behaved and accessible, and for its Radon-Nikodym derivative $\frac{d\mu}{d\nu}$ with respect to a standard distribution function ν, like a uniform distribution $U[0,1]$ to exist and have a closed expression.

For an untractable distribution μ, one can apply a *Markov Chain Monte Carlo (MCMC)* method based on constructing such a Markov Chain whose stationary distribution approximates the distribution μ. This is usually complicated to perform, especially for multivariate distributions of high dimensionality. One of the methods applied for such a problem is the *Metropolis-Hastings (MH)* algorithm based on the iterative proposal and rejection of samples [52].

Other important Monte Carlo methods in simulations are *Hamiltonian* MC (HMC) used to model conservative physical systems and *Multi-Level* MC that leverages the correspondence between the density μ and quality of the numerical resolution of a physical problem [53]. Combining the methods, like the Multi-Level approach and Markov Chain Monte Carlo (MCMC), is also possible [54].

The high computational cost of MC methods in terms of the number of function evaluations can be alleviated by the inclusion of more information, like the high-order derivative values, function evaluations of multiple fidelity levels, and assumptions on the geometry of the sampled PDF [55, 56].

Worth noting is a popular method of *Quasi-Monte Carlo (QMC)* used to improve the convergence rate of MC algorithms by applying pseudo-random sampling schemes utilising low-discrepancy number sequences [57].

2.3.4. Literature Review

In recent years, the increasing accessibility of computational resources has enabled the performance of uncertainty propagation and, more generally, the numerical analysis of epistemic uncertainties in practically relevant mathematical models in multiple applied computational sciences.

One area that faces multiple challenges similar to those of plasma turbulence transport is *MD* simulations, which also require HPC codes for model solutions, experience uncertainty of epistemic and aleatoric types, and are inherently multiscale. There are multiple works dedicated to UQ for MD simulations. One of the works addresses two issues of the field, namely the Bayesian approach to forward uncertainty problem and software automation of UQ methods [58].

The mathematical challenges of the robust and computationally effective UQ, data utilisation for surrogate, and engineering effective software are tackled by multiple research groups [59, 60, 61, 62, 63, 64, 65, 66].

There are multiple examples in the area of fusion research. One of the works in the field of material sciences applies PCE to recover the dependency of the sputter yield of iron and tungsten surfaces on the properties of a deuterium beam, which also allowed for global SA [67]. The UQ and SA were previously applied to high-fidelity plasma turbulence simulations focusing on sensitivity-adaptive sparse grids, significantly saving computational cost [68]. A uniform framework for analysing both data coming from experiment and simulation requires performing UQ for the latter. A paper suggests an approach for kinetic models to assimilate simulation data into a Bayesian statistics framework [69]. In a similar manner, another work utilised impurity transport simulation and experimental spectroscopy to infer plasma profiles [70].

The previous work on the UQ applied to the multiscale fusion plasma simulation workflow from this thesis is addressed in [1].

Moreover, the UQ allows for a rigorous statistical comparison of experimental and modelling results, leading to a more precise *validation* of numerical models. Several works were done in this area, including ones applied to fusion plasmas. A validation effort was performed in the context of a multi-component turbulent transport model in plasma in [2], as well as other models [71].

A statistical approach to measure uncertainties based on a finite number of function evaluations and data of specific parameterised PDF is *variational inference*. Methods of this class leverage the parameterisation and metrics in the space of PDFs, e.g., Kullback-Leibler divergence, as well as the ability to access information on the function derivatives to deduce the parameters values [72, 73].

2.4. Sensitivity Analysis

One of the fields that is often tightly connected to uncertainty quantification is *sensitivity analysis (SA)*. In general, this field covers algorithms to conclude how different factors influence solutions of mathematical models in terms of their relative importance. Usually, by factors, we mean values of concrete quantities, e.g. parameters in parametric PDEs. Often, this field's final object of study is some measures of such sensitivity and how to obtain their numerical values. These measures could be produced in different ways, for instance, based on derivatives with respect to the model's parameters, calculated numerically, or based on statistics of model solution QoI. Also, they could be local, defining the model's response to some small parameter perturbation, or global, capturing the model's behaviour for wide PDFs of the uncertain parameter. In this work, we mainly employ *Sobol indices* as a measure of sensitivity, a statistic based on the decomposition of the total model's solution QoI variance as a multivariate function, as it is done in the ANOVA [21, 74, 75]:

$$f(x_1, \ldots, x_n) = f_\varnothing + \sum_{i=1}^{n} f_{\{i\}}(x_i) + \sum_{1 \leq i < j \leq n} f_{\{i,j\}}(x_i, x_j) + \cdots + = \sum_{\mathcal{I} \subseteq \mathcal{N}} f_{\mathcal{I}}(x_{\mathcal{I}}) \tag{2.6}$$

This quantity denotes a partial variance explained by the variation of a subset of components of model input parameters. They are calculated as a variance $\mathbb{V}$ of an expected value $\mathbb{E}$ of function output Y conditioned on X_i a subset of input components $i \in \{i\} \subset \mathcal{I}$ and normalised by the total variance of the output. In this work, we consider the first-order Sobol index S_i for estimating the influence of a single parameter i, as well as higher-order Sobol indices $S_{\{i\}}$ to calculate the impact of interacting parameters in a set of $\{i\}$. Furthermore, we consider total S_i^{tot} indices, defining both influences of a single parameter and all its interactions, using a set $-i = \mathcal{I} \setminus i$ of all components excluding the one of interest i:

$$S_{\{i\}} = \frac{\mathbb{V}[\mathbb{E}[Y|X_{\{i\}}]]}{\mathbb{V}[Y]}, \quad S_i^{\text{tot}} = 1 - \frac{\mathbb{V}[\mathbb{E}[Y|X_{-i}]]}{\mathbb{V}[Y]} \tag{2.7}$$

The Sobol indices of all orders until the total order sum up to one $\sum_{I \in 2^{\mathcal{N}}} S_{\mathcal{I}} = 1$ and serve as a partition of unity. The value of the Sobol index indicates how strongly this input parameter, or combination of parameters, influences the variation of the chosen QoI.

2.4.1. Literature Review

One of the issues often omitted in practical applications is that assumptions of probabilistic independence of the input parameters are usually not valid. In situations when inputs are themselves functions of some arguments, especially in cases when collocation methods are applied to existing data with known correlations, one has to apply an appropriate transformation, e.g., Rosenblatt one, to account for input correlations [76].

The Sobol indices are not the only metrics used in the ANOVA methodology. Another type of indices measuring the importance of dependent high-dimensional input are non-negative *Shapley values* [77].

A number of other works touch multiple trends in SA and aim method comparison [78, 79, 80].

2.5. Surrogate Modelling

In this work, we utilised a surrogate modelling approach to design several algorithms to quantify uncertainties in Plasma Physics simulations. This approach is based on using simulation data to regress over a sample of argument-value pairs to create a fast proxy function that can be used in place of an expensive to-evaluate function for an algorithm that requires a large number of function evaluations. The schematic depiction of the surrogate approach for uncertainty propagation is shown in figure 2.5.

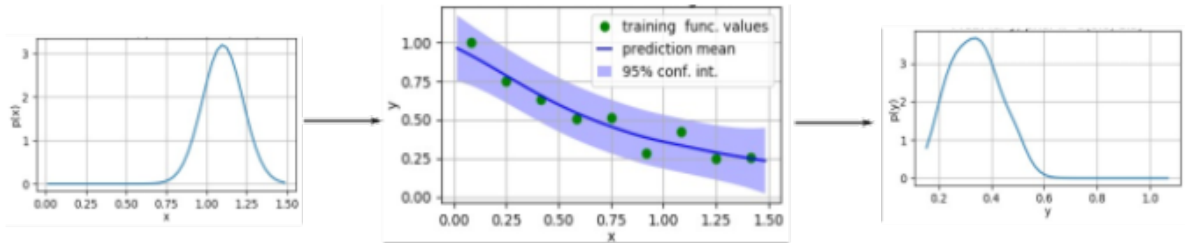

Figure 2.5.: A cartoon of uncertainty propagation using a surrogate model. A data sample is used to create a regression likelihood model $p(y|x)$ for the studied dependency, which could be used, together with a prior argument distribution $p(x)$, to estimate the distribution of function values $p(y)$.

This method is usually applied to a class of *multi-query* problems, which include uncertainty quantification, variational influence, and optimisation using simulation-in-the-loop. Often, these methods would like to use high-fidelity simulations to evaluate function values studied in a problem, and they require large numbers of simulations, which is, in practice, not feasible even when using modern supercomputers. Then, one can use a finite sample of function evaluations to approximate the studied dependency over some parametric region of interest [81]. This approach requires the preparation of a data sample of simulations and the use of data-based regression models $D = \{(\boldsymbol{x}_i, y_i)\}_{i=1..n_s} \rightarrow y = f_{\text{surr}}(\boldsymbol{x}|D)$. For complex multivariate dependencies requiring large amounts of data, these methods are usually based on ML.

An important aspect of surrogate simulation methods is that data acquisition is a very expensive process compared to other areas using statistical regression. This raises a question of the effective utilisation of existing data and the effective design of numerical experiments to sample the data required to create an efficient surrogate for solving the studied problem. This aspect led to the development of various *Active Learning (AL)* approaches for surrogate models, discussed in later sections.

2.5.1. Literature Review

The surrogate modelling approach effectively solves problems requiring extreme computational costs to resolve processes on extremely disparate scales. It was applied in multiple areas, starting from its pioneering field of geosciences and continuing with climate science, plasma physics, molecular dynamics, and other computational areas of mathematics, physics, and engineering [82].

The surrogate approach gains particular traction in fusion research. In the area of plasma turbulence, it was applied as a surrogate for a high-fidelity code for micro-tearing modes [83]. Furthermore, the Gaussian Process Regression surrogates for nonlinear gyrokinetic turbulence codes were used to predict kinetic profiles for new potential magnetic confinement devices via a Bayesian Optimisation targeting flux matching [84]. In the area of MHD, the Artificial Neural Network (ANN) surrogates were utilised to infer equilibria for a stellarator [85].

One of the examples resolves the properties of active materials utilising the simulations down to the molecular scales and dynamically constructing surrogates to resolve energetic QoI at the microscopic scales [86]. The problem of a dynamical update of a surrogate and interaction of its inferred results with high-fidelity simulations during time-integration is approached in the work suggesting an online neural network surrogate learning to reduce instabilities in climate models [87]. Another work approaches the issue of the data amounts produced by high-fidelity fluid simulations and the capturing of the correct information required by an effective surrogate from this data via reduced spectral models [88].

2.5.2. Machine Learning Aspects

Since the surrogate method, contrary to *Reduced Order Models (ROM)* method, assumes a data-driven approach, the problem of creation of such a surrogate is usually formulated as a regression problem, meaning a process of finding a mathematical expression for a particular dependency given a sample of argument-value pairs describing this dependency.

One of the inherent issues for the regression algorithms applied for non-linear dependencies is a poor ability to extrapolate. In a probabilistic context, it is often defined as an issue of a *distribution shift* between the PDF of data used to train a regression model and the one PDF of the new data to which a regression model is applied. Due to such an issue, one of the critical diagnostics of the applicability of any surrogate model is to check whether it is used in its interpolation regime. In ML, this is also known as *Out-of-Distribution (OOD) Detection* [89].

For this, in this work, we applied a wide range of checks on whether data for which a surrogate model infers certain QoI adhere to the training sample density and whether the surrogate is well applicable for it.

One such check uses explicit knowledge of the training sample distribution for every new input value. The other check utilises the properties of a Gaussian Process Regression model to have high posterior prediction uncertainties for data far away from the training sample.

Producing a review of the vast successes and advances of Machine Learning is a highly challenging task, even if one is up to limiting oneself to the areas of physics and computational sciences. A recent overview of Machine Learning applications in nuclear fusion research mentions various applications of ML both for processing experimental data and for decreasing the computational cost of fusion plasma modelling, as well as the usage of ML models in operating fusion devices [90].

2.6. Gaussian Process Regression

One of the classes of regression models widely used in Bayesian statistics and related methods applied to various fields of engineering and science is *Gaussian Process Regression (GPR)*. It is also known as *Kriging* after one of the first applied uses by Krige [91].

2.6.1. GPR Theory

This method is non-parametric, which here means that its functioning, hence inference, depends on an entire data sample of some observable dependencies. In the context of applying such a regression model, such a data sample would consist of pairs of numerical model input parameters, usually forming a vector of real numbers and corresponding values of the solution QoI, also forming a vector of real numbers [20].

Covariance Matrix and Kernels

The main ingredient of the regression model is the *covariance matrix* K describing how the outcomes are correlated with respect to change in the input variables. Besides the dataset of input values, the covariance matrix is defined by the covariance structure in the input parametric space, usually characterised by the *covariance function* $k(\cdot, \cdot)$ defined for all pairs of values in the input space.

Covariance functions, also known as *kernels*, also encode information on the function that is being modelled. The kernel is chosen to be a *symmetric positive semi-definite (s.p.s.d.)* function $\int k(x,x')f(x)f(x')d\mu(x)d\mu(x') \geq 0, \forall f \in L_2(\chi, \mu)$.

By Mercer's theorem, the s.p.s.d kernel can be represented as an eigendecomposition of form $k(x,x') = \sum_{i=1}^{\infty} \lambda_i \phi_i(x) \phi_i^*(x')$ where λ_i are the eigenvalues and $\phi_i(x)$ are normalised eigenfunctions [92]. Such a representation builds a correspondence between the representation of the process in a high (infinitely) dimensional feature space and the representation in terms of covariance function, which is known as *kernel trick* and allows for efficient computation of kernel values.

In the majority of cases, the random process describing the sought dependency is assumed to be *stationary*, hence depending only on the metrics in the input parameter space $k(x_i, x_j) = k(||x_i - x_j||) = k(r)$.

Bochner's theorem allows expressing a stationary kernel as a function of distance via its eigendecomposition $k(x,x') = k(x-x') = k(\tau) = \int_{\mathbb{R}^D} e^{-i2\pi\langle s,\tau \rangle} d\mu(s)$ for a finite positive measure μ [93]. This is analogous to the Fourier decomposition of the measure in terms of kernels.

One of the most commonly used kernels is *Squared Exponential (SE)*:

$$k^{\mathrm{SE}}(x_i, x_j) = \sigma \exp\left(-\frac{1}{2}\frac{|x_i - x_j|^2}{\lambda}\right) \tag{2.8}$$

also known as *Radial Basis Function (RBF)* or *Gaussian* kernel.

The other widely applied covariance functions belong to the *Matérn* class, expressed by

$$k_\nu^{\mathrm{Matern}}(r) = \frac{2^{1-\nu}}{\Gamma(\nu)}\left(\frac{\sqrt{2\nu}r}{\lambda}\right)^\nu K_\nu\left(\frac{\sqrt{2\nu}r}{\lambda}\right) \tag{2.9}$$

where K_ν is a modified Bessel function. This covariance function has an additional degrees-of-freedom parameter ν which in infinity limit would give SE kernel $\lim_{\nu\to\infty} k_\nu^{\mathrm{Matern}}(r) = e^{-r^2/\lambda^2}$. This expression could be simplified for half-integer values $\nu = p + \frac{1}{2}$ and in this work, in some cases, a rough and quickly changing Mateŕn-3/2 kernel $k(r) = \sigma^2\left(1 + \sqrt{3}\frac{r}{l}\right)\exp\left(-\sqrt{3}\frac{r}{l}\right)$ was used.

In principle, one can generate new kernels as linear and multiplicative combinations of existing kernels, as operations of addition and multiplication do not break symmetricity and positive semi-definiteness.

The choice of the kernel to describe a Gaussian Process corresponds to a choice of a basis for a *reproducible kernel Hilbert space (RKHS)* defining the class of functions that the process can approximate. This correspondence allows us to interpret a choice of a kernel for GPR as a choice of prior over the functional space and the training with new observable data as Bayesian update of the posterior over the functions in the selected RKHS [20].

The covariance matrix of a stationary process given its covariance function would be simply defined as having elements equal to the value of the covariance function for every pair of observations $K = \left(k(x_i, x_j)\right)_{i,j=1..N} \in \mathbb{R}^{N\times N}$.

Such a covariance function should have ones on the main diagonal due to the limitations of kernel functions in having a unit value for zero argument, meaning that observations with zero distance apart are equivalent and perfectly correlated. Being constructed out of values of s.p.s.d functions, the covariance matrix is also positive semi-definite $xKx^\top \geq 0, \forall x \in \mathbb{R}^n$. Often, one would like to consider the noise of the observations or, in the context of uncertainties in mathematical models, an aleatoric irreducible uncertainty of certain QoI. In this situation, one would add a diagonal matrix with σ_n free parameter of noise intensity to the covariance matrix $K_y = K + \sigma_n^2 I$ to capture such a noise.

Given the covariance function, the inference of the model, the posterior probability for the values of the process for an arbitrary new observable is Normal, with the following mean and variance expressions

$$\begin{aligned}
&p(f(x_*)|\mathbf{X}, \mathbf{y}, x_*) \sim \mathcal{N}(\mu(x_*), \sigma^2(x_*))\\
&\mu(x_*) = K(x_*, \mathbf{X})^\top \left(K(\mathbf{X}, \mathbf{X}) + \sigma_n^2(I)\right)^{-1}\boldsymbol{y}\\
&\sigma^2(x_*) = K(x_*, x_*) - K(x_*, \mathbf{X})^\top \left(K(\mathbf{X}, \mathbf{X}) + \sigma_n^2(I)\right)^{-1} K(x_*, \mathbf{X})
\end{aligned} \tag{2.10}$$

The particular choice of the kernel to describe the dependency is generally a complicated question, and the kernel type is usually treated as a categorical hyperparameter during the model fitting.

Model Fitting

In principle, once we express the inverse of the covariance matrix, the inference process is based on matrix-vector multiplication. It has $\sim O(N^3)$ complexity, where N is the sample size describing the Gaussian Process. For a given regression model, one would usually express the inverse of the covariance matrix via *Cholesky decomposition* $K = LL^\top$, which is of $O(N^3)$ complexity with the size of the matrix [20]. In numerical linear algebra, it is one of the most convenient methods to solve systems of linear equations for symmetric positive definite matrices, like a covariance matrix, via the decomposition of a matrix into a product of two lower-triangular matrices, which can be efficiently done with direct methods. For large covariance matrices describing large samples of data, the computational cost of the decomposition could be reduced by a multitude of methods. For example, by utilising the fact that covariance is a quickly decaying function and by applying a cut-off, a covariance matrix can be transformed into a block or a band matrix [20].

In order to perform the regression, one would like to find such parameters $\boldsymbol{\theta}$ of the covariance matrix, hence the covariance function, which would lead to the best explanation of the observed training data sample. These parameters may include all the free parameters of the compound kernel, like its lengthscale λ and amplitude σ_f, as well as the noise level σ_n. In practice, there are several ways to do it. In this work, we employ the *Maximum Likelihood Estimation (MLE)* method of Bayesian model selection, meaning that we find parameter values that would maximise the regression model's likelihood of explaining the data. For a GPR model with Normal prior, that would mean that we would like to optimise $\log p(\boldsymbol{y}|X,\boldsymbol{\theta}) = -\frac{1}{2}\boldsymbol{y}^\top K_y^{-1}\boldsymbol{y} - \frac{1}{2}\log|K_y| - \frac{n}{2}\log 2\pi$.

This optimisation is usually performed via some gradient-based method like LBFGS where the computation of the derivative $\frac{\partial}{\partial\theta_j}\log p(\boldsymbol{y}|X,\boldsymbol{\theta}) = \frac{1}{2}\mathrm{tr}\left((K^{-1}\boldsymbol{y}\boldsymbol{y}^\top K^{-1^\top} - K^{-1})\frac{\partial K}{\partial\theta_j}\right)$ takes $O(N^2)$ operations in sample size.

Furthermore, other methods may imply that the parameter $\boldsymbol{\theta}$ itself is a random variable, and we would like to optimise the Marginal Likelihood over the entire $p(\boldsymbol{\theta})$. Such a fully Bayesian approach for model parameter selection requires integration over complex intractable functions, typically done numerically with an *MCMC* approach.

2.6.2. Modifications

The general concept behind GPR is not limited to the processes with particular Gaussian likelihood. A similar approach could be applied to model data as realisations of different stochastic processes.

This work included several tests on regression for *Student-t* processes. Student-t distribution has a degree-of-freedom parameter ν, which would yield a Normal distribution in infinity limit $\lim_{\nu\to\infty}\mathcal{STP}(\mu(x),k(x,x'),\nu) = \mathcal{GP}(\mu(x),k(x,x'))$. The update of GPR to Student-t processes requires a change of likelihood in the fitting and modifying covariance matrix update expression to accommodate additional DoFs [94]. The application of Student-t processes allows for accommodating heavier tails of data distribution and makes the regression significantly more robust to outliers [95, 96].

2.6.3. Literature Review

The GPR is a type of model widely used for statistical regression and as a surrogate for expensive models in science and engineering. The last decades have led to many areas where it is applied and many methods to improve the regression and adapt to particular situations.

Being a stationary process, it is difficult for a GPR to model a parametric dependency of noise, i.e., to model *heteroscedastic* process. A number of methods allow for GPR heteroscedasticity, including a replication approach using multiple outputs for a single input value [97].

One of the ideas that increases the efficiency of GPR is adding information on the derivatives of the dependency into the training data and process, which allows having a data-based surrogate for PDE solvers in an efficient way [98, 99].

GPR, as a non-parametric method, is equal to a *Bayesian Neural Network (BNN)* in the limit of an infinitely large layer. The combination of ideas of composition of GPRs and matching kernels to activation functions lead to the method of Deep Gaussian Processes [100].

Furthermore, in fusion research, GPR is used for a variety of purposes, including modelling and regressing over experimental measurements of tokamak profiles [101].

2.7. Artificial Neural Networks

The other type of Machine Learning model employed in this work was *Artificial Neural Network (ANN)*. The core idea of this type of model is that a particular dependency is described as a composition of a number of parametrised multivariate functions, often nonlinear $y = f_n \circ \cdots \circ f_1(x)$. Each elementary transformation $f_i(\cdot)$ is called a *layer*, having n of those in the aforementioned expression. It usually works as a combination of matrix multiplication and a non-linear transformation $f_i(x) = \mathrm{NL}_i(W_i \cdot x + b_i)$ where W is a matrix of free parameters called *weights*, b is *intercept* or *bias* allowing for dependencies not-aligned with the origin, and NL is non-linear *activation function*, e.g., a ReLU function.

For complex high-dimensional dependencies, one should use a large number of layers. Model of such type is called a *Deep Neural Network (DNN)* and the respective field is called *Deep Learning (DL)*.

The particular set of parameter values needed to express the dependency is found through optimisation of a certain *loss* or *risk* function $\theta^* = \operatorname{argmin} L(X, y, \theta)$ that usually expresses how well the model captures the studied dependency given a sample of arguments and values of this dependency, known as the training data set. In the case of regression, it can be an L_2 norm of the error with respect to the training data sample, and for the classification task, it can be a cross-entropy measure. Such an optimisation procedure is enabled via the backpropagation of the loss gradients $\nabla_\theta L$ with respect to the model parameters and implemented via *automatic differentiation*. From the optimisation theory point of view, the search of optimal parameter values is performed via gradient descent-based methods, like the popular ADAM algorithm [102], which employs multiple features due to several common issues inherent to the usual form of the loss function in ML, like its high dimensionality, non-convexity, and poor regularity, including the presence of multiple local optima. These features helping with the optimisation procedure include stochastisation, i.e., using a subsample for performing an optimisation (to speed up high-dimensional search), regularisation of the loss function (like a family additive Tikhonov regularisation, which targets better generalisation properties of the model), Nesterov moment based algorithm (to avoid local extrema).

2.7.1. Literature Review

Machine Learning is a field that gets particular traction in all areas of science, not excluding fusion research. Being the most widely used ML model for classification, regression, and sampling (generation), they were studied as a methodology in depth in multiple reviews, including ones in [103, 22, 90]. An example of classification is using experimental data from various fusion devices to define whether a particular experiment will experience a disruption or to define confinement types [104, 105, 106].

Physics Informed Neural Networks (PINNs), an architecture of ANNs that gained a particular traction as a surrogate for a surrogate of physical systems. Their base idea is to utilise the structure of the PDE structure to solve, as well as the information on the gradient values of the employed differential operators and the initial and boundary conditions applied, to explicitly formulate residual loss so that ANN training can solve the PDE [107]. This architecture was applied for multiple problems of fusion research, for example, for calculating heat fluxes on the device wall [108].

Furthermore, physics-aware ANN models were applied to predictor-corrector numerical schemes to upscale simulation of plasma fluid Hasegawa-Wakatani turbulence simulations [109], as well as to direct numerical simulations of NS fluid turbulence [110]. A vast amount of plasma turbulence simulations data were utilised to create fast ANN surrogates for turbulent transport inference [111].

2.8. Active Learning

Since one of the critical aspects of creating and applying a Machine Learning model or any type of data-based regression or classification model is data that is used for creating and training such a model, fields or applications without an abundance of pre-existing data have to find workarounds for the lack of data. In particular, this applies to the areas where data samples have to be gathered from scratch, and acquisition of such new data is especially costly and associated with performing costly real-life or experimental measurements or, in more theoretical fields, performing computationally expensive simulations to solve particular mathematical equations for a specific model. Often, in such situations, one would like to apply one of the *AL* approaches, where information about the performance of the model using existent data is utilised to define how to acquire new data, for example, what are the important or interesting independent variables values of experiment or simulations for which we would like to know some QoI values [112]. These methods are often implemented in the venue of the sequential design of experiments, and one of the approaches how to design a particular sequential sampling scheme is to apply *Bayesian Optimisation* algorithms.

2.8.1. Bayesian Optimisation

Bayesian Optimisation (BO) usually deals with optimisation of a specific function $f(\mathbf{x})$ over support $\mathbf{X}$ under uncertainty, which is specified in terms of a conditional PDF $p_f(\mathbf{x})$ defined for function values over the studied domain or parametrisation of this PDF. This conditional PDF could be understood as a likelihood for functional values given a certain observed value of a function argument. One of the straightforward approaches is to assume this likelihood to be Normal. In such a case, this likelihood could be provided by a Gaussian Process Regression performed for a finite sample of pairs of dependent and independent variables.

Given such a likelihood model, BO algorithms leverage the additional information to find the extrema of a function by performing optimisation on a proxy function, helping to define new points in the support, called acquisition function.

BO is a class of gradient-free methods requiring only a finite sample of function evaluations and a regression model over it, with a gradient defined by this regression model. Every next iteration of the optimisation procedure assumes one or more function evaluations, which could be added to the existing pairs sample and followed by an update of the regression model to include new data.

A result of a GPR surrogate training for a noisy GEM0 model, introduced in the later chapter 3.4, as a univariate function for a visual demonstration is shown in figure 2.6.

Here we used the global variance as the acquisition function $f_{\text{acq}}(\boldsymbol{x}) = \mathbb{V}_y(\boldsymbol{x})$ which here, without any additional information about the target function, yields a uniform coverage of parameter space and reduces a compound error over its chosen region of interest [113].

2.8.2. Literature Review

The field of BO applied to Active Learning of surrogates for computationally expensive surrogates reached significant successes in various fields in recent years. One of the first works on AL for data-based models was performed to minimise the variances of the neural networks [114]. One of the works suggests domain-adaptive GPR for CFD problem [115]. The studies on different types of acquisition functions for BO were also performed [113].

In the area of fusion research, the BO using GPR was applied to a number of problems, including predicting kinetic profiles consistent with turbulent transport [116], modelling and calibrating parameters for runaway electron simulations [117], and for inertial fusion confinement research [118]. Another case on the intersection of plasma and molecular dynamics applies an ensemble of ANNs instead of GPRs to produce predictions of transport

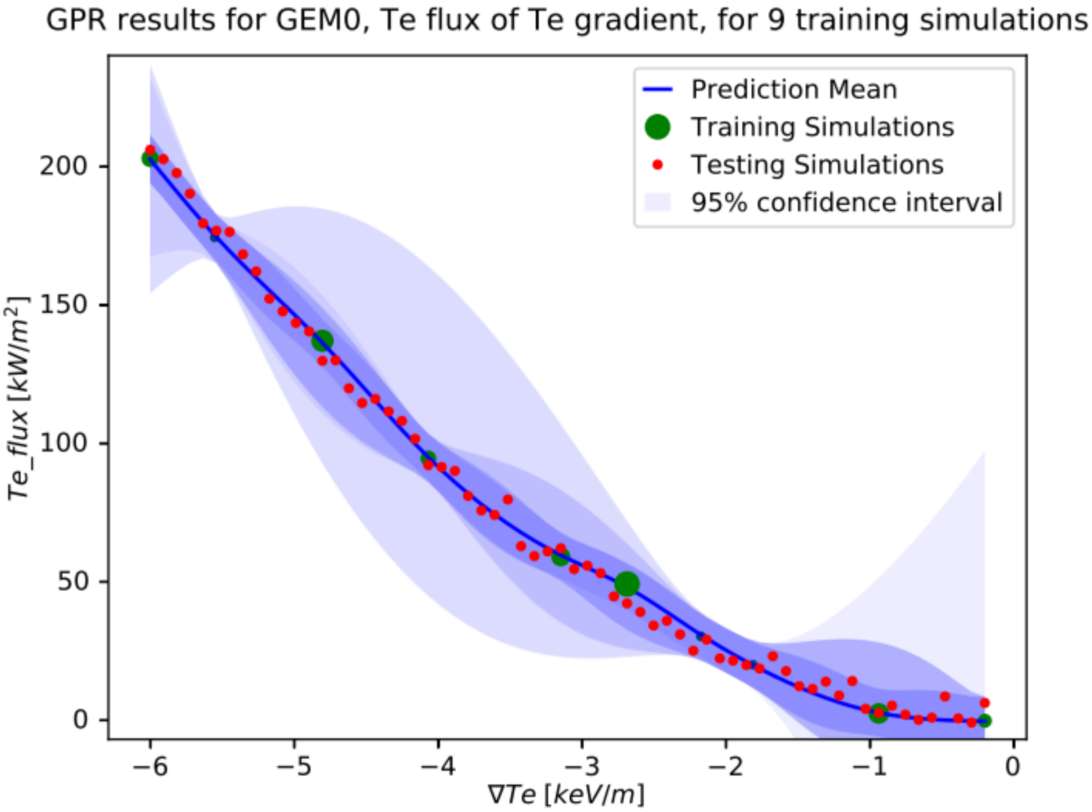

Figure 2.6.: A result of AL for a GEM0 GPR surrogate model via BO using maximum variance as an acquisition function. The learning is performed in a $\mathcal{O}(10)$ number iterations for a univariate dependency of ion heat flux on ion temperature surrogate. The larger green markers indicate newer function evaluations and the darker shaded area indicates the reducing (epistemic) uncertainty of the surrogate model.

coefficients with uncertainties, actively learn ANN surrogates, and use them for multiscale modelling [119].

2.9. Software and HPC Aspects

All the elements described in this work required the implementation of computational procedures to analyse uncertainties, including running and managing multiple (10^{1-3}) instances of simulations using High-Performance-Computing (HPC) machines.

2.9.1. HPC

Most of the presented work was done using Cobra, a supercomputer of Max-Planck Computing and Data Facility (MPCDF) [120].

MPCDF Cobra

Operational from the spring of 2018, Cobra is a supercomputer managed by MPCDF and equipped with $136,960$ CPU cores. The CPU partition is organised into nodes, each consisting of two Intel Xeon Gold 6148 processors of 20 2.4 GHz cores of the Skylake family. The nodes are connected with 100 Gb/s OmniPath interconnect and are organised into islands, 636 nodes each, which is the limitation for a single batch job submission. The supercomputer manages the jobs submitted by users with SLURM workload manager [121].

2.9.2. Pilot Job Mechanism

Multi-query problems, like uncertainty quantification and multi-component computational workflows, require the execution of multiple jobs that can be highly inhomogeneous in their dependencies and resource requirements. Conventional workload managers are mostly designed to operate with large jobs or groups of highly similar jobs, using resources in a homogeneous way. The need to deal with the inhomogeneity led to the rise of elements of *High-Throughput-Computing (HTC)* in the domain of HPC. One such hybrid approach to deal with parallel computational jobs, used for most of the workflows presented in this work, is *pilot job* mechanism. It provides a program abstraction and the related middleware implementation to redistribute resources among multiple jobs of various types from what a supercomputer considers a single large job submission. In this work, we use QCG-PilotJob [122] for such a mechanism to run the Multiscale Fusion Workflow (MFW), introduced in the later chapter 3, which solves the physical problem, to manage multiple code instances while propagating uncertainties, and to manage various surrogate instances during their hyperparameter tuning.

2.10. Chapter Summary

This chapter presented an algorithm to measure aleatoric uncertainties if time traces of fluctuating quantities in solutions of computational models that allow regulation of the simulation duration and accuracy of statistical estimates, as well as the underlying theory. It gave an overview of methods for parametric uncertainty propagation, namely Polynomial Chaos Expansion, Stochastic Collocation, and Monte Carlo integration. Furthermore, it discusses the surrogate approach to solving problems requiring large numbers of simulations, focusing on Gaussian Process Regression as a surrogate model of choice and its Active Learning method. Finally, it describes some aspects of applied computer science of the presented algorithms.

3. Computational Model

To study the effects of turbulent processes happening on the scales of microseconds in time and millimeters in space on the temperature and density profiles of a nuclear fusion device across the confinement scales of meters and seconds, we solve non-stationary transport equations inferring its anomalous transport properties from solving local gyrofluid equations.

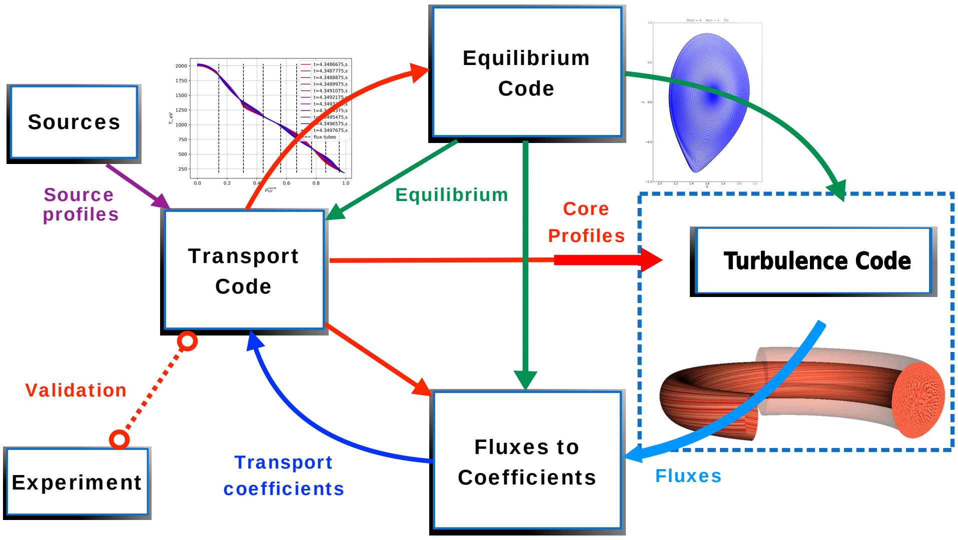

Figure 3.1.: A schematic view of Multiscale Fusion Workflow. The rectangles denote codes, each solving a system of equations, and the arrows denote the information exchange on partial solutions. The red denotes the flow of information about core profiles, the green is for MHD equilibrium, and the blue is for turbulence. Some workflow parameters, such as power and particle sources, must be specified from external information. One of the usage scenarios is model validation against the experimental data.

The software implementation of a solution for such a system of equations used in this work is the *Multiscale Fusion Workflow (MFW)* [123, 124, 23]. It consists of three main models, each implemented as a separate computer code, independently solving its subset of studied equations. Each code serves as an independent component, being a black box to others but capable of exchanging information on the partial solution with other components.

The codes, as shown in figure 3.1, are the equilibrium code that describes plasma geometry, the transport code that evolves the transport profiles, and the turbulence code that is able to compute effective transport fluxes.

Each code reads and writes information about external and computed physical quantities into a standardised data structure called *Consistent Physical Objects (CPOs)* [125]. These data structures consist of a hierarchy of dictionaries, each containing scalar $0D$ quantities values or $1D$ or $2D$ arrays for spatially resolved quantities. Several types of CPO data structures are employed in this work, each explaining its aspect of a plasma state or plasma evolution. In this work, we use `coreprof` CPOs for a description of plasma core kinetic profiles, `equilibrium` CPOs to describe MHD equilibrium quantities and magnetic field geometry, and `coretransp` CPOs to represent transport properties, including fluxes and transport coefficients. Each CPO object can be stored as a separate `ascii` file and serve as an input for a standalone code or as a unit of data exchange between workflow components in binary format.

The codes exchange information in the form of CPOs in a point-to-point fashion using the MUSCLE3 coupling library [28]. To manage computational resources required by heterogeneous code, this library uses QCG-PilotJob middleware [122].

During the work, the workflow implementation was updated from version using MUSCLE2 [24] to MUSCLE3 library, which allowed reducing dependencies on an additional level of wrapping implemented in Ruby language and allowed the introduction of new features.

Such a modular workflow implementation allows for easy choice of different transport parametrisation, meaning here: models for computing anomalous turbulent transport coefficients and implementations of their solvers.

3.1. Multiscale Fusion Workflow Components

In this section, we will give an overview of the main components of the Multiscale Fusion Workflow that allow for a self-consistent solution of the non-stationary heat and particle transport equation considering turbulent anomalous transport properties.

3.1.1. Transport Code: ETS

From the point of view of the largest scale, and the point of view of integrating information from different sources, the main component of the workflow is the transport code. This work uses the European Transport Solver (ETS) [126, 26] as the transport model. This code uses mean field approximation of 3D conservation laws and solves 1D equations for quantities as functions of time and a monotonic radial coordinate describing distance from the core axis of a toroidal magnetic device. As such a radial coordinate, this work uses $\rho_{\mathrm{tor}}^{\mathrm{norm}} = \rho_{\mathrm{tor}} / \max\left(\rho_{\mathrm{tor}}\right)$ where, $\rho_{\mathrm{tor}} = \sqrt{\frac{\phi}{\pi B_0}}$ is a function of toroidal magnetic flux ϕ and axial magnetic field B_0. For this coordinate, we can transform the volume V enclosed by a flux surface on this radial coordinate.

The exemplary mean filed heat conservation equation governing the change of ion temperature over time and radius are similar to those in ASTRA [27] and has the form of:

$$\frac{3}{2}\left(\frac{\partial}{\partial t} - \frac{\dot{B}_0}{2B_0} \cdot \frac{\partial}{\partial \rho}\rho\right)\left(n_i T_i V'^{\frac{5}{3}}\right) + V'^{\frac{2}{3}} \frac{\partial}{\partial \rho}\left(q_i + T_i \gamma_i\right) = V'^{\frac{5}{3}} \left[S_i^{\mathrm{exp}} - S_i^{\mathrm{imp}} \cdot T_i + Q_{ie} + Q_{\gamma i}\right] \quad (3.1)$$

where $\frac{\partial}{\partial t}$ is time derivative, $\frac{\dot{B}_0}{2B_0} \cdot \frac{\partial}{\partial \rho}\rho$ is adiabatic compression term, V' is volume derivative, q_i is ion heat flux, S_i^{exp} and S_i^{imp} are explicit and implicit heat sources, and Q_{ie} and $Q_{\gamma i}$ are exchange terms [127]. The explicit terms should be provided by the external description of sources.

Furthermore, the conversion from magnetic flux surface average quantities into SI quantities is needed. This requires using elements of the metric tensor to convert values into units of meters, meaning that here we need only the diagonal elements $g^{\rho\rho}$ representing spatial scaling. The scaling metric element factor $\sqrt{g^{\rho\rho}}$ here is computed via the framework of the equilibrium code described in the following subsection 3.1.2.

In this work, for the purpose of having a smaller effective problem dimensionality and in order to analyse key concepts, we mainly focus on the heat transport inside the core of a tokamak.

In general, information that is sufficient to describe transport is averaged over time and integrated over flux surface fluctuation-based transport flux, mainly considering particle flux for different species (here we consider only deuterium ions and electrons) Γ_i, Γ_e and associated with them heat flux Q_i, Q_e.

However, since the transport code solves the equations using the effective diffusivity D and convection velocity v_{conv}, calculation of these transport coefficients and splitting the flux values into the respective diffusive and convective terms is a necessary step. The presence of different transport channels, however, introduces an additional degree of freedom into the model and requires some analytical and practical considerations on how to resolve it. Here, we use an auxiliary code IMP4DV to recompute D and v based on the given heat fluxes [128].

A transport simulation is performed on a characteristic time scale of energy confinement time, for AUG considered $\tau_{\mathrm{conf}} \approx 0.3s$. A single time step of kinetic profile evolution is set to $\Delta t_{\mathrm{ETS}} = \Delta t_{\mathrm{MFW}} = 10^{-3}s$. However, the ETS time step can adapt by reacting to changes in time derivatives of kinetic profiles and their spatial gradients so that turbulence can adapt to those changes [129]. This approach halves the next time step size if on the current step profiles and the gradients change more than a prescribed limit:

$$\begin{gathered}
\Big|\frac{T_{e,i}(t-\Delta t) - T_{e,i}(t)}{T_{e,i}(t-\Delta t)}\Big| > \Delta T_{e,i}^{\mathrm{lim}} \\
\Big|\frac{\partial_\rho T_{e,i}(t-\Delta t) - \partial_\rho T_{e,i}(t)}{|\partial_\rho T_{e,i}(t-\Delta t)| + \frac{T_{e,i}(t-\Delta t)}{a}}\Big| > \Delta \partial_\rho T_{e,i}^{\mathrm{lim}}
\end{gathered} \tag{3.2}$$

3.1.2. Equilibrium Code: CHEASE

In order to solve the transport equations for a plasma in a toroidal magnetic field, one should know the particular geometry of the equilibrium magnetic field. By equilibrium, we mean the situation when the forces described by the model of plasma describing *MHD* are in balance, namely, the plasma pressure p is balanced by Lorenz forces of the current $\boldsymbol{J}$ induced by the magnetic field $\boldsymbol{B}$. The MHD equations describe these quantities:

$$\begin{aligned} \boldsymbol{J} \times \boldsymbol{B} &= \nabla p \\ \nabla \times \boldsymbol{B} &= \boldsymbol{J} \\ \nabla \cdot \boldsymbol{B} &= 0 \end{aligned} \tag{3.3}$$

where $\boldsymbol{B}$ is magnetic field, $\boldsymbol{J}$ is current density and p is plasma pressure. The magnetic field is usually described as $\boldsymbol{B} = T\nabla\phi + \nabla\phi \times \nabla\Psi$ in terms of its flux function Ψ.

Usually, the system is casted into the non-linear second-order elliptic Grad-Shafranov equation:

$$\nabla \cdot \frac{1}{R^2}\nabla\Psi = \frac{j_\phi}{R} = -p'(\Psi) - \frac{1}{R^2}TT'(\Psi) \tag{3.4}$$

The code we use to solve this system of equations is fixed-boundary equilibrium code CHEASE[130]. We prescribe the boundary of the closed-flux-surface region of plasma $\partial\Omega$ where magnetic flux is zero $\Psi|_{\partial\Omega} = 0$.

When dealing with axisymmetric tokamak plasma, the primary description of equilibrium is done in terms of the $2D$ distribution of quantities across a poloidal cross-section of a plasma. Describing level sets of particular quantities is done discretely by employing bicubic Hermite finite elements.

The quantities that are interesting in the context of the transport equation include the safety factor or inverse rotational transform $q(\rho)$ describing the ratio of toroidal to poloidal windings of the magnetic field around the torus. Also, its derivative quantity, the magnetic shear $\hat{s} = \frac{d \log q}{d \log \rho}$ plays a role. Moreover, the information of the magnetic flux defines the geometry of the coordinate system used for the turbulence and transport computation, including the computation of ρ_{tor}. The metric tensor, which $gm_3(\rho)$ component, describes the scaling of the magnetic-field-aligned radial coordinate with respect to the lab system and is the most important for the radial transport description. Finally, some global scalar magnetic quantities are important, like the ratio of hydrodynamic plasma and magnetic pressures β.

3.1.3. Turbulence Code: GEM

The transport processes simulated with the workflow are governed by the effective transport coefficients as a function of the plasma state, defined by average effective radial outward heat and particle fluxes across the magnetic flux surface. Since the highest contribution to overall transport processes is attributed to anomalous turbulent transport, in order to compute transport fluxes, we use a code that solves equations that capture turbulence. This work uses the gyrofluid electromagnetic nonlinear 3D code GEM [131, 132].

This code uses the gyrokinetic theory of plasma to derive equations described in terms of a finite number of moments of a distribution function in particle phase space. In the particular model of the code, six moments are used to describe the distribution function of a toroidal plasma.

Here, the analytical reduction of kinetic theory to gyrokinetic theory is followed by the expression of the velocity-space components in terms of integral (over velocity space) quantities. The kinetic theory assumes that 6 degrees of freedom, namely a spacial coordinate vector $\boldsymbol{x} \in \mathbb{R}^3$ in some coordinate system and its velocity vector $\boldsymbol{v} \in \mathbb{R}^3$, fully describes a particle state. Conversely, a system of N particles is described via its distribution function $f(\boldsymbol{x},\boldsymbol{v}) : \mathbb{R}^6 \to [0,1]$ describing a probability density of particles over the particle phase space, or in other interpretation a fraction of particles being in an infinitesimal $d\boldsymbol{x} \times d\boldsymbol{v}$ volume of phase space. The phase space of states of the kinetic system, thus, is a Hilbert space $L^1(\mathbb{R}^6, \lambda, [0,1])$ of unit-integrable functions weighted with a natural Lebesgue measure with a valid symbolic substitution $d\lambda(x) \to \prod_i dx_i$: $\int_{\mathbb{R}^6} (f(x))^1 d\lambda(x) = \int_{\mathbb{R}^3} \int_{\mathbb{R}^3} f(\boldsymbol{x},\boldsymbol{v}) d\boldsymbol{x} d\boldsymbol{v} = 1$. The gyrokinetic approach allows for the reduction of the velocity part of the particle phase space to a dimensionality of 2, utilising the fact that the fastest particle motion scale is gyration around magnetic field lines. For example, using gyro-averaging operators brings the equation towards the description of the motion of gyrating discs and their respective centres, and a change of coordinates leads to an effective $2D$ particle velocity phase space [13]. The next step is describing the velocity space distribution with a finite number of moments, meaning integrals of power functions of quantities employing the distribution function as weight $M_n = \int_{\mathbb{R}^3} \boldsymbol{v}^n f(\boldsymbol{x},\boldsymbol{v}) d\boldsymbol{x} d\boldsymbol{v}$. One might also see this approach in a dual manner as describing the distribution function f in terms of projections on a finite polynomial basis $\{\boldsymbol{v}^n\}$. This description is called a fluid model and describes systems in the vicinity of an equilibrium distribution function. Here, the code is implementing a six-moment fluid model, using the first moments of the distribution function, namely number density $n_{e,i}$, moment density $n_{e,i}\boldsymbol{u}$, and temperatures $T_{e,i}$, conductive heat flux $q_{e,i}$.

In this workflow, we use the *flux tube* approximation of the code, where all the processes are defined for the vicinity of a particular magnetic field line winding around the torus and described with a small interval of poloidal coordinate values. Such a flux tube is considered a line object with respect to transport scales. The flux tube description also assumes periodicity at the poloidal and toroidal angles. This model includes the gradient drive terms explicitly and considers locally linear temperature and density profiles with constant gradients, defined as $n(\rho) = -|L_\perp \nabla_\rho \ln n| \cdot \rho$ and $T(\rho) = -|L_\perp \nabla_\rho \ln T| \cdot \rho$. As one of the first steps to start turbulence computations for transport, we define the flux tube locations spanning the entire minor radius of the device to interpolate well the transport properties for the whole plasma. In this work, we simulate AUG tokamak for which we choose 8 flux tubes at different radial locations of plasma. For the AUG plasma, we are using the location of flux tubes defined by $\rho^{\text{norm}}_{\text{tor, fluxtube}} = \left(\frac{n+0.5}{N}\right)^{0.7}$.

All the properties defined by the transport code on its more finely-resolved radial grid shall be interpolated onto the transport grid of flux tube locations, which we perform using 3^{rd} order Lagrange polynomials in a fashion similar to the stochastic collocation method and formula 2.3.2 while using core profile radial grid points as collocation points.

The internal GEM time step is computed using profile perpendicular length scale $L_\perp$ and speed of sound c_s as $\Delta t_{\text{GEM}} = 2 \cdot 10^{-3} L_\perp / c_s$. This definition makes turbulence simulation time step vary when the profiles are updated by a transport code.

The last part of the turbulence simulation we are interested in is the computation of the effective radial fluxes. In order to capture the effective mean level of flux value, we reduce the influence of the fluctuation of this integral quantity in time via exponential averaging $\langle Q(t+\Delta t)\rangle = (1-\alpha)\langle Q(t)\rangle + \alpha Q(t+\Delta t)$, or in continuous version $\langle Q(t)\rangle = \int_{-\infty}^{t} \alpha e^{-\alpha(t-t')} Q(t')dt'$, leaving a low coefficient of forgetting at the level of $\alpha = 1/200$. This allows to smooth the fluctuations of the flux values on the finest internal time scale of the turbulence simulation.

Regarding the parallelisation scaling properties, the local flux tube version of the code scales well up to 1024 cores for 8 flux tubes, requiring time of order 30s to compute $50\Delta t_{\mathrm{GEM}}$. In this work, we used 32 cores dedicated to a single flux tube simulation, which fits on a single node of all major contemporary HPC systems and leverages fast intra-node memory communication.

Turbulence is a highly nonlinear process, and, to some degree, it could be decomposed on different scales, and while exposing highly nonlinear local behaviour with time on a small spatial scale, the dependency of its averaged integral QoIs, like heat flux, on independent parameters like free energy is more regular.

This model captures the ITG turbulence, often dominant in transport for tokamak core plasma. This consideration led to particular attention to the code sensitivity towards ion temperature gradient ∇T_i, which is the main drive for this type of turbulence.

3.2. Formalism for Multiscale Components

This understanding of *scale separation* allows such system splitting and treating turbulence as a slow-changing quasi-stationary process and turbulent transport model as a map $\boldsymbol{q} : \mathbb{S} \to \mathbb{Q}$ between independent global parameters, for instance temperature gradients $\nabla T_i, \nabla T_e$, or more generally some global plasma state $\boldsymbol{s} \in \mathbb{S}$, which in this work is described via core kinetic profiles (a set of functions of radial coordinate) and magnetic equilibrium, and possible values of transport, in this work primarily described as average turbulent heat flux values from $\mathbb{Q}$, but more generally a set of different transport coefficients. The consideration of scale separation and the multi-component approach to multiscale problems was discussed in an earlier section 2.1. Furthermore, this allows for a distinction between notions of aleatoric and epistemic uncertainties, where the former describes fluctuations around some uncertain effective value of transport flux $\boldsymbol{q} \sim p_{\mathrm{al}}(\boldsymbol{q})|_{\boldsymbol{s}}$ for a fixed plasma state $\boldsymbol{s}$ and the latter describes the variation of the transport flux values $\boldsymbol{q} \sim p_{\mathrm{ep}}(\boldsymbol{q}) = p_{\mathrm{ep}}(\boldsymbol{q}(\boldsymbol{s}))$ due to a finite (probabilistic) perturbation of the plasma state $\boldsymbol{s} \sim p(\boldsymbol{s})$ itself.

With the given notation, the equilibrium code implements a mapping $EQUIL : \mathbf{S}^{\mathrm{c.p.}} \to \mathbf{S}^{\mathrm{eq}}$ of core profile properties of a plasma state $\boldsymbol{s}^{\mathrm{c.p}} \in \mathbf{S}^{\mathrm{c.p.}}$ to a self-consistent equilibrium part of the plasma state $\mathbf{S}^{\mathrm{eq}} \ni \boldsymbol{s}^{\mathrm{eq}} = EQUIL(\boldsymbol{s}^{\mathrm{c.p}})$, describing quantities like the inverse rotational transform $q(\rho)$. The transport code, respectively, evolves the core profile component, including temperature profiles $T_{e,i}(\rho)$ and their gradients $\nabla T_{e,i}(\rho)$, of a plasma state $\boldsymbol{s}^{\mathrm{c.p}}(t)$ with time for a given time step Δt of time integration of the ODE solver for a given fixed magnetic equilibrium $\boldsymbol{s}^{\mathrm{eq}}$ and for a given transport coefficients or transport flux parametrisation $\boldsymbol{q}(\boldsymbol{s})$, shortened to $TRANSP : \mathbf{S}^{\mathrm{c.p.}} \times \mathbf{S}^{\mathrm{eq}} \times \mathbf{Q} \times T \to \mathbf{S}^{\mathrm{c.p.}}$ and $\boldsymbol{s}^{\mathrm{c.p}}(t+\Delta t) = TRANSP\left(\boldsymbol{s}^{\mathrm{c.p}}(t), \boldsymbol{s}^{\mathrm{eq}}, \boldsymbol{q}(\boldsymbol{s}), \Delta t\right)$. The transport parametrisation, the main object of the study, is implemented via plasma turbulence code $TURB : \mathbf{S}^{\text{c.p.}} \times \mathbf{S}^{\text{eq}} \rightarrow \mathbf{Q}$, capable of computing average transport fluxes, like radial heat flux $Q_{e,i}(\rho)$ and respective effective transport coefficients, like diffusivity D and convection velocity v_{conv} for a given quasi-stationary magnetic equilibrium $\boldsymbol{s}^{\text{eq}}$ and core profiles $\boldsymbol{s}^{\text{c.p.}}$, expressing that as $\boldsymbol{q}(\boldsymbol{s}) = TURB(\boldsymbol{s}^{\text{c.p}}, \boldsymbol{s}^{\text{eq}})$.

Furthermore, having introduced these elements, we can speak of a total plasma evolution operator describing the working of our computational workflow $WF_{\Delta t}(\boldsymbol{s})$ acting as an endomorphism on the space of plasma states $WF_{\Delta t}$: $\mathbb{S} \rightarrow \mathbb{S}$ and describing the change of plasma state $\boldsymbol{s}(t + \Delta t) = WF_{\Delta t}(\boldsymbol{s}(t))$ in a time of Δt. We define this operator as a composition of our three mappings

$$WF_{\Delta t}(\boldsymbol{s}) = \langle TRANSP\,(\boldsymbol{s}^{c.p}, EQUIL(\boldsymbol{s}^{c.p}), TURB(\boldsymbol{s}^{c.p}, EQUIL(\boldsymbol{s}^{c.p})), \Delta t)\,, \quad EQUIL(\boldsymbol{s}^{c.p})\rangle \tag{3.5}$$

with a natural computational re-utilisation of equilibrium operator result and a notational shortening to $WF(\cdot)$ for future usage convenience.

3.3. MFW Parameters

The presented coupled system requires a number of parameters to close the system and make it solvable. These parameters define a case for which the model has to be solved, and, from the system's point of view, they must be provided as a piece of external information. This could be done via further coupling with other physical models or via a choice of parametrisation by a user for a particular study.

As for most PDE systems, these parameters include the right-hand side of the equation (RHS), the boundary conditions (BC), and the initial condition (IC). Here, the RHS is defined by the sources and sinks of energy and particles for a particular plasma. In general, sources have to be defined by the physical model for plasma heating and injection systems and sinks via a model of exhaust through the plasma's edge, its scrape-off layer, and tokamak wall and divertor processes [12]. Here, however, we model sources as a beam of heat and particles with a Gaussian radial profile, which is parametrised by three scalars: its position H_0, width H_w and total injected heat $Q_{e,i}^{\text{tot}}$. For the radially outer part of the plasma, the BC is considered to be of Dirichlet type with a prescribed ion and heat temperature T_e^{BC}, T_i^{BC} at $\rho_{\text{tor}}^{\text{norm}} = 1$. Given that the model assumes that the plasma transport adapts self-consistently to the injected power, the model is not expected to be sensitive to the initial conditions. However, those have to be specified in terms of particular radial profiles of temperatures $T_{e,i}^{\text{IC}}(\rho)$ and densities $n_{e,i}^{\text{IC}}(\rho)$, as well as an equilibrium magnetic field. These are taken from a particular AUG discharge. Here, for most of the cases, we use standard H-mode discharge #28906. With this description, the small number of scalar physical quantities that define the temperature evolution of a discharge is required to specify the macro-model $n_M = 6$.

Furthermore, a number of numerical parameters have to be specified. Most importantly, one has to define the total duration of the simulation. For the given discharge, it has to be taken at least of the order of confinement time, which is $\tau_E \approx 0.3s$. For fluctuating profiles, one might need to gather sufficient statistics in time, requiring several seconds of simulation duration. Furthermore, one should define the transport time step Δt, which would be sufficient to capture the characteristic time derivatives and rate of change for the evolution of plasma profiles. In the general case, the time step has to adapt to the time derivatives of profiles and their gradients, which was implemented into ETS and MFW via time bridging technique [129]. Finally, one needs to specify the resolution of the configurational domain, both a global radial domain for transport scales and a local domain for turbulence simulations. Here, we use 100 points to resolve the radial coordinate $\rho_{\mathrm{tor}}^{\mathrm{norm}}$, the normalised toroidal magnetic flux.

3.3.1. Plasma State and Dimensionality Reduction

The introduced notion of *plasma state*, while being quite abstract, has a concrete meaning for the implementation of the computational workflow. Here, the plasma state is defined by a finite vector in $\mathbb{R}^d$ where d is defined by the core profile and equilibrium radially-distributed quantities like temperature for two species $T_{e,i}$, their gradients $\nabla T_{e,i}$, and rotational transform q. Each of these quantities is resolved at a finite number of radial points. In practice, it means storing parameter values in suitable data structures, like CPO, and reading and saving data in particular types of files.

However, in general case, one could define a state of plasma in a number of ways, including $2D$ and $3D$ fields. The computational efficiency and even applicability of algorithms for uncertainty quantification, including algorithms involving surrogates, would depend on the effective dimensionality of such a plasma state representation, meaning the total number of scalars or DoFs needed to fully distinguish any two meaningfully different plasma states for a given model.

In the general case, bridging the physical models, requiring multiple, often $\mathcal{O}(10^d), d > 6$ DoFs, and uncertainty capturing methods, should include *dimensionality reduction* methods.

One of the most widely used methods is *Principle Component Analysis (PCA)*, related to *Singular Value Decomposition (SVD)*, which allows us to find linear combinations of factors or input components, explaining the most variance in the data. This method, however, utilises only linear interaction between independent variables and usually does not account for the problem for which its results would be used; hence, it does not involve data on QoIs. The issue of linearity could be solved by utilising *encoder-decoder* network architecture, including *Variational Autoencoder (VAE)*, that can find small effective latent spaces describing objects of high dimensionality very effectively. These models are artificial neural networks that allow for arbitrary complexity of the data and high levels of non-linear interactions among the degrees of freedom. On the other hand, the methods based on the Sensitivity Analysis are designed to explain contributions to the variance of particular QoIs for the problem and are wired to be used for the solutions of computational problems. Methods that utilise both sensitivity indices and non-linear encoding, like *Deep Active Subspace (DAS)* [133, 134], often serve as effective methods for dimensionality reduction for algorithms that use surrogate models in various fields of computational science.

3.4. Workflow Time Integration

The workflow implements time integration of a solution of plasma heat and particle equations. It starts with setting up the initial core profiles, initial MHD equilibrium, and the heat and particle sources. Also, the toroidal magnetic field, impurities, and neutrals must be specified. The latter two are for simplicity and are absent in this work. From the software point of view, it required reading the corresponding CPO data structures and informing the *TRANSP* component to initialise the solution.

On every iteration, the workflow would modify the core plasma kinetic profiles $T_{e,i}$ and $n_{e,i}$, resolved on 100 points of the radial grid, according to the time discretisation of the equations, which changes our macroscopic state of the plasma $\boldsymbol{s}_{\text{c.p.}}(t) \to \boldsymbol{s}_{\text{c.p.}}(t + \Delta t)$. This is usually done with a fixed time step size Δt. However, it can adapt to the derivatives of the profiles as described in section 3.1.1. This work, however, keeps density profiles constant for the sake of simplicity. Each step of the iteration, programmatically, starts with receiving the necessary information in CPO format by a respective component and ends with sending out the information on the partial solution.

For every iteration, the workflow computes a MHD equilibrium $\boldsymbol{s}_{\text{eq}}(t)$ consistent with kinetic profiles by the *EQUIL* component. An iteration substep then starts with the *TRANSP* component sending out a signal and a `coreprof` CPO to the *EQUIL* code, followed by the run of CHEASE and sending a final signal and a `equilibrium` CPO to all other active workflow components. In practice, the equilibrium changes significantly only during the first few iterations of the workflow, reaching fast convergence.

Also, on every iteration, the workflow computes the transport coefficient D and v consistent with core profiles $\boldsymbol{s}_{\text{c.p.}}$ and equilibrium $\boldsymbol{s}_{\text{eq}}$. This is done first by performing turbulence simulations, calling the *TURB* component, computing the radial heat fluxes $\boldsymbol{q}$ (and possibly particle fluxes $\Gamma_{e,i}$), and then recomputing effective transport coefficients via a "Fluxes-to-coefficients" component, implemented via IMP4DV utility code in this work $D, v = FLUX2COEFF(Q_e, Q_i)$. This step requires deciding on a particular flux splitting between diffusive and convective components, resolving this additional degree of freedom with physical considerations. The iteration substep starts with *TURB* receiving the `coreprof` and `equilibrium` CPOs, followed by a GEM simulation, the most expensive part of the workflow, sending `coretransp` to IMP4DV to recompute transport coefficients, and finally sending the completed version of `coretransp` to *TRANSP* workflow component.

For simple turbulence models, the profiles and the transport coefficients change drastically over a time comparable to confinement time $t \sim \tau_E$, equivalent to several tens of transport time steps $n_{\text{t.st.}}^{\text{TRANSP}} \sim \mathcal{O}(10)$ and then reach a slower phase of convergence towards a stationary plasma state $\boldsymbol{s}^{\text{st}}$. The convergence corresponds to a self-consistency between profiles and the transport model, capturing the balance between the power sources and the turbulent transport via finding a consistent turbulent drive.

A time step Δt too large could lead to a convergence of the solution towards a steady state and non-physical oscillations of the system around the equilibrium of fluxes and profiles that should be treated as numerical artifacts.

In the general case, the profiles dynamically and non-linearly interact with the microscopic turbulence, which results in fluctuating behaviour. The high-fidelity turbulence codes are highly non-linear and sensitive to their external parameters, which could be expressed in non-trivial conditions for the turbulence onset, turbulence marginality, and complicated dependency of the effective transport. This naturally leads to more complicated interactions between profiles and fluxes, resulting in fluctuations. Furthermore, they expose the *stochastic* dynamics property in the computation of the effective transport, meaning there will be a finite perturbation from the equilibrium state on every time iteration. This leads to the emergence of *irreducible* uncertainty and the impossibility of finding the exact equilibrium of plasma state $\boldsymbol{s}^{\mathrm{st}}$ due to the uncertainty in turbulent fluxes. The workflow time integration reaches a *quasi-steady state* fluctuating in the vicinity of the attracting equilibrium state $\boldsymbol{s}^{\mathrm{st}}$. Given the uncertain nature of turbulence computation, it is required to capture all the related uncertainties, primarily the uncertainties of the macroscopic plasma state $p(\boldsymbol{s})$ and in the turbulent response $p(\boldsymbol{q})$ to describe the quasi-stationary state of plasma experiencing turbulent transport.

3.4.1. Workflow Run with Analytical Turbulence Model

As a tool for prototyping and proof-of-principle methodology development, the work was started by using the GEM0 model. This model estimates the anomalous heat and particle fluxes for electron and ion species based on GyroBohm scaling

$$\begin{aligned} Q_i &\sim \frac{n_i T_e^{\frac{5}{2}} |\nabla \ln T_i|}{a|B_0|\ L_\perp} \sim \frac{T_e^{\frac{3}{2}} |\nabla T_i|}{L_\perp} \\ Q_e &\sim \frac{n_e T_e^{\frac{5}{2}} |\nabla \ln T_e|}{a|B_0|\ L_\perp} \sim \frac{T_e^{\frac{3}{2}} |\nabla T_e|}{L_\perp} \end{aligned} \tag{3.6}$$

where a is a tokamak minor radius, B_0 is magnetic field, and $L_\perp = \min\left(R, a|\frac{T_e}{\nabla T_e}|, a|\frac{T_i}{\nabla T_i}|\right)$ is a profile length scale.

This model serves as a simple plug-in replacement of the gyrofluid GEM model, identical from the software organisation point of view; however, $\sim 10^5$ times less in computational cost. Apart from the low physical fidelity, the GEM0 model does not expose any aleatoric uncertainty. Firstly, it means that the convergence of a coupled transport workflow happens significantly faster, and the self-consistent solution does not expose any fluctuations. Secondly, this model is unsuitable for studies of aleatoric uncertainties in turbulent simulations, and this problem requires another approach.

The figure 3.2 shows an MFW simulation of an AUG discharge around a steady state using an analytical GEM0 model for turbulence.

3.4.2. Workflow Run with High-fidelity Turbulence Model

This subsection describes previous results in running a multiscale multi-component Transport-Equilibrium-Turbulence workflow MFW using nonlinear 3D electromagnetic code GEM to compute turbulent transport. Figure 3.3 shows the resulting self-consistent ion and electron

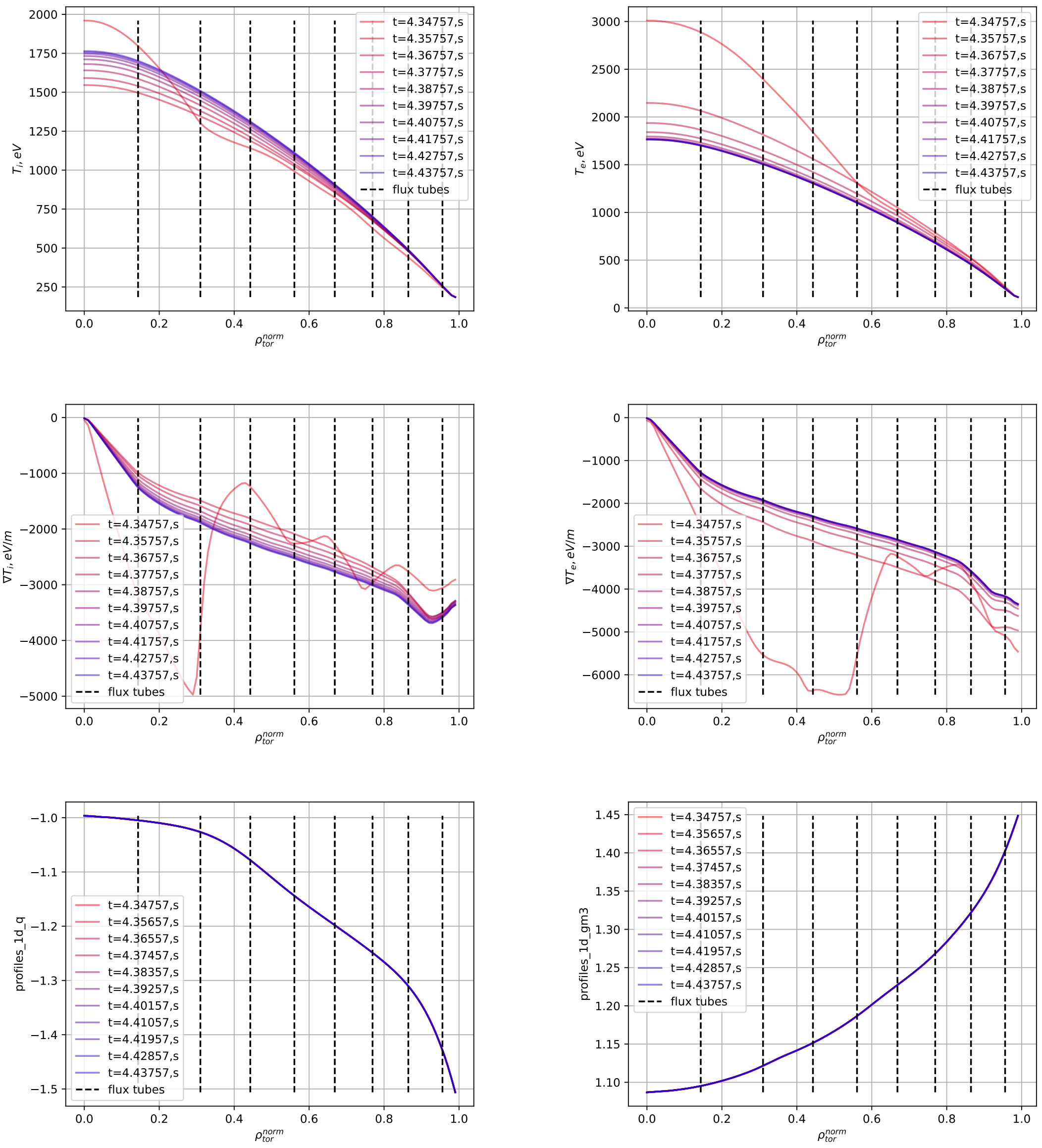

Figure 3.2.: Convergence of T_i and T_e core profiles, and their gradients ∇T_i and ∇T_e, as well equilibrium $1D$ profiles of q and gm_3, to a self-consistent plasma state during an MFW simulation with GEM0 for turbulent transport model. The transition from red to blue denotes the pass of the simulation time, and the dashed lines represent the flux tube locations.

temperature profiles computed by the workflow compared to the experimental measurements of different AUG shots. The demonstrated results were conducted in [123]. The run had 7500 time steps required to capture the quasi-stationary state of plasma. Experimental uncertainties show the need to estimate their modelling counterparts for consistent and sound validation. The fluctuating flux values indicate the uncertain nature of turbulence, with statistical preservation of mean values dictated by the power balance but overall stochastic dynamics.

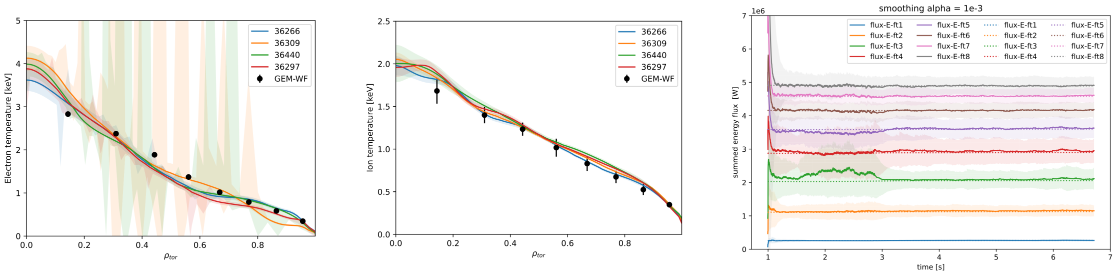

Figure 3.3.: The stationary plasma quantities computed by MFW using GEM for turbulence solution. The two leftmost figures show the electron (left) and ion (middle) temperature profiles. The solid line indicates the comparable experimental measurements, with shaded areas denoting the experimental uncertainties. The black lines and error bars are respective quantities computed by the workflow at the flux tube locations. The rightmost figure shows the evolution of total fluxes computed by the workflow for 8 respective flux tubes. The dotted lines indicate the respective fluxes computed via power source integration. The presented results are from [123].

The results were performed with an old version of the workflow using MUSCLE2 library. They do not account for the initial epistemic uncertainty and are computationally expensive. The modification introduced in this work and described in a later chapter, section 5.1, brings capabilities to significantly speed up the workflow via considerate surrogate exploitation, which enables uncertainty propagation for such a workflow.

Figure 3.4 shows MFW run, in its modification implemented in this work, using GEM for turbulence component. Due to the high computational cost, the simulations were not performed until the system reached a quasi-stationarity state.

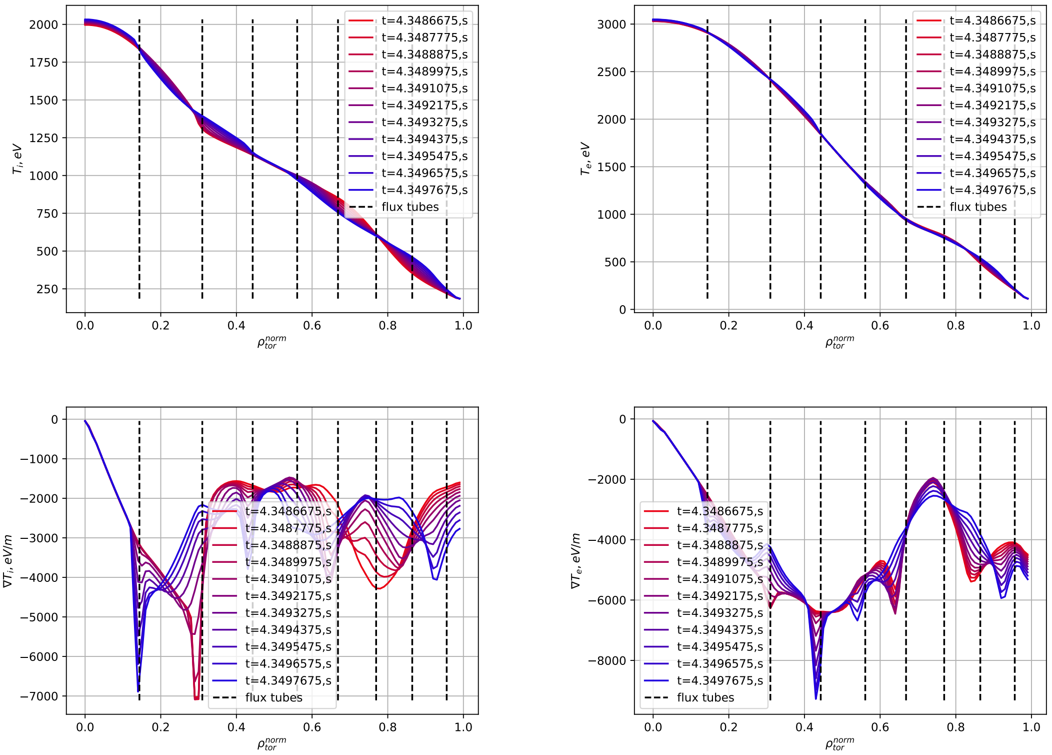

Figure 3.4.: An evolution of T_i, T_e core profiles, and their gradients ∇T_i,∇T_e using MFW simulation with GEM for turbulent transport model. The transition from red to blue denotes the pass of the simulation time, and the dashed lines represent the flux tube locations.

3.5. Chapter Summary

This chapter presents the Multiscale Fusion Workflow, a multi-component computational workflow that solves nonstationary heat and particle transport equations for core tokamak plasmas self-consistently. We presented components to solve equations for transport, equilibrium, and turbulent fluxes and discussed the software and other aspects of the implementation. Furthermore, we demonstrated a solution for stationary core profiles using the workflow.

The following two chapters cover the questions on utilising data-driven surrogates to enable uncertainty quantification of turbulent transport models and how both aleatoric and epistemic uncertainties manifest in high-fidelity turbulence models.

4. Uncertainty Quantification and Surrogate Modelling Results for Turbulence Code

This chapter focuses on the studies performed on the gyrofluid electromagnetic plasma turbulence code GEM. We discuss the procedure for estimating properties of the time series describing the fluctuating turbulent transport fluxes, which allows computing the uncertainties of the effective transport and deciding on the sufficient turbulence simulation duration. Furthermore, the quantification of epistemic parametric uncertainties on GEM and its parametric sensitivity analysis are discussed. Finally, the chapter talks about surrogate models for turbulent transport simulations, and their training, validation, and utilisation aspects are addressed.

4.1. Aleatoric Uncertainty and Fluxes Time Traces Analysis

The study of uncertainty benefits from identifying different sources of uncertainty and uncertainties of various types. This section describes the uncertainties of estimating effective transport fluxes from plasma turbulence simulations. We consider turbulent heat fluxes attributed to ions Q_i and electrons Q_e as a function of time and radial coordinate, computed as integral quantities of plasma turbulence simulations by GEM code. Furthermore, from the macroscopic point of view, the flux values are described as a function of plasma state, mainly the kinetic profiles, which in this work mean ion and electron temperature T_i, T_e and their gradients ∇T_i and ∇T_e, with the ion temperature gradient often being the most influential parameter, driving the ITG turbulence prominent for transport studies in this work. The time dependency of the fluxes during a simulation follows the general pattern of initial exponential growth followed by domination of non-linear effects leading to turbulence saturation and quasistationary behaviour of fluxes around a particular value. Without a change in external parameters, the mean values of fluxes stay the same; however, the overall flux dynamics are chaotic and fluctuate with time. Such behaviour leads to the analysis of transport fluxes as a result of a stochastic process and to the importance of time traces analysis of flux values.

Figure 4.1 shows a typical situation of the parametric dependency cases of the turbulence model, here ion heat fluxes produced by GEM. In this case, which is complicated to analyse from a naïve perspective, there is a high possibility of a meaningful dependency of QoI values on the model's independent parameters. However, the level of fluctuations of QoI and resulting aleatoric uncertainty does not allow an apparent reconstruction of such dependency.

Such a situation, often appearing in turbulence simulation, leads to the need for the separation of aleatoric and epistemic uncertainties. Figure 4.2 shows how an algorithm 2.2.2, applied

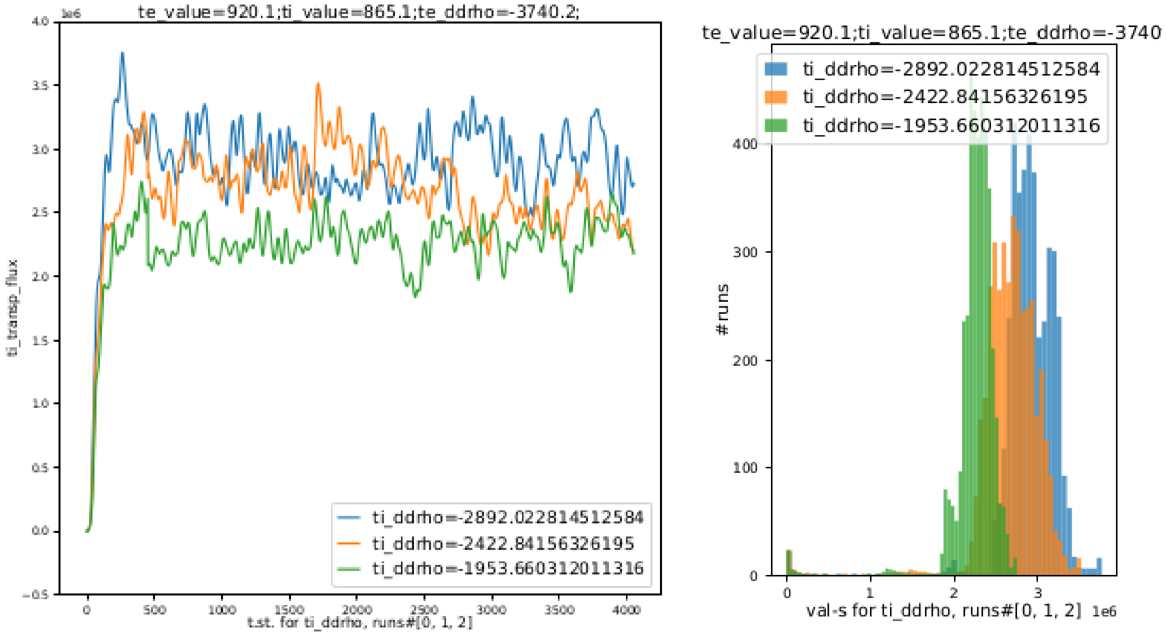

Figure 4.1.: The dependency of Q_i on ∇T_i at $\rho = 0.7$ for a fixed set of other core profile values computed by GEM. The distributions of the heat flux values significantly overlap, and the estimation of the mean has an error that should not be neglected while reconstructing the dependency and performing epistemic uncertainty studies.

to every instance of turbulence simulation, together with a consideration of controllable numerical parameters like simulation duration, allows distinguishing well-posed, from the point of uncertainty, simulation cases and influence the separation of uncertainties to recover the dependencies of interest.

In this work, we created a software workflow to manage the numerical experimental design and sampling of the uncertain parameter values, the simulation performance and analysis, as well as the collection, management, analysis, and storage of simulation result data, targeting the subsequent workflow concerning the work with surrogate database models. As a part of simulation management and the result analysis steps, the functionality to sequentially process the time-dependent scalar quantities of interest described in procedure 2.2.2 was implemented. As mentioned, the primary quantities of interest analysed in the turbulent transport GEM model solution are time traces of outward radial heat fluxes associated with a specific particle species.

Every GEM simulation run for a fixed set of input parameters, namely the core profile description of a plasma state $\boldsymbol{s}_{\text{c.p.}}$ (temperatures for every species, density, and respective gradient), as well as plasma equilibrium $\boldsymbol{s}_{\text{eq}}$ geometry of magnetic field (metric tensor, magnetic flux, pressure, safety factor, and current), underwent such an analysis described in procedure 2.2.2.

The result of such an analysis gave us the final statistical error for the estimation of effective transport level for a given case as well as a quantification of the fluctuation of the respective

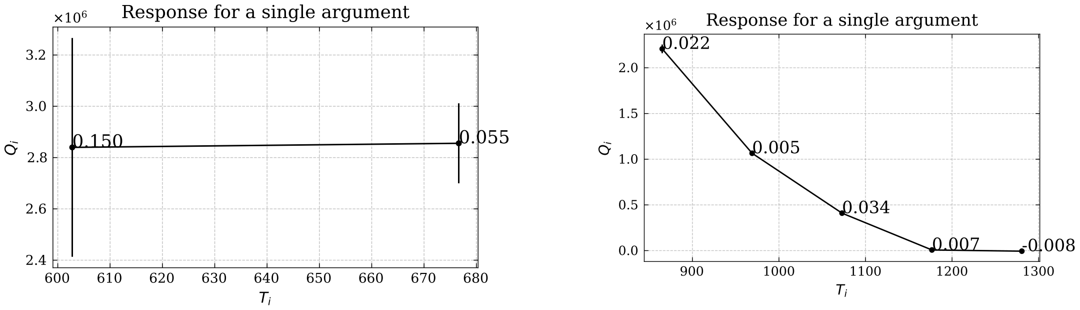

Figure 4.2.: Different cases of aleatoric uncertainty levels of fluxes computed by GEM. Introducing control of numerical parameters to manage the uncertainties in the model allowed us to recover model behaviour with sufficient clarity. Here, we iteratively increase the polynomial order of PCE analysis to raise the level of accuracy in capturing the ion heat flux value dependent on the ion temperature of a flux tube. The level of temperature uncertainty defines the study's domain, and the simulation's duration is defined to control the error in the mean behaviour estimate.

fluxes, characterising the aleatoric uncertainty.

Here is a result of such an analysis using the 648 GEM turbulence simulations sample. Approaching the problem retrospectively during the postprocessing allowed us to judge the accuracy of the mean flux value estimates as a function of simulation duration and tolerance criteria values and analyse the estimation accuracy as a function of tolerance criteria.

4.1.1. Time Traces Analysis

A result of this algorithm applied for an ion heat fluxes time evolution at a particular flux tube ρ during a single instance of the GEM simulation for fixed values describing plasma's kinetic profiles and magnetic equilibrium $\boldsymbol{s}$ is shown in the figure 4.3, and the statistics convergence is shown in figure 4.4. For GEM simulations in the studied regime, the SEM converges after $\mathcal{O}(10)$ of simulation time windows.

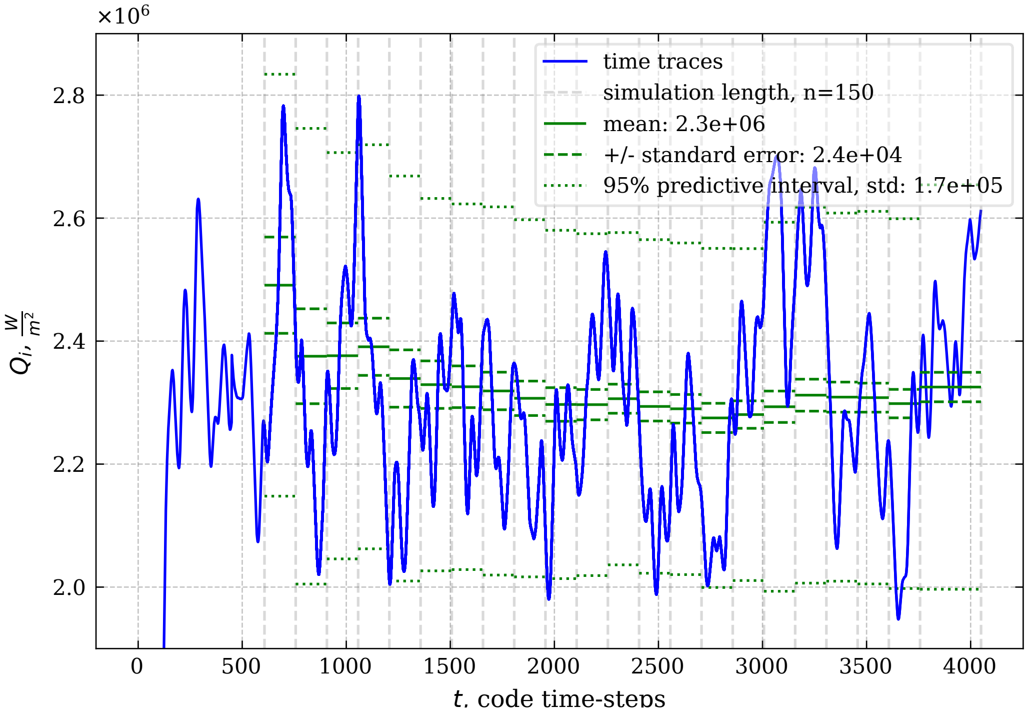

Figure 4.3.: An example of the results of the postfactum application of time trace analysis. Readings left of the leftmost grey vertical line are considered to relate to the turbulence growth phase and not to be part of the stationary process. The black vertical line indicates the suggested end of the simulation, as all the readings right to it do not bring much additional information for the statistical estimates. The vertical dashed lines denote the right-hand limits of the simulation time windows. Each iteration of the sequential analysis uses all readings left to the corresponding dashed line (but to the right of the leftmost dashed line). The horizontal solid line denotes the mean of the effective sample for this window, the dashed lines represent the standard error of the mean, and the dotted lines denote 95% confidence intervals. For a given simulation case and given convergence thresholds for mean change and SEM, the accuracy of statistical estimates for flux QoI would not significantly improve after a total of 2600 steps.

Finding the effective values of transport fluxes allows us to distinguish the aleatoric and epistemic uncertainty and apply uncertainty propagation analysis to average fluxes. Furthermore, for this procedure, we use convergence criteria crit$_i$ and simulation duration $t_{\text{turb}}^{\text{tot}}$ as a free numerical parameter to control the fidelity of turbulent flux estimation. Setting convergence criteria for the estimates of simulation data allows us to balance the accuracy and the simulation cost and to effectively use the compute resources via an early-stopping mechanism.

The following section describes the approach and results in creating surrogate models for GEM as a dependency of heat fluxes $\boldsymbol{Q} = (Q_e, Q_i)$ on the plasma state $\boldsymbol{s}$ and its core profiles

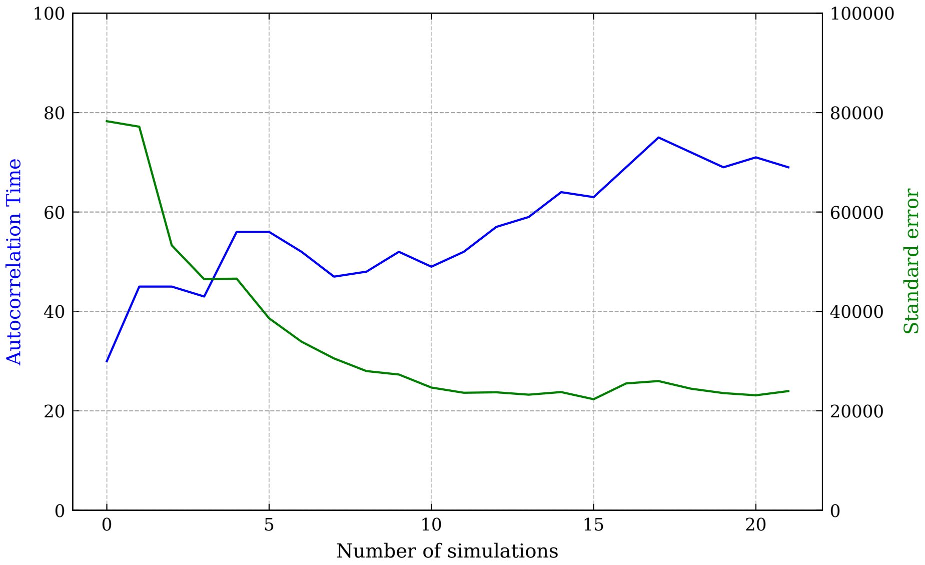

Figure 4.4.: A plot of the convergence of ACT and SEM with the sequential processing of the time trace from 4.3.

and equilibrium, to use it as a component of the turbulence-transport-equilibrium workflow. Furthermore, it will elaborate on how to manage the fidelity of the surrogate in the workflow, re-train it in an active manner, and use surrogates to estimate uncertainties.

4.2. Parametric Studies of Gyrofluid Turbulence Model GEM

Having a procedure to analyse the aleatoric uncertainties for single turbulence simulations allowed the next step to be performed, which is the analysis of the behaviour of the 3D electromagnetic gyrofluid model GEM in its flux tube approximation. This was done by performing a Polynomial Chaos Expansion of the model with respect to its most influential independent parameters. The choice of this algorithm for uncertainty analysis is multi-fold. First, it is proven to be a working method to capture uncertainty in the QoI of the model solution for many practical mathematical models in the form of PDE systems, including non-linear ones, yet sufficiently well-behaved. Second, for a given sample of model solution function evaluations, this method provides a comprehensive set of essential statistics for QoI up to high moments of its PDF (variance, skewness, kurtosis) and partial variance associated with an arbitrary subset of independent parameters. The statistics that could be calculated the easiest are Sobol indices, which are used to judge the sensitivity of the model to a given input component and, hence, define the importance of inputs. Third, in association with the first two, for many types of independent parameter PDF, PCE allows refining the resolution associated with particularly important inputs based on increasing the polynomial order of expansion with respect to a chosen quantity, in some cases, reusing the old model solution function evaluations. Fourth, but not least by motivation, the default variants of the

PCE method use quadrature schemes based on full tensor products of the grids for chosen independent parameters and to have the possibility to clearly distinguish the influence of a single parameter by fixing values of others is very important for all initial steps of studying any model.

The flux tube model assumes that the development and behaviour of the plasma turbulence are essentially local and uniform for an entire flux tube, a part of the plasma that follows a given part of a poloidal cross-section along the lines of the magnetic field. Such local behaviour is mainly defined by the local values of the several first central moments of the distribution function, hence the kinetic quantities of temperature and density and, more importantly, their gradients as the main drive of the flow and source of free energy.

For a given AUG discharge, heat transport driven by temperature gradient is believed to be of the most importance. Hence, we focus the most on the variation in this parameter and use it as the exemplary quantity to describe the plasma state from the turbulence point of view most often. Furthermore, the measurements of the temperature gradients, being a derived quantity, are especially problematic to capture and usually have high experimental uncertainties.

4.2.1. Epistemic Uncertainty Quantification

In this work, we analysed a number of GEM runs for flux tubes located at different toroidal plasma radial coordinate values ρ. We varied both temperature $T(\rho)$ values and gradients $\nabla T(\rho)$ for both electrons and main ion species of deutrons.

For an exemplary study of the propagation of epistemic uncertainties for GEM, we took a plasma state $\boldsymbol{s}$ with core profiles and magnetic equilibrium of AUG shot #28906 and assumed a uniform uncertainty for core profiles and their gradient at the locations of 8 flux tubes. To describe the influence of uncertainty in every scalar parameter, we performed a Polynomial Chaos Expansion of the model with the respective choice of samples for every uncertain input, visualised in figure 4.5.

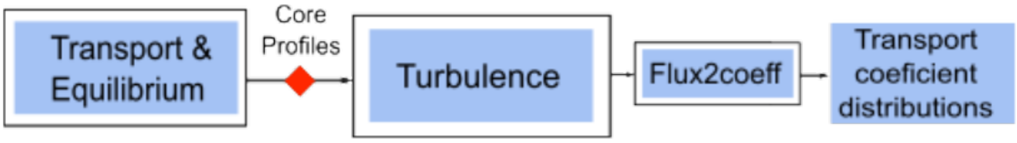

Figure 4.5.: A simple visual explanation of the uncertainty propagation for a turbulence code computing transport coefficients as a dependency of kinetic profiles.

To capture the epistemic uncertainty in the behaviour of turbulent transport heat fluxes $p_{\text{ep}}(\boldsymbol{q})$ depending on the uncertainty in the heat profiles $p(\boldsymbol{s}^{\text{c.p.}})$ we used several quantities. We estimated the *Coefficient of Variation (CV)*, the ratio of standard deviation and mean value $\text{CV}[\boldsymbol{x}] = \sigma[\boldsymbol{x}]/\mu[\boldsymbol{x}]$, of the flux uncertainty. Furthermore, we used the *Uncertainty Amplification Factor (UAF)*, a ratio of *CVs* of functions values to the one for its arguments $\text{UAF}[y|x] = \text{CV}[y]/\text{CV}[x]$ as an important uncertainty metrics for the model [135]. Such a quantity can help judge the global stability of the equation solution with respect to uncertainties and indicate whether a propagating uncertainty would explode or stay at a finite level. To include information on the interdependency of the input and output variables, one could use a modified quantity of *Correlated Uncertainty Amplification Factor* via additional multiplicative covariance factor $\mathrm{cUAF}(y|x) = \mathrm{UAF}(y|x) \cdot \rho(x,y) = \mathrm{UAF}(y|x) \cdot \frac{\mathrm{Cov}(x,y)}{\sigma(x)\sigma(y)}$. Another modification of this metric is to include information per input component by scaling the metrics with Sobol indices to get the *Sensitivity-scaled Uncertainty Amplification Factor* $\mathrm{sUAF}(y|x) = \mathrm{UAF}(y|x) \cdot \mathrm{S}_i$.

For the model flux tube at $\rho = 0.7$, the coefficient of variation for ion heat flux is $\mathrm{CV}[Q_i] \approx 2.96$. The uncertainty amplification factor is $\mathrm{UAF}(Q_i|\boldsymbol{x}) \approx 11.9$, which shows a significant sensitivity of the computational model to the parametric uncertainty.

Choice of parameters: perspective of transport model, analytical theory, and surrogates

It is analytically described and practically proven that turbulence and its effective fluxes are well described as a function of dimensionless parameters like relative profile gradient lengthscales $\mathrm{RLT}_{e,i}$, or $L_{T_{e,i}}$, temperature fraction $\kappa = \frac{T_e}{T_i}$, as well as parameters like plasma β and ρ_*. However, since both experimental measurements and transport codes describe plasma state $\boldsymbol{s}$ in terms of absolute physical quantities like temperature and density, in this work, we have chosen to describe uncertainties and variations in a plasma state with the presented parameters. Even though transforming the plasma state description to the dimensionless parameters can potentially reduce the dimensionality of the problem (and effective surrogate dimensionality, e.g. number of function evaluations required), the turbulent flux model as a function of κ and $L_{T_{e,i}}$ loses some information. Furthermore, the choice of turbulence model and the type of plasma discharges is suitable for studying the influence of the ITG type of turbulence on transport.

4.2.2. Sensitivity Analysis

A class of useful and interesting statistics calculated with the PCE study is Sobol indices, which indicate the model's sensitivity to its parameters by attributing fractions of QoI variance.

It is a global metric, which denotes which fraction of the QoI variance is attributed to a variance of each component, given their prior distribution. As a full expansion of variance as a multivariate function, an approach known as ANOVA also measures sensitivity to the interactions of arguments.

This work uses Sobol indices as a measure of the importance of specific parameters, targeting the utilisation of such measures to decide how well each parameter should be studied and how well its variation should be presented in surrogate models.

Here we indicated the sensitivity of Q_e and Q_i for T_e, T_i, and ∇T_e, ∇T_i.

Figure 4.6 shows the first Sobol sensitivity indices for the GEM0 code. No radial dependency is observed, and further, almost no sensitivity of Q_e on T_i and ∇T_i, and of Q_i on T_i. Electron heat flux can be approximated well as a function of electron temperature gradient only and ion heat flux as a function of both gradients only, respectively.

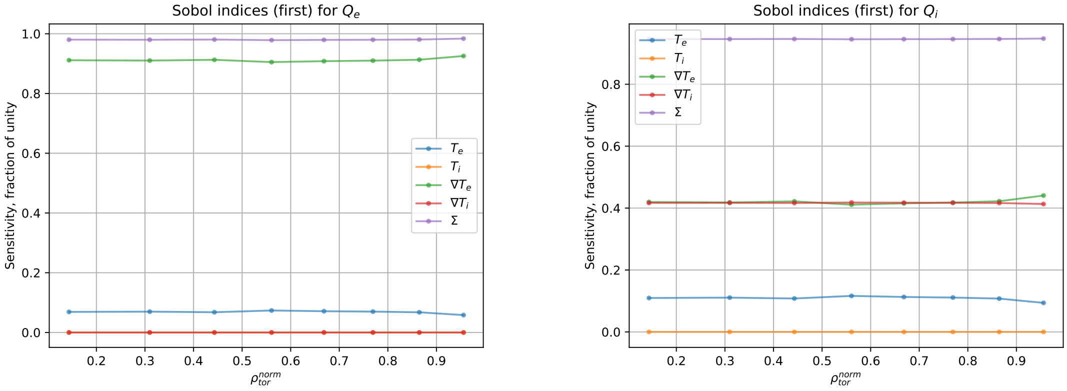

Figure 4.6.: The result of SA ANOVA analysis performed via PCE of order 4 method for GEM0 at different regions of a tokamak plasma.

Figure 4.7 shows the influence of different profile parameters on the turbulent heat fluxes. The variation in Q_i is predominantly explained by the uncertainty in the ∇T_i for most of the radial location of plasma, and T_i being the less influential parameter. Also, in the middle of the plasma at $\rho = 0.3 - 0.7$, some amount of the flux variation is explained via interaction terms.

Unlike the GEM0 case, the sensitivity of the solution of the gyrofluid model GEM shows a significant dependency on the radial coordinate. Even though the turbulence in flux tube approximation should depend mainly on the local quantities and drive terms, the variation of priors distribution of those parameters and variation of equilibrium properties and magnetic geometry introduces a dependency on ρ. Also, unlike GEM0, all parameters play a non-negligible role in the flux as a function of profiles, and the particular level of sensitivity of parameters is different for GEM.

4.3. Surrogate Models

The prime reason for the interest in surrogate models is the cost of the single run of a high-fidelity simulation, which makes prohibitively expensive any extensive parametric studies, including forward and backward uncertainty quantification, as well as usage of the turbulence code in a simulation of a workflow that includes multiple components, each solving its own physical sub-problem.

Furthermore, the surrogates are usually differentiable with respect to the micro-model parameters $\boldsymbol{x}_m$, whereas the legacy HPC codes typically are not, which makes it impossible to use them with different gradient-based methods, including optimisation and derivative-based sensitivity analysis.

In the next part, we are presenting surrogates for heat flux as a function of plasma state $\boldsymbol{q}_{\text{surr}}(\boldsymbol{s})$, results on its validation, as well as multiple scenarios in which one could use surrogates for multiscale simulations and their uncertainty quantification.

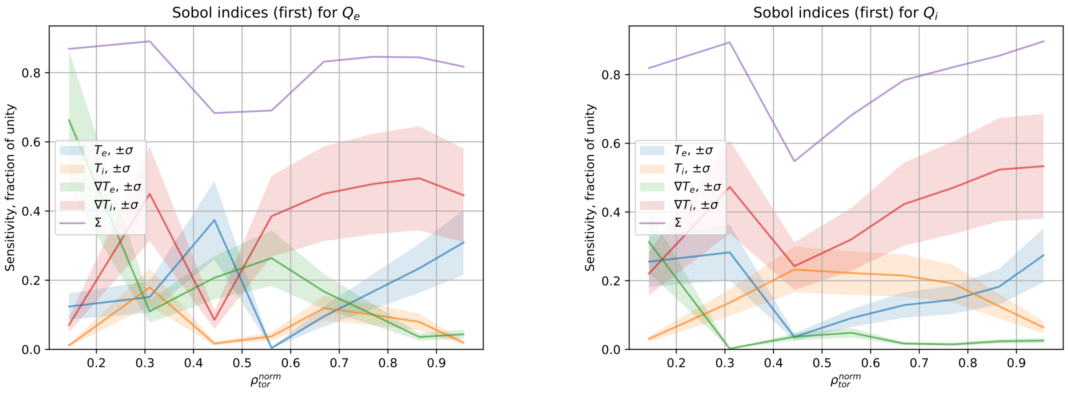

Figure 4.7.: The result of SA ANOVA analysis performed via PCE method for GEM at different regions of a tokamak plasma. Here, we show the first-order Sobol indices of $Q_{e,i}$ computed by GEM as a function of flux tube location $\rho_{\text{tor}}^{\text{norm}}$ for $T_{e,i}$ and $\nabla T_{e,i}$. The sum of index values, showing the interaction effects, is shown. The shaded area indicates the estimated one standard deviation error based on the sample size used to calculate the statistic. The graph indicated that the fluxes are most sensitive to the ion temperature gradient for most flux tubes. The electron temperature gradient is most impactful in the plasma core but loses its importance in the outer parts of plasma. Ion and electron temperatures in all plasma regions explain a fraction of flux variation. Also, the interactions of parameters influence fluxes for all radial locations, but most prominently for $\rho = 0.3 - 0.7$. The character of dependency is more regular for the outer parts of the tokamak core, with some variation in the inner regions.

4.3.1. Gaussian Process Regression

Gaussian Process Regression, described in section 2.6, is this work's prime machine learning surrogate model type. As this type of model predicts the likelihood for values of certain QoI for given independent variables, it would be a good choice of model to capture uncertainties in the underlying process, first of all, due to the lack of parametric data. Furthermore, such a method would be suitable for an application in Bayesian statistics, for example, for Beysian inference of parameter values given some prior PDF of model input parameters, for statistical comparison, validation, and calibration using experimental observable data, and other statistical methods.

The data that were used to train a surrogate describe the response of a turbulence model on a variation of core profiles for an AUG shot (#28906). This data does not include a variation of magnetic geometry as it mainly varies weakly during the chosen phase of the discharge and due to solid indications that turbulent transport fluxes are defined primarily by the kinetic profiles.

Given that core turbulence, analysed in this work, is local and defined on small scales, the surrogate model to capture dependency of ion and electron heat fluxes over a specific radial domain is implemented as a set of independent regression models, each describing a process happening at a particular flux tube located at a specific radial position of a tokamak.

The GPR models in this work use linear regression over the training data to express the model for the mean of the process $\mu(\boldsymbol{x}) = h(\boldsymbol{x})^\top \beta$. This transformation is equivalent to removing the linear trend of the dependency and modelling the residuals as a Gaussian Process: $y(\boldsymbol{x}) = f_{\text{GPR}}(\boldsymbol{x}) + h(\boldsymbol{x})^\top \beta$. Furthermore, the preprocessed data are whitened, meaning rescaled to have zero mean $\mu_X = 0$ and unit standard deviation $\sigma_X = 1$. The GPR training is done in this space, which is defined by the composition of the two affine transformations of linear trend removal and whitening.

For different scenarios, surrogates for different data dimensionality were used, which include parametric spaces of one dimension (∇T_i), two dimensions (additionally T_i or ∇T_e), four dimensions (additionally T_i, T_e, and ∇T_e) and six dimensions (additionally q, gm_3). Furthermore, the surrogates consider either a scalar QoI (Q_i) or a vector of length two as an output (additionally Q_e). The default version of the surrogate discussed in this work is a surrogate for $n_{\text{f.t.}} = 8$ independent and decorrelated flux tube locations, mapping a four-dimensional output $\boldsymbol{x} \in \mathbb{X} \subset \mathbb{R}^4$ onto a two-dimensional output $\boldsymbol{y} \in \mathbb{Y} \subset \mathbb{R}^2$, not considering correlation between the output components.

Some diagnostics of a GPR model behaviour include analysis of its hyperparameters and validation.

Generally, GPR's statistical modelling assumes that its hyperparameters could be random variables with a particular PDF. In one of the approaches, tuning GPR by maximising the likelihood of explaining observable data assumes finding deterministic optimal hyperparameter values. Most kernels have length scale parameter λ as the most descriptive one, and it is often used to describe how smooth the dependency is with respect to specific input parameters. The short length scales can also indicate that a given parameter is the most influential for the modelled dependency.

The validation in this work was typically performed by reserving 20% of the data set and computing validation loss. As the most important performance metric, this work uses the *coefficient of determination*:

$$R^2 = 1 - \frac{\sum_{i=1}^{n_{\text{test}}} \left(y_i^{\text{g.t.}} - y_i^{\text{surr}} \right)^2}{\sum_{i=1}^{n_{\text{test}}} \left(y_i^{\text{g.t.}} - \overline{y^{\text{g.t.}}} \right)^2} \tag{4.1}$$

Here $(\cdot)^{\text{g.t.}}$ superscript denote the ground truth values from the test data set, $(\cdot)^{\text{surr}}$ mean values produced with a surrogate and $\overline{(\cdot)}$ stands for averaging operation over the test data set. This quantity indicates how well the given model can explain the variation in the validation data set.

Furthermore, the machine learning model's response was analysed by per-case comparison against the ground truth validation dataset in terms of capturing the actual values within the model's uncertainty intervals as shown in figure 4.8.

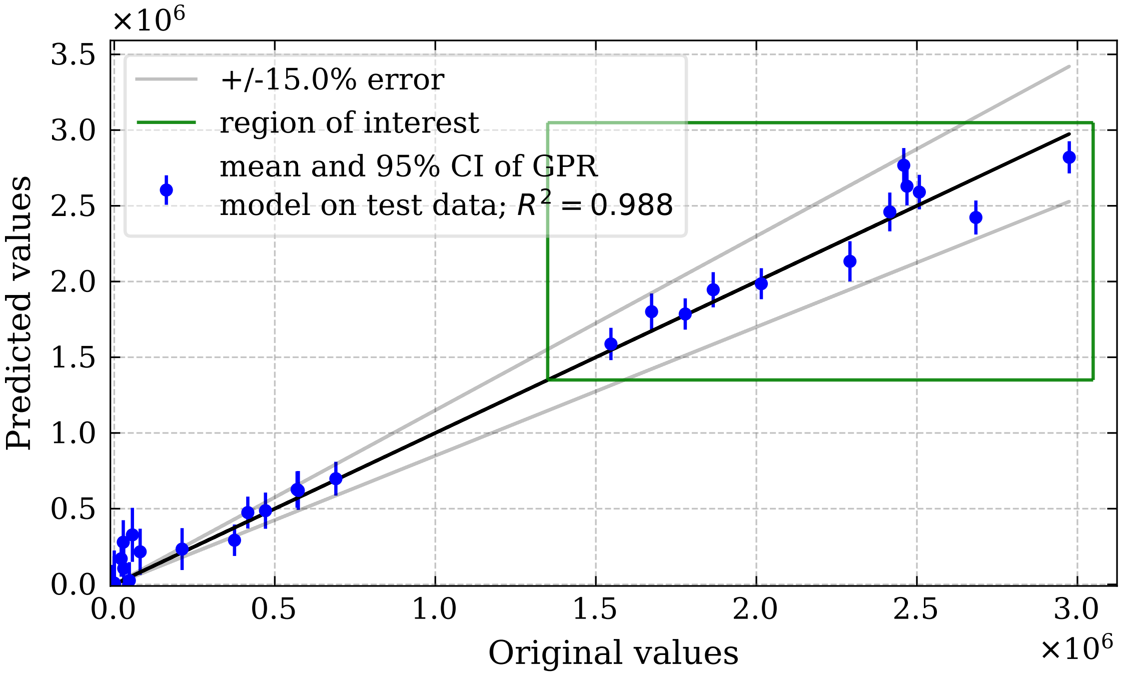

Figure 4.8.: The ground-truth flux values produced by GEM against the ones predicted by a GPR surrogate model for the validation dataset for a model flux tube. For most cases in the region of interest of plasma core profile states, defined by the MFW simulations, the true values lie in the $1.96\,\sigma$ uncertainty values (95% confidence intervals) and within 15% relative error bounds. The coefficient of determination value of $R^2 = 0.988$ indicates the high explainability quality of the surrogates.

A test of the continuous response of a GPR model for turbulent transport fluxes is shown in figure 4.9. Here, one can see the key properties of such a model, including the choice of length scale depending on the regularity of the dependency, uncertainty decorrelation due to lack of data, and high uncertainties in the extrapolation regimes. The surrogate finds very high length scales of dependency on ion temperature and its gradient $\lambda_{T_i}/\sigma[T_i] \approx 14$ and $\lambda_{\nabla T_i}/\sigma[\nabla T_i] \sim 10^6$ with a much higher uncertainty on the electron temperature and its gradient $\lambda_{T_e}/\sigma[T_e] \approx 0.13$ and $\lambda_{\nabla T_e}/\sigma[\nabla T_e] \approx 3.5$.

The dependency of turbulent fluxes on core profiles $\boldsymbol{q}(\boldsymbol{s}_{\text{c.p.}})$ computed by GEM has exposed more complexity and nonlinearity. These properties should lead to more complexity of the type of model used for a surrogate. For a non-parametric GPR model, it means searching through more complex kernel functions, more possible base regression models for mean trends, and more possible parameter space representations. Also, such a model could lead to less apparent results in the found model hyperparameters $\boldsymbol{\theta}$ like the length-scales of covariance for different parameters. The figure 4.10 shows central cuts of the GPR model fitting data from GEM. Here, we assume that the heat flux should be outwards, which makes us avoid situations with an inward pinch transport component $v_{\text{conv}} < 0$ and steep profiles. Moreover, it allowed us to work in the space of logarithmic flux values $y = \ln Q_{e,i}$, which is

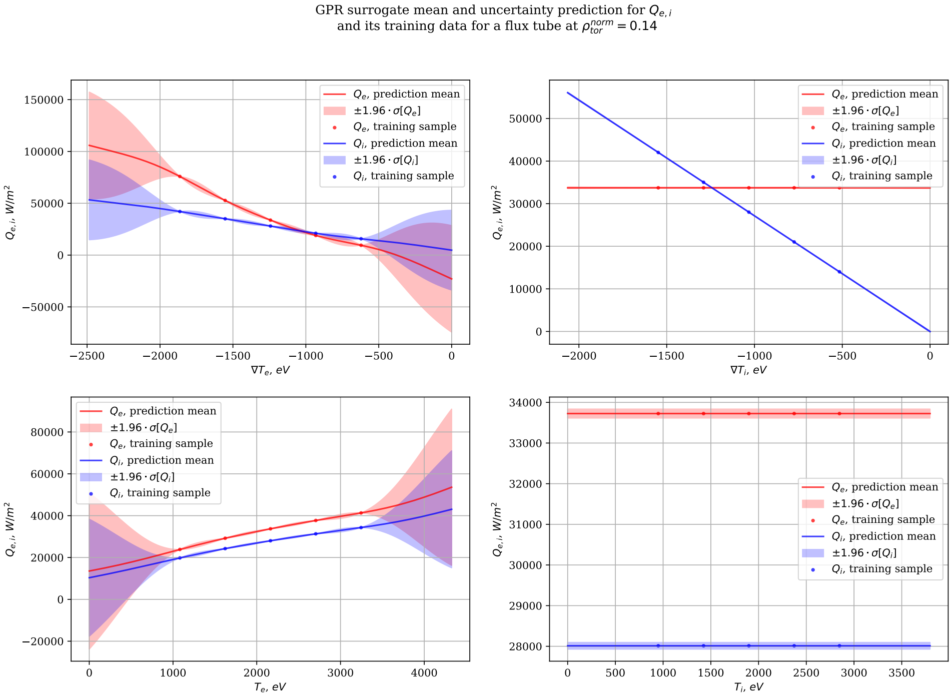

Figure 4.9.: The depiction of a GPR model response of Q_e,Q_i scan for only a single scalar out of T_e, T_i, ∇T_e, ∇T_i being varied. The model was trained on the sample of 5^4 evaluations of GEM0 code. Red denotes electron heat flux, and blue denotes the ion channel. Solid dotes indicate the training data points appearing in this cut of the parametric space. The solid lines and the shaded area denote the model prediction's mean and $\pm 1.96\sigma$. The surrogate has a shorter covariance length scale for more complex dependencies and quickly loses certainty outside its training domain. In the presented cases the lengthscales are likely to be defined by ion temperature profiles $L_\perp \sim a\frac{T_e}{|\nabla T_e|}$ given a baseline model for fluxes as $Q_i \sim T_e^{\frac{1}{2}} \nabla T_e \nabla T_i$ and $Q_e \sim T_e^{\frac{1}{2}} (\nabla T_e)^2$.

equivalent to the exponential model for the mean $Q_{e,i} = Q_{e,i}^{\text{exp}} \cdot e^{\sum_{i \in S} \lambda_i \cdot x_i}$, which is employed in the model in figure 4.10. A further assumption of monotonic profiles allows to work in space of negative logarithmic gradients $x = \ln(-\nabla T_{e,i})$ which is equivalent to the power mean model of $Q_{e,i} = Q_{e,i}^{\text{pow}} \cdot \prod_{i \in S} x_i^{\alpha_i}$. The next transformation of the parametric space is working with temperature ratios $\frac{T_e}{T_i}$ and relative or gradient logarithmic profiles $\ln \nabla T_{e,i}$.

The data acquisition from the high-fidelity GEM turbulence model is much more expensive, $\sim$ 5 orders of magnitude, compared to a simple analytical model. Thus, here, we used 3 points per dimension instead of 5, influencing the quality of surrogate models. Moreover, the GEM model exposes aleatoric uncertainty in the QoI, which is non-trivial to capture with a surrogate model, especially in the case of heteroscedastic uncertainty being itself a function of

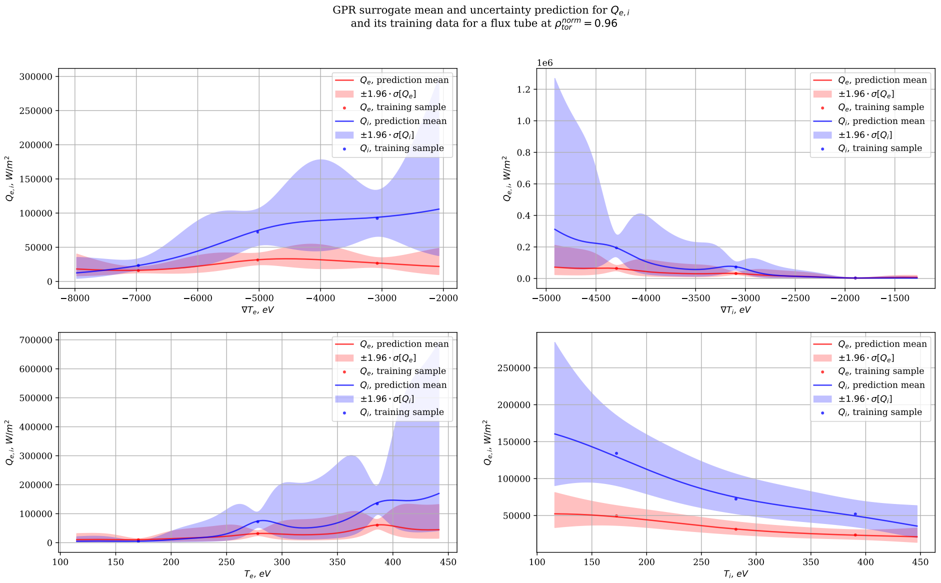

Figure 4.10.: The graph of a central cuts GPR model response of Q_e, Q_i as a function of T_e, T_i, ∇T_e, ∇T_i being varied. The model is trained using the 3^4 GEM evaluations data sample. The target quantity of training was $\ln Q_{e,i}$, which led to an approximately exponential dependency of the mean value, strictly positive values of flux covered only, a highly skewed confidence interval, and even faster growth of uncertainty outside of the learned domain.

independent parameters. The aspect of aleatoric dependency also raised the question of the quality of the training data in the sense of having enough data sample size and simulation duration to estimate the statistical quantities. Simulation convergence is especially important for the inner plasma regions where turbulence develops slower and requires more extended simulation runs to converge and produce a sufficiently large effective sample. The surrogates behave differently for different flux tubes at different values of $\rho_{\text{tor}}^{\text{norm}}$, which is apparent when they are used as a part of the workflow, which will be discussed in the later chapter.

Software Implementation

All the functionality related to the GPR surrogate was implemented into the `EasySurrogate` Python software library [136]. This includes obtaining data samples, both from physical code wrappers and the databases of UQ studies via EasyVVUQ, data preprocessing and feature engineering specific for GPRs and plasma turbulence models, surrogate training, its validation, hyper-parameter optimisation on an HPC machine using a combination of EasySurrogate, EasyVVUQ and QCG-PJ, surrogate model persistence and storage, regression model testing, Bayesian Optimisation with a data feedback loop to EasyVVUQ uncertainty quantification description.

4.3.2. Active Learning

In this section, we present the results of the Active Learning approach for surrogate training based on Bayesian Optimisation.

Figure 4.11 shows cases when a surrogate is trained on simulation data, acquired sequentially based on two types of acquisition functions that utilise uncertainties for a GPR model. A method based on the maximal variance only produces a space-filling set of points, whereas the technique that also minimises the distance to a flux (expected from a power balance calculation) captures a level set of fluxes. Both approaches require $\mathcal{O}(10)$ simulations for a two-dimensional surrogate.

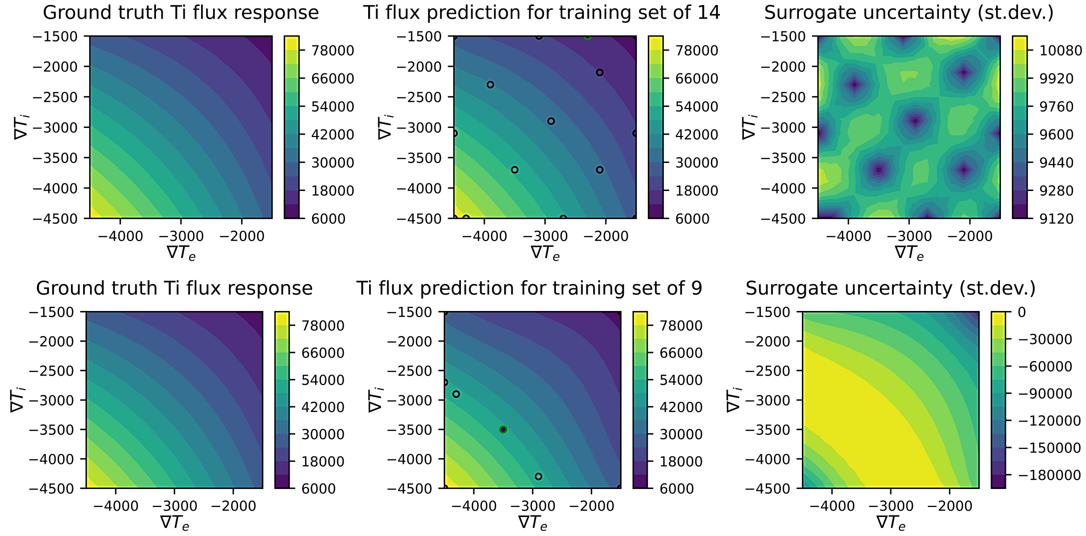

Figure 4.11.: The demonstration for sequential AL as a sequential design of experiments with BO. Here, we prepared a surrogate that finds ion heat flux Q_i at $\rho_{\text{tor}}^{\text{notm}} = 0.7$ based on the ion temperature and its gradient ∇T_e, ∇T_i using data from GEM0. The global variance acquisition function $x_{i+1} = \arg\min_X \sigma^2_{\text{GPR}}(x)$ produces a space covering data set and a surrogate of validation error RMSE ≈ 100 with 14 simulations. A more elaborated acquisition function targeting minimisation of distance to a target flux $y_* : \; x_{i+1} = \arg\min_X \frac{(f_m(x_*)-y_*)^2-(\mu_{\text{GPR}}(x)-y_*)^2+\epsilon}{\sigma_{\text{GPR}}(x)+\epsilon}$ convergence after 9 iterations.

4.3.3. Artificial Neural Networks

As an additional type of surrogate data-based model for this work, Artificial Neural Networks (ANNs) were used.

In their most common implementations, the inference or forward pass usually returns a deterministic vector value of length equal to the size of the last layer.

Direct usage of such models to assess the likelihood of a specific QoI outcome based on inputs requires the usage of a specific type of neural networks, like *BNNs*, or introducing some assumptions, like interpreting the value of a function with $\boldsymbol{y} \in [0.0, 1.0]$ range (logistic function and others) as probabilities. However, in most non-intrusive UQ approaches applied in this work, a surrogate was required only to produce a deterministic answer, so a wide variety of ANNs were applicable.

This work tested an ANN of a *Mutli-Layer Perceptron (MLP)* type or a *Feed-forward Neural Network (FFNN)* with various layer and neuron numbers and different activation functions.

In this case, the same validation approach, including the same validation metrics as for GPRs, was applied to ANN surrogates.

For this dataset size, the quality of regression using FFNN is comparable to that of a GPR model without the advantage of predicting entire likelihood distributions for QoIs.

Furthermore, ANNs have more free hyper-parameters to analyse and require a more extensive search in these parameters' space, taking care of overfitting, gradient vanishing and explosion, and other issues.

Figure 4.12 shows the central cut of the response of the ANN model trained with a data sample of flux values computed by GEM.

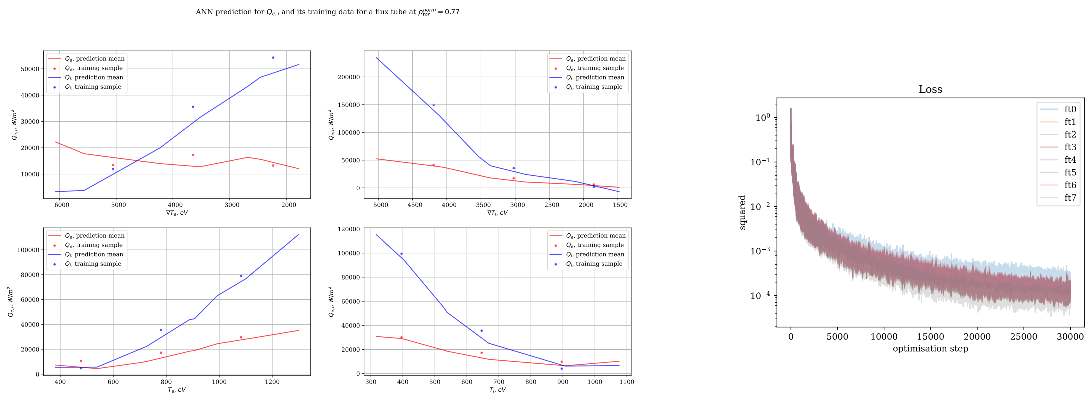

Figure 4.12.: The testing of ANN MLP model with 2 hidden layers 8 neurons each, with ReLU activation functions and trained to minimise L_2 loss function. The model is trained using 3^4 GEM evaluations. The differences in the loss curve during training indicate the complexity of the underlying dependency for various flux tubes.

The application of an ANN surrogate also required more careful analysis of the generalisation properties of the model as higher complexity of the model can lead to overfitting issues. Furthermore, it led to more considerations about the error analysis during learning and the correct duration of the training process.

4.4. Chapter Summary

In this chapter, we present the results of studies that involve the computation of transport fluxes with a high-fidelity turbulence code. This includes the study of its irreducible aleatoric uncertainty in fluctuating quantities, how to effectively decide on simulation duration, and the amount of data required for the estimates. Moreover, it covers the parametric study of the code and the quantification of epistemic parametric uncertainties, showing the level of the model's sensitivity towards uncertainty and quantitatively recovering the most important parameters influencing turbulent transport. Finally, we discuss the creation of surrogate models for such a turbulence code, as well as ways to diagnose their quality and dynamically utilise and update with new simulation data.

5. Multiscale Turbulent Transport Workflow Studies

This chapter presents the modifications to the Multiscale Fusion Workflow done to enable the usage of the surrogate for the turbulence model and to control the surrogate quality. Moreover, it presents a way to re-train surrogates using information on multiple scales in the learning loop. Finally, we present a forward uncertainty quantification study of the coupled transport-equilibrium-turbulence simulation workflow, focusing on how uncertainties in turbulent flux estimations due to their stochasticity influence the uncertainties in the tokamak plasma core kinetic profile prediction.

5.1. MFW Modifications

Due to the limited fidelity of any data regression-based surrogate, primarily due to extrapolation weakness, as well as to enable the idea of dynamical and physics-informed update of a surrogate, several modifications that combine different implementations of the expensive model for turbulent transport explaining the smallest scales are required and were implemented. The approach of combining different models explaining the same processes allows further blending information of different levels of fidelity for a single problem solution.

5.1.1. Introduction of New Components

In order to enable the usage of the surrogate models, the workflow was modified to have a flexible, dynamic choice of implementations of models providing transport information. Here, the “Turbulence” component was replaced by a subgraph tree of components. The element, which communicates with other codes, is Turbulence Manager, and it has ports allowing accepting *equilibrium* and *core profile* information and sending *core transport* data.

The other new components communicate only with the new manager component, and based on the information received during the iteration, the manager decides which component to run and what information to expose to other codes.

In basic implementation, the turbulence *manager* receives the updated equilibrium and core profile values and passes information to the surrogate component. Apart from the respective flux values, the surrogate provides the manager component with information on whether the received data were within the learned support of the surrogate and values of its uncertainty estimates in terms of its relative standard deviation. Based on this information, the manager component decides whether to perform the respective flux value calculations via a higher fidelity numerical code. As a general check, the manager decides to run a full-pledged turbulence simulation if the core profile values are simply outside the margins of the surrogate training data set, information on which is stored by workflow and accessible by the surrogate component on any iteration. The modified workflow is depicted in figure 5.3. Every workflow iteration is then modified according to the following steps in algorithm 1:

Algorithm 1 Turbulence Manager extension logic

```
...
receive $\boldsymbol{s}_{\text{c.p.}}$ and $\boldsymbol{s}_{\text{eq}}$
send $\boldsymbol{s}_{\text{c.p.}}$ to $\text{TURB}_{\text{surr}}(\cdot)$
receive fluxes $\boldsymbol{q}$ and fidelity information $\mathbf{fid}_{\boldsymbol{q}}$    ▷ Here: fidelity information is surrogate learned domain extrema $\mathbf{S}^{\text{surr}}_{\text{c.p.}}$ and the surrogate uncertainty $\sigma^2_{\text{surr}}$
if fidelity information $\mathbf{fid}_{\boldsymbol{q}} \notin \mathbf{FID}_{\boldsymbol{q}}$ trusted fidelity region then    ▷ Here: whether the input core profiles $\boldsymbol{s}_{\text{c.p.}} \notin \mathbf{S}^{\text{surr}}_{\text{c.p.}}$ outside of surrogate domain
    send $\boldsymbol{s}_{\text{c.p.}}$ and $\boldsymbol{s}_{\text{eq}}$ to $\text{TURB}_{\text{hi-fi}}(\cdot)$
    receive $\boldsymbol{q}$
end if
send flux values $\boldsymbol{q}$ out
...
```

The fidelity information could use detailed information on the learned domain. Also, the decision criterion can include information on the distance to the boundary of the learned domain, the relative or absolute size of surrogate uncertainty, the combination of those criteria for multiple flux tubes or sub-models, including logical `OR`, `AND`, majority or unanimous voting or other ways to combine criteria.

5.1.2. Profile Trajectory Tracking and Surrogate Fidelity Domain

Furthermore, the workflow tracks whether the trajectory of the core profile evolution stays within the surrogate applicability bounds. This tracking functionality is used to suggest which region of core profile parameter space the surrogate training dataset should be expanded into. Figure 5.1 shows a run of an MFW workflow with a GPR surrogate for turbulent transport while the profile values for a given flux tube stay within the domain learned by the surrogate model.

This *tracking* functionality is based on the explicit knowledge of the surrogate module on its training data distribution. From the point of view of the workflow's current software implementation, this requires specifying a particular file from which a surrogate component computes the hypercube envelope of its fidelity domain $\mathbf{FID}_{\boldsymbol{q}}$, serving as a bound of the training data set. A future extension for a more precise description of training data distribution is also possible. Each iteration involving surrogate inference is enhanced with an additional check on the surrogate fidelity in terms of whether the incoming input parameters are inside the fidelity domain $\boldsymbol{s} \in \mathbf{FID}_{\boldsymbol{q}}$. Here, the input parameters $\boldsymbol{s}$ are ion and electron temperature values and their gradients at the chosen flux tube locations $T_e(\rho)$, $T_i(\rho)$, $\nabla T_e(\rho)$, $\nabla T_i(\rho)$. The variation of a fidelity criterion based on the epistemic uncertainty of the surrogate was also

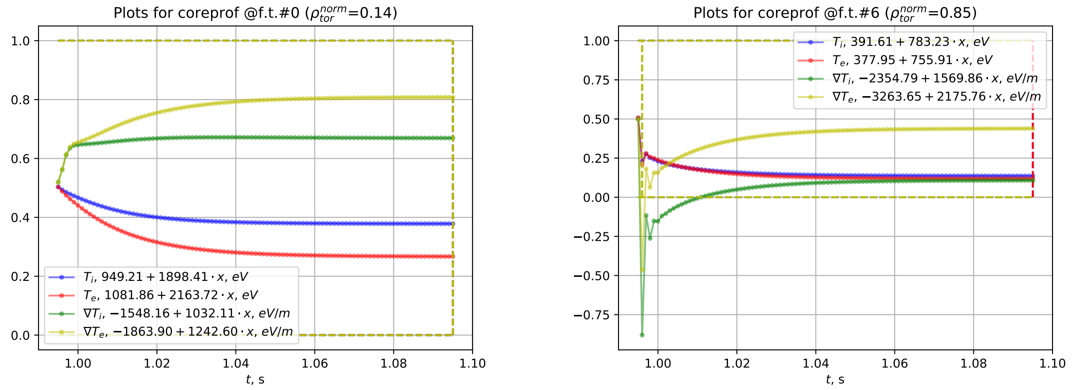

Figure 5.1.: Time evolution of $T_{e,i}$ and $\nabla T_{e,i}$ during an MFW simulation of AUG with quantities renormalised to fit $[0, 1]$ interval corresponding to the extreme input values of a surrogate training dataset. The expression in the legend shows the transformation of the quantity values back to the SI units, indicating the actual values of the surrogate fidelity domain boundaries. The workflow uses a surrogate based on GEM0 data to infer turbulent fluxes and the workflow convergence to a steady state within the 1% error relative to the ground-truth one after ~ 60 iterations. Here, the plot on the left demonstrates a benign case for the innermost flux tube at $\rho_{\text{tor}}^{\text{norm}} \approx 0.14$ when a surrogate can be trusted during the whole workflow simulation. The plot on the right demonstrates the profile evolution for an outer flux tube located at $\rho_{\text{tor}}^{\text{norm}} \approx 0.85$, with a rapid change in the gradient values at the beginning of the simulation, and a return in the fidelity region later on.

implemented. However, this method is more difficult to interpret due to different sources of uncertainty and complicated generalisation for other surrogate model types. The latest workflow version was left with the explicit boundary check as criteria for surrogate fidelity and as the prevention measure against extrapolation.

On every iteration, for the incoming surrogate input parameters describing plasma state $\boldsymbol{s}(t)$, the turbulence component *TURB* determines the point $\boldsymbol{s}_{\text{inter}}$ where the trajectory of evolving scalar quantities extracted from the current `coreprof` intersects the surface of the surrogate fidelity hypercube $\mathbf{FID}_{\boldsymbol{q}}$. For this point $\boldsymbol{s}_{\text{inter}}$ it creates a suggestion of new parametric points $\{\boldsymbol{s}_{\text{sug}}\}$ for which new turbulence simulations should be performed to enhance the turbulence surrogate quality.

One might also choose to stop the workflow simulation due to losing trust in the consistency of future turbulent flux inference. However, in the working version of the workflow, this halting was not used because it was observed that there were cases when the system would return to the fidelity region. Such an approach revealed that the surrogate fidelity differs for different flux tube locations with respect to its ability to capture relevant plasma evolution and the convergence towards self-consistent kinetic profiles.

For the cases based on the GEM0 turbulence model, the surrogate stays more accurate in the inner parts of the tokamak plasma, smoothly converging to the steady state. In contrast, the outer regions of the core plasma tended to rapid changes at the beginning of the simulations, with possible temporary leaving of the fidelity domain. However, the experience with the GEM model sometimes showed the reverse of the trend concerning the easiness of capturing turbulence in the inner and outer regions of a tokamak plasma. The flux tube difference highlights the importance of the initial experimental design for turbulence simulations when preparing data for a surrogate and choosing the representative initial conditions, e.g., initial core kinetic profiles, for turbulent transport simulations.

The simulations of the recent MFW version using a GPR surrogate based on log-fluxes computed by GEM show that the collected data is insufficient to simulate quasi-stationary AUG profiles. Figure 5.2 shows that at some point in the simulation, workflow exposes unrealistic from the physics point of view behaviour, with profiles flattening in the inner parts of the tokamak core plasma. The flattening happens likely due to the poor capturing of the turbulent behaviour of plasma due to the slow growth and saturation of turbulence for innermost flux tubes and due to the small range of temperature gradient values presented in the training data for the intermediate values of the radial coordinate. The outermost parts of the core plasma, for $\rho_{\mathrm{tor}}^{\mathrm{norm}} > 0.4$, show a reasonable behaviour of core profiles.

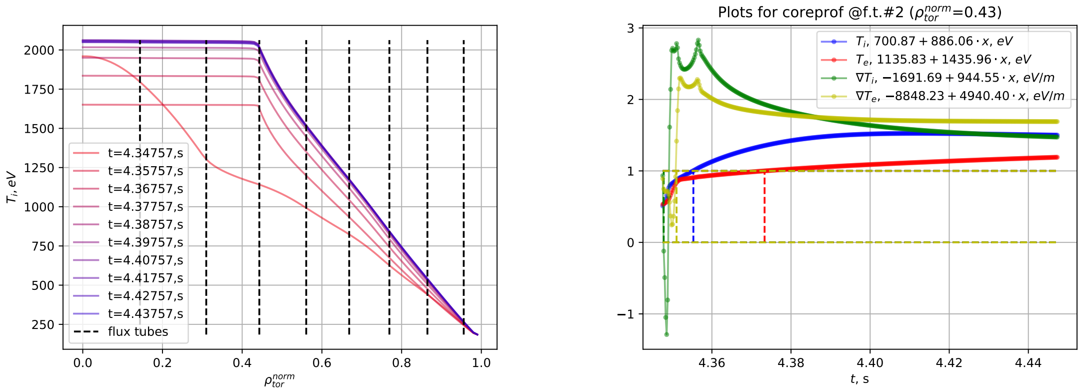

Figure 5.2.: The plot on the left shows the intermediate results of ion temperature evolution by the MFW using a surrogate for GEM. The flattening of profiles in the plasma core indicates that surrogates do not capture physics correctly in this region and require more data. The overall behaviour of profiles in the outer regions is less problematic compared to the inner core. The plot on the right shows the evolution trajectory of temperatures and their gradients for the third flux tube counting from the tokamak axis. Starting from the centre of the surrogate fidelity region, the parameters quickly leave the learned domain and converge on the values outside of the surrogate's area of trust, which can cause unphysical behaviour of the workflow deeper in the core.

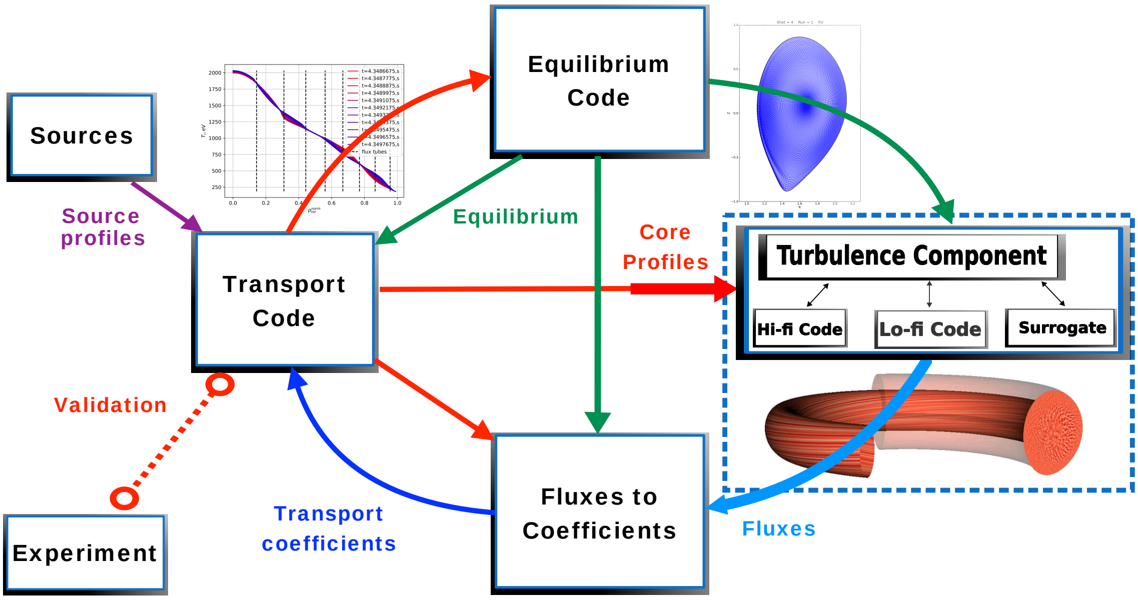

Figure 5.3.: The workflow variant using the Turbulence Manager configuration. This component abstracts out the implementation of the turbulence model, providing standard ports for core profile, equilibrium, and fluxes data. At the same time, it manages communication with existing turbulence models, all with their standard ports. It tries to compute the fluxes based on the least accurate model, which is the surrogate one. Also, it uses information on its fidelity level, namely the operational parametric domain and uncertainties, to decide whether the QoI value should be trusted or whether a more expensive model should be used.

5.1.3. Computational Cost and Other Studies

Introducing a surrogate as a model for turbulent transport significantly decreases the cost of performing a simulation of a coupled workflow and changes the cost composition. As the estimation of turbulent flux values becomes cheap, as shown in figure 5.4, the dominant fraction of the computational cost is shifted to the equilibrium computation.

As the equilibrium quantities and the geometry of the magnetic field change only within small bounds (for $\sim 1\%$ of nominal values), this cost could be further reduced by fixing the equilibrium state $\boldsymbol{s}_{\text{eq}}$ after several ($\sim 2-3$) iterations of the workflow, which would introduce a finite error to the final core profiles state $\boldsymbol{s}_{\text{c.p}}^{\text{fin}}$. However, a preferable way to reduce the cost for the equilibrium calculation would be creating a surrogate for the *EQUIL* model as a function of profile state $\boldsymbol{s}_{\text{c.p}}$, which requires capturing dependency on a large number of DoFs describing the equilibrium and performing a large number of simulations on itself.

An uncertainty propagation study for the uncertain independent parameters of the workflow, like the parameterisation of sources or boundary conditions influencing the core profiles, was previously conducted using simple deterministic turbulence models [137].

Addressing the absence of aleatoric uncertainty analysis and following the idea of analysing

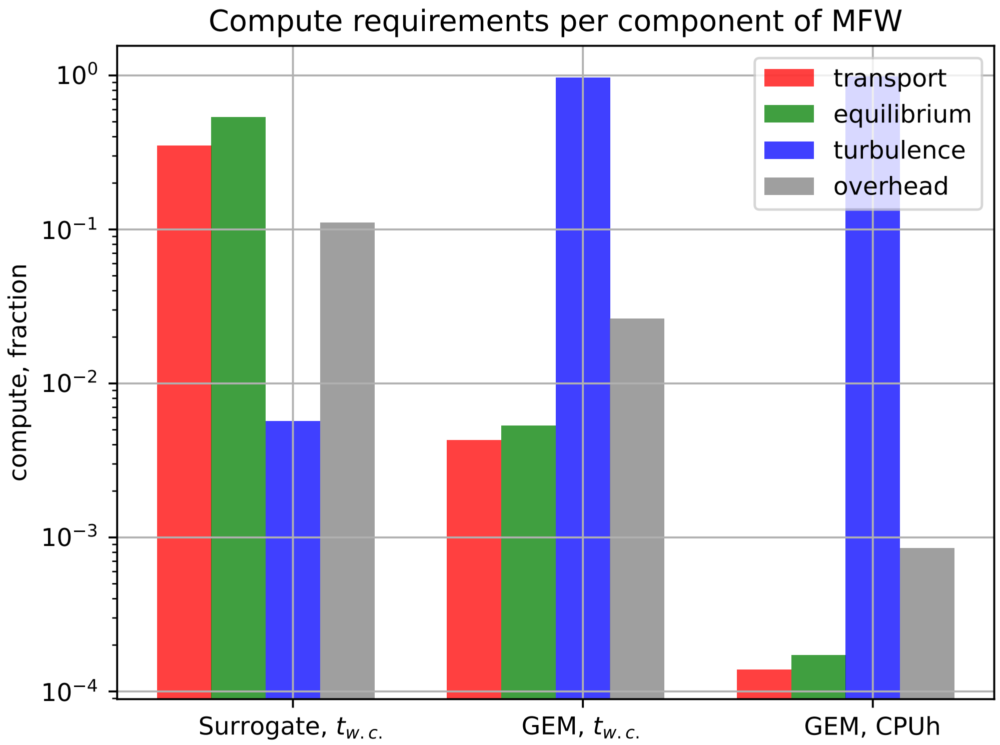

Figure 5.4.: The fraction of computational resources required for different components of the MFW. The first group shows the wall-clock time spent per component in a version of workflow using a surrogate for turbulent fluxes inference. The second group shows the wall-clock time for the workflow employing GEM code. The third group shows the same workflow version, but resources are measured in CPU time.

ensembles of dynamically evolving MFW simulations, the work presents a procedure to quantify such aleatoric uncertainties influencing global kinetic quantities. The key idea of this approach is to utilise variations of data to train different surrogate models and capture aleatoric uncertainty.

5.2. Simulation-coupled Surrogate Retraining Workflow

The other approach to make a surrogate capture the transport behaviour for the relevant distributions of independent variables it depends on is assuring it is accurate enough around the quasi-steady-state solution of the simulation workflow by iteratively adding new data around the assumed steady state.

The iteration proposed in this type of workflow consists of several steps described in the listing 5.2:

1. Prepare an initial guess for the state of the plasma $\boldsymbol{s}_0 = \left(\text{coreprof}_0, \text{equilibrium}_0\right)$ with some uncertainty $p_0(\boldsymbol{s})$
2. Produce a data set $D_0 = (\boldsymbol{S}, \boldsymbol{Q})$ of turbulent transport flux function evaluations (using a turbulence code) $\boldsymbol{q} = \boldsymbol{q}(\boldsymbol{s})$ to capture uncertainty $p_0(\boldsymbol{s})$: sample $\{\boldsymbol{s}_k\} \sim p_0(\boldsymbol{s})$ with a Monte-Carlo scheme or using a polynomial quadrature
3. Train a surrogate $q_0^*(\boldsymbol{s}|D_0)$ using the training data D_0
4. Run a workflow with transport parametrisation using $q_0^*(\boldsymbol{s})$ until some convergence criterion $\textbf{crit}_{\text{WF}}$ to solve for a stationary solution with a plasma state $\boldsymbol{s}_{i=1}$ (estimate the new state uncertainty $p_{i=1}(\boldsymbol{s}_{i=1})$)
5. Use $p_{i=1}(\boldsymbol{s}_{\boldsymbol{i=1}})$ to create a new data set $D_{i=1}$ and train a new surrogate $q_{i=1}^*(\boldsymbol{s}|D_{i=1}^+)$ on an expanded dataset $D_1^+ := \bigcup_{i<1} D_i$
6. Repeat the workflow run with a new transport parametrisation $q_{i=1}^*(\boldsymbol{s})$ and get a new candidate for a stationary state of the system $\boldsymbol{s}_{i=2}$
7. Compare the states of $\boldsymbol{s}_i$ and $\boldsymbol{s}_{i+1}$ (for $i = 1$) according to some metric for plasma states $d_s(\boldsymbol{s}_k, \boldsymbol{s}_l)$
8. Repeat steps 5-7 until the some convergence criterion $\textbf{crit}_{\boldsymbol{s}}$ for $d_s(\boldsymbol{s}_i, \boldsymbol{s}_{i+1})$

In such a fashion, the surrogate is be refitted with more data on every iteration, with the new data coming close to the candidate stationary states of the plasma for the model described by the workflow.

Given a sufficiently regular problem in the vicinity of an attractor described by the stationary state for the transport parametrisation surrogate tries to learn, on every iteration both the surrogate for transport flux function and the stationary state should be closer to the one described by the original turbulent transport description with a high-fidelity code.

5.2.1. Computational Complexity Considerations

Such a re-training loop saves computing time (in the first place, in terms of a number of turbulence transport model solutions) if only a few re-training iterations are needed.

One of the ways to view this algorithm is that it utilises the simulation $WF(\cdot)$ that changes the state of the system (parametrised via a number of scalars, vectors, fields) $\boldsymbol{s}_{\text{init}}$ to $\boldsymbol{s}_{\text{fin}}$ to determine an essential direction, or a basis state, of state parameter space $\Delta\boldsymbol{s}^* = \boldsymbol{s}_{\text{fin}} - \boldsymbol{s}_{\text{init}}$.

This idea allows talking about a vector space of plasma state evolutions $\Delta\mathbb{S}$ to which such $\Delta\boldsymbol{s}$ would belong. Such a view would be helpful when describing the behaviour of plasma states near a single stationary self-consistent state $\boldsymbol{s}^*$. This space could also be defined as a space of endomorphisms on the plasma state space $\Delta\mathbb{S} \ni \Delta\boldsymbol{s} : \mathbb{S} \rightarrow \mathbb{S}$. Moreover, given the continuous nature of evolution $\Delta\boldsymbol{s}$ in the course of PDE time integration, it is a state of diffeomorphisms. For a space of states $\mathbb{S}$, this would serve as a fixed point for evolution mapping of the introduced type $WF(\cdot)$. Stating distance from the stationary state $\boldsymbol{s}^*$ to an arbitrary state in the domain, hence, is a helpful definition of a norm for a plasma state $\|\boldsymbol{s}\|_{\Delta\mathbb{S}} = d(\boldsymbol{s}^*, \boldsymbol{s})$ in the state space.

Furthermore, such a $\Delta\boldsymbol{s}^*$ may be considered important if the system's evolution is sufficiently regular. If the conditions allow for such an interpretation of $\Delta\boldsymbol{s}^*$, it could be used to cast the epistemic uncertainty of the plasma state $p_{\text{ep}}(\boldsymbol{s})$ not in terms of uncertainty over the space state, but in terms of possible perturbations of the given state. This view allows using a set of $\Delta\boldsymbol{s}$ as a basis element to generate data sets of low dimensionality, which the surrogate or simulation model has to capture.

Each of the algorithm's iterations would introduce comparatively little change to the evolution dynamics, and hence, the set of vectors $\Delta\boldsymbol{s}_i$ would be rather colinear with respect to cosine-like metrics. Hence, such an algorithm is expected to bring the most of the practical improvement to the surrogate in its first single or several iterations.

With this, one could compare creating a data set of model solution evaluation by varying every input component of $\boldsymbol{s}_{\text{init}}$ and yielding effective dimensionality d_1 of the surrogate with varying $\boldsymbol{s}_{\text{init}}$ in a changed basis, only along new components proportional to a set of basic $\Delta\boldsymbol{s}_i^*$ with a smaller effective dimensionality of $d_2 \ll d_1$. Here, since a surrogate is a function of a data set of function evaluations D, and for a vast majority of non-parametric methods that explicitly use function evaluations as basis elements, the dimensionality of a surrogate is proportional to the cardinality of the training dataset $d \sim |D|$.

Thus, such an algorithm would be computationally profitable if it can converge to a surrogate that can capture the behaviour of the parametrised function sufficiently well around the stationary state of the system, at least as well as the one using a surrogate of size d_2, for a given domain in such a number of iterations n_{it} such that $n_{\text{it}} \cdot d_2 < d_1$.

5.2.2. Convergence for Different Scenarios

We would like to present the run of such an algorithm performed for the initial state being core profiles and magnetic equilibrium of AUG shot #28906. The system is being evolved using the previously described configuration of MFW, using analytical approximation code GEM0 as parametrisation of turbulent transport. The choice of turbulence model is defined by the possibility to demonstrate some core prototype ideas with compute requiring to compute of the order of $\mathcal{O}(10^2)$ CPUh instead of $\mathcal{O}(10^{4-6})$ CPUh for a high fidelity PDE solver. Worth noting that the initial magnetic equilibrium state $\boldsymbol{s}_{\text{init}}^{\text{eq}}$ is in practice made consistent by running a single iteration of equilibrium code $EQUIL(\cdot)$ for a given core profile component $\boldsymbol{s}_{\text{init}}^{\text{c.p.}}$ of the initial step.

The metric that is used to describe the distance between two plasma states $\boldsymbol{s}_1, \boldsymbol{s}_2 \in \mathbb{S}$ is the mean of *Symmetrised Relative Root Mean Square Error (srRMSE)* for core temperatures and its gradients $x \in \mathbb{X}_S = \{T_i, T_e, \nabla T_i, \nabla T_e\}$ taken at the $\rho_{\text{tor}}^{\text{norm}}$ values for the 8 flux tube locations P:

$$d_{\text{srRMSE}}(\boldsymbol{s}_i, \boldsymbol{s}_j) = \frac{1}{|\mathbb{X}_S|} \sum_{x \in \mathbb{X}_S} \sqrt{\frac{4}{|P|} \sum_{\rho \in P} \left(\frac{x_i(\rho) - x_j(\rho)}{|x_i(\rho)| + |x_j(\rho)|} \right)^2} \tag{5.1}$$

Here, for every iteration of the retraining workflow introduced previously, we calculate the d_{srRMSE} between the final stationary state achieved using the turbulence code as transport parametrisation and the workflow's micromodel $\boldsymbol{s}_{\text{fin}}^{\text{g.t.}}$, denoted as the ground truth one as it is the highest fidelity we can achieve using the simulation workflow, and the final state achieved by the surrogate on this iteration $\boldsymbol{s}_{\text{fin}}^{i}$: $d(\boldsymbol{s}_{\text{fin}}^{\text{g.t.}}, \boldsymbol{s}_{\text{fin}}^{i})$. This quantity is a measure of the quality of the surrogate model as a proxy in the simulation workflow and is supposed to asymptotically converge to a finite small value. Furthermore, we measure the difference between the final states between this and the previous iteration $d(\boldsymbol{s}_{\text{fin}}^{i-1}, \boldsymbol{s}_{\text{fin}}^{i})$, which is the quantity that is supposed to asymptotically converge to zero, if there is a fixed point best surrogate model for such an algorithm. Lastly, we measure how much a simulation changes the state of the plasma between the initial state and the final state $d(\boldsymbol{s}_{\text{fin}}^{\text{i}}, \boldsymbol{s}_{\text{init}})$ to have an idea of the scale of the metric. The following metrics convergence is summarised in the table 5.1.

Table 5.1.: The results of surrogate retraining algorithm. Each row indicates a procedure iteration consisting of data preparation, surrogate training, workflow simulation, and results. The algorithm was applied to capture stationary profiles for GEM0 model. The first column indicates the distance between the final plasma state computed by a workflow using a surrogate and the corresponding state computed by using the underlying code. The right column indicates the distance between the final states of workflow achieved by consecutive iterations of surrogate retraining. The $(2, 2)$ cell of the table is the scale of metrics for the surrogate improvement, and the $(1, 1)$ cell shows how much the workflow evolves the state in a single simulation.

Iteration#	$d(\boldsymbol{s}_{\text{fin}}^{\text{g.t.}}, \boldsymbol{s}_{\text{fin}}^{i})$	$d(\boldsymbol{s}_{\text{fin}}^{i-1}, \boldsymbol{s}_{\text{fin}}^{i})$
0	176495.9	—
1	603.41	177104.4
2	396.99422	277.11
3	396.90099	1.0399
4	396.90088	0.010746
5	396.90089	0.00034825

For the given case, indeed the convergence criterion $d(\boldsymbol{s}_{\text{fin}}^{\text{i}-1}, \boldsymbol{s}_{\text{fin}}^{i})$ reach a small value of $\mathcal{O}(10^{-5})$ after 2 iterations. For the same 2 iterations the surrogate quality $d(\boldsymbol{s}_{\text{fin}}^{\text{g.t.}}, \boldsymbol{s}_{\text{fin}}^{i})$ converges (up to the 5-th significant digit) to $\mathcal{O}(10^2)$, which is a value $\mathcal{O}(10^3)$ smaller than how much the state changes during the iteration. For some applications critical to accuracy, one can reduce this finite error by performing a simulation workflow with a transport parametrisation for a few steps.

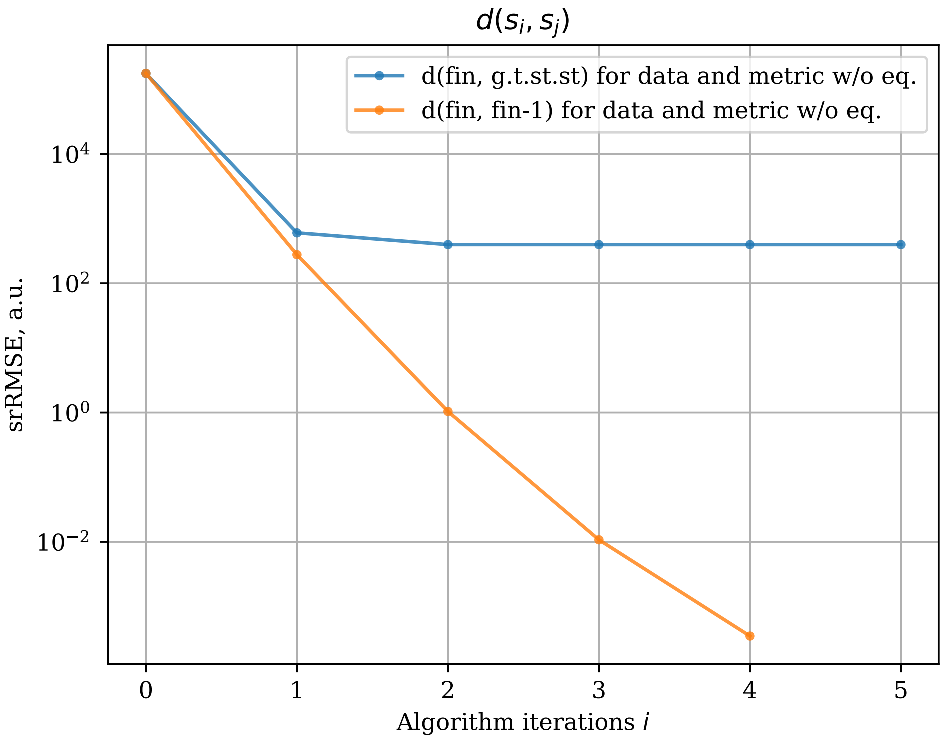

Figure 5.5.: A convergence plot of the algorithm that iteratively retrains a surrogate based on the stationary state produced by simulation workflow using this surrogate as the transport model. The metrics between two states is srRMSE described in 5.2.2, The distance between the simulation final stationary states for two consecutive retraining iterations falls exponentially with $\mathcal{O}(10^{-2n})$. However, the algorithm does not converge to the surrogate that would produce a final stationary state equivalent to the one while using the ground-truth model, but it has a finite systematic bias.

The systematic discrepancy for the final plasma states obtained by using code and its surrogate, however, is by all practical means negligible for the given metric value of $d_{\text{srRMSE}}(\boldsymbol{s}^{\text{code}}_{\text{fin}}, \boldsymbol{s}^{\text{surr}}_{\text{fin}}) \approx 396.9$, as shown in the figure 5.6.

Since the original procedure of surrogate training considers uncertainty and variation only of core profiles $\boldsymbol{s}^{\text{c.p.}}$, but not the equilibrium $\boldsymbol{s}^{\text{eq}}$, to create a data set for a surrogate, such a quick convergence additionally supports the idea that magnetic equilibrium does not strongly influence the behaviour of turbulent transport model. However, the importance of equilibrium should be supported by introducing metrics between equilibrium components of the plasma state and, ideally, by studying uncertainty in equilibrium $p(\boldsymbol{s}^{\text{eq}})$ and sensitivity indices of turbulent transport fluxes with respect to it.

An additional test was conducted on the level of influence of equilibrium parameters on

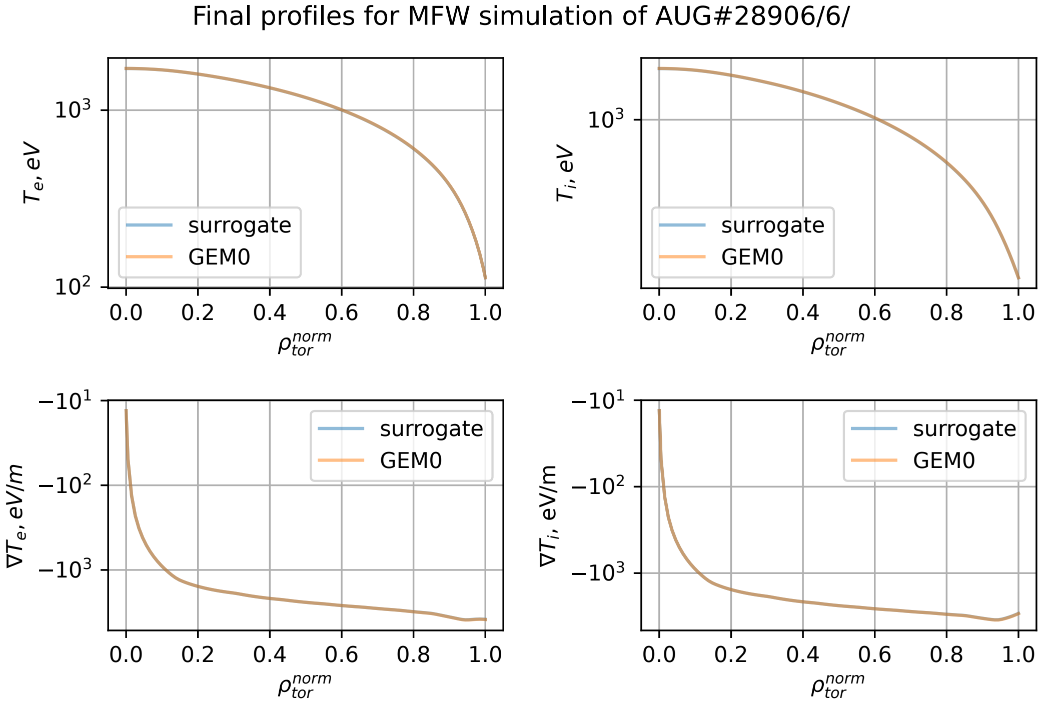

Figure 5.6.: Radial profiles for ion and electron temperatures as well as their gradients for the convergent stationary state of MFW using code and surrogate as turbulent transport model, are visually indistinguishable. The density gradients serve as a constant background input for the workflow and are not being evolved.

the effective heat fluxes. Here, we considered the $1D$ equilibrium quantities, being a function of radial coordinate ρ but poloidally homogeneous, primarily the safety factor q and metrics scaling coefficient gm_3.

The consideration of the variable equilibrium quantities within the same data set but without the explicit dependency of the surrogate model on any equilibrium parameters means that the model has to account for the uncertainty due to a lack of equilibrium information and the noise or error introduced by throwing away equilibrium dependency and considering only the projection of the plasma state $\boldsymbol{s}$ on its core profile subspace $\mathbb{S}^{\mathrm{c.p}}$.

In this attempt to explain the inconsistency of the method to find stationary self-consistent profiles with surrogate retraining, we expanded the number of independent variables by adding q and gm_3 to the epistemic uncertainty description and to surrogate training dataset variation. Using these new quantities required adding these new terms to the d_{srRMSE} metrics definition by expanding the summation over new $\mathbb{X}_{\mathbb{S}} := \mathbb{X}_{\mathbb{S}} \cup \{q, \mathrm{gm}_3\}$, which transforms the scale of the metric and the norm, taken as a distance to the "ground true" state. Furthermore, since we use a full-tensor-product-based method to capture the profile uncertainty and generate new data for a surrogate, it exponentially increased the expense for single algorithm iteration in the number of turbulence model solution function evaluations. In our case, the data set grew 9 times with the addition of the equilibrium data.

The comparison of the surrogate retraining algorithm results, however, shows that adding data did not solve the inconsistency issue and improved the surrogate approximation power for the stationary plasma state by around 1% only, as seen in table 5.2.2. Both cases, with equilibrium data and without, reached saturation after a single iteration on decent for practical purposes level as shown in the figure 5.7. Furthermore, the algorithm with equilibrium data has a slower surrogate improvement rate per iteration of $\frac{d_n}{d_{n+1}} \sim \mathcal{O}(10^{r_{\text{eq}}})$, $r_{\text{eq}} \sim 1.67$ compared to previous $r \sim 1.78$. Overall, the inclusion of the equilibrium data to the metric definition only changed the convergence down from $r \sim 2.18$.

This observation leads to a conclusion that for the given turbulent transport model, GEM0, the influence of equilibrium is insignificant, and its consideration by a surrogate brings overhead costs much higher than possible benefits.

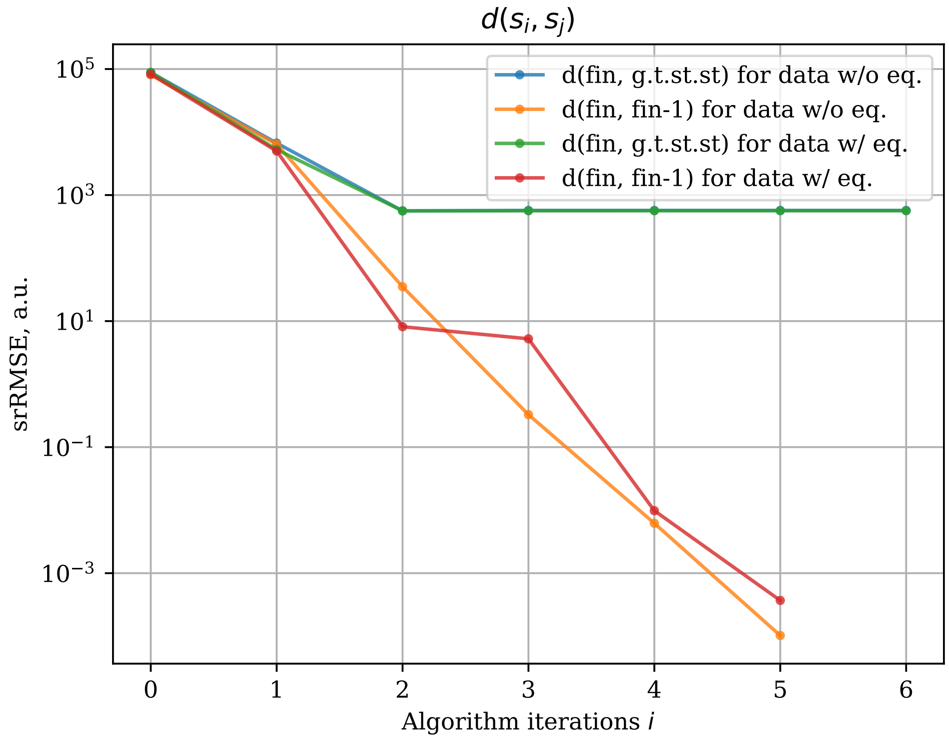

Figure 5.7.: Adding equilibrium data to the surrogate does not significantly change similar surrogate quality levels, saturation speeds, and algorithm convergence rates.

Table 5.2.: Comparison of retraining procedure results using surrogates that either explicitly include dependency on equilibrium or not.

Iteration#	$d(\boldsymbol{s}_{\text{fin}}^{\text{g.t.}}, \boldsymbol{s}_{\text{fin}}^{i})$	$d(\boldsymbol{s}_{\text{fin}}^{i-1}, \boldsymbol{s}_{\text{fin}}^{i})$
0	88251.68	—
1	6684.846	81224.63
2	561.1208	6415.358
3	569.631339	35.08944
4	569.664854	0.32789178
5	569.665844	0.0061919296
6	569.665872	0.0001025095

Iteration#	$d(\boldsymbol{s}_{\text{fin}}^{\text{g.t.}}, \boldsymbol{s}_{\text{fin}}^{i})$	$d(\boldsymbol{s}_{\text{fin}}^{i-1}, \boldsymbol{s}_{\text{fin}}^{i})$
0	88251.68	—
1	5277.631	82685.02
2	562.5339	4994.789
3	564.417244	8.14617533
4	564.945749	5.23832979
5	564.945318	0.009808316
6	564.944993	0.0003694049

The figure 5.8 shows the distance between the stationary plasma state achieved with a workflow using a surrogate and the one using the code. It further shows which quantities change the most at which locations for a workflow simulation.

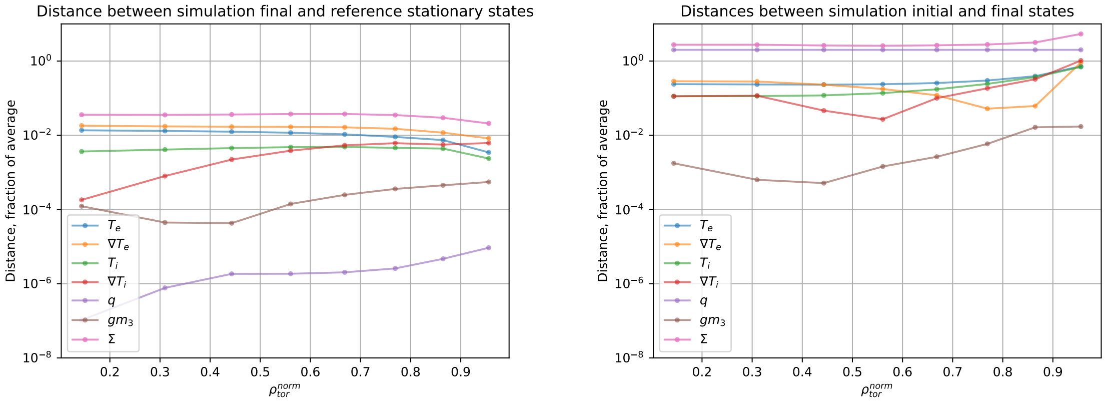

Figure 5.8.: The contribution of different quantities computed at different locations to the difference between plasma states. The first figure shows the distance between the final workflow states with surrogate and code as turbulent transport model. The highest contribution to the systematic error is due to the ∇T_e resolution. The second figure shows the evolution of quantities made during a workflow simulation. The temperatures evolve significantly compared to their gradients.

A further consideration is to improve the quality of a surrogate by getting rid of clusters of data set reading and leaving a training data set of constant size and via updating step 5 as $D_i^+ := D_i$, showed the best approximation of the stationary plasma states as shown in figure 5.9.

A potential remedy for the systematic error $\Delta \boldsymbol{s}_{\text{err}}$ in the algorithm would be fine-tuning the stationary profiles by performing a short simulation using a high-fidelity turbulent transport model starting with the candidate stationary profiles suggested by the algorithm $WF_{\text{hi-fi}}(\boldsymbol{s}_{\text{fin}})$.

Furthermore, in the current implementation of the algorithm, we consider a fixed epistemic uncertainty of the plasma state final for a simulation workflow $p_{\text{ep}}(\boldsymbol{s}_{\text{fin}})$. A further development would be to replace every simulation workflow with a number of those to capture the self-consistent uncertainty of the current state for every iteration $p_{\text{ep},i}(\boldsymbol{s}_{\text{fin}})$.

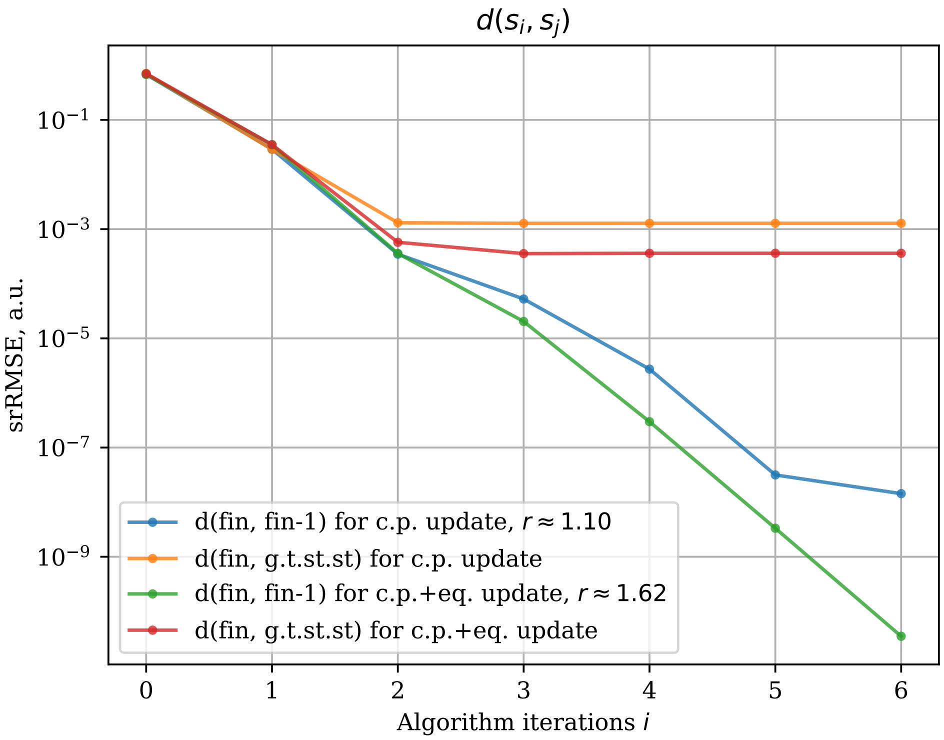

Figure 5.9.: Results of a retraining algorithm using only the latest dataset for surrogate training. In this case, the surrogate has no explicit dependency on equilibrium quantities. For core profile and equilibrium update on every step, the quality of the final surrogate is the best among all trials.

5.3. Aleatoric Uncertainty Influence in Coupled Simulations

One of the interesting questions that could be posed about the aleatoric uncertainties is how they, accounted in the solutions of some subproblem, usually on a microscale, could influence QoIs in the solution of coupled simulation on the largest scale.

5.3.1. Preliminary Steps for Aleatoric Analysis

Given that we consider the outcome of a micromodel solution to be probabilistic, there are several ways to incorporate this uncertainty analysis into the coupled system. The algorithm employed in this work assumes a separation of the measurement process of aleatoric uncertainty and its propagation. The first step is studying the model behaviour with respect to its aleatoric uncertainties for the parametric region of interest. Then, the procedure assumes the aleatoric uncertainty can be captured via a surrogate or a number of surrogate models. For this procedure, we employ an ensemble of surrogates, each trained on perturbed data capturing aleatoric uncertainties. The final step is using the knowledge of the aleatoric uncertainty to propagate it across the multi-scale model and measure its influence on the global quantities of interest.

5.3.2. Algorithm for Analysis of Turbulence Model Aleatoric Uncertainties on Global Plasma Parameters

In this part of the work, we suggest using a surrogate approach to capture aleatoric uncertainties since this scenario requires a large number of micromodel solutions due to multiple time iterations of the coupled system solution and sampling from an aleatoric uncertainty PDF.

One of the approaches would be to utilise a probabilistic surrogate for the micro-model and sampling from its trained PDF whenever the evaluation of this model solution is required.

The other approach would be using an ensemble of surrogates, each providing mean values for the solution.

Here we suggest and approach to first analyse the aleatoric uncertainties $p_{\mathrm{al}}(\boldsymbol{q})$ of the underlying model and sample possible QoI values $\boldsymbol{q}_i \sim p_{\mathrm{al}}(\boldsymbol{q})$ to create a set of possible model behaviours for some set of independent variables values $\{\boldsymbol{s}_j\}$ coming outside of the micromodel. This employs a set of datasets $\{(\boldsymbol{S}, \boldsymbol{D}_i)\} = \{(\{\boldsymbol{s}_j\}, \{\boldsymbol{q}^i(\boldsymbol{s}_j)\})\}$. Having such an ensemble of possible micromodel evaluations, perturbed from its mean behaviour, each represented by its own surrogate, one could run an ensemble of coupled simulation instances, and then analyse statistics in global solution QoI due to aleatoric uncertainties.

This procedure requires having aleatoric uncertainties $p_{\mathrm{al}}(\boldsymbol{q}(\boldsymbol{s}))$ analysed beforehand, which is, in principle, possible due to the scale separation principle. The preliminary aleatoric uncertainty analysis can be done by an analysis of independent micromodel aleatoric uncertainties for a set of independent variable values, via fitting parameters of a $p_{\mathrm{al}}(\boldsymbol{q}(\boldsymbol{s}))$ from some PDF family, and then $p_{\mathrm{al}}(\boldsymbol{q})$ generalised for any $\boldsymbol{s}$.

In this work, we fit such an aleatoric uncertainty for heat flux values $Q(\rho)$ as a function of kinetic profiles and magnetic equilibrium as Normal distribution, requiring specification only of its mean $\mu[\boldsymbol{q}(\boldsymbol{s})]$ and standard deviation $\sigma[\boldsymbol{q}(\boldsymbol{s})]$.

Procedure Steps

The following steps describe the procedure implemented in this work to analyse the influence of aleatoric uncertainties of the turbulence model in the value of effective radial outward turbulent transport heat fluxes $Q(\rho)$ due to its local fluctuations in time on the quasi-stationary profiles of the transport simulations using this turbulence model as transport parametrisation. The steps of aleatoric analysis algorithm are described in listing 5.3.2:

1. Based on a finite set of plasma states $\{\boldsymbol{s}\}$ described with its kinetic profiles and magnetic equilibrium, realise a finite set of turbulence model solution function evaluations $\{\boldsymbol{q}(\boldsymbol{s})\}$

2. For every evaluation, estimate aleatoric uncertainty $p_{\mathrm{al}}(\boldsymbol{q}(\boldsymbol{s}))$ and form a generalised description for (marginal) aleatoric uncertainty $p_{\mathrm{al}}(\boldsymbol{q})$ (here: as a function of radial coordinate ρ but not plasma state $\boldsymbol{s}$)

3. Prepare a set of perturbations of turbulent flux behaviour $Q = \{q_j(\cdot)\}$, based on the general aleatoric uncertainty $q_j(\cdot) \sim p_{\text{al}}(\boldsymbol{q})$

4. For a plasma state of interest $\boldsymbol{s}^*$ and its epistemic uncertainty estimate $p_{\text{ep}}(\boldsymbol{s}^*)$, prepare a sample of perturbed plasma states $S = \{\boldsymbol{s}_i\}$

5. Based on a full (tensor) product of sets of perturbations of turbulence model behaviours and plasma states $Q \bigotimes S$ prepare a set of training datasets $\mathbb{D} = \{D_j\}$, $D_j = \left(Q_j, S\right)$, $Q_j = \{q_j(\boldsymbol{s}_i)\}$ of all the model perturbations applied to all state perturbations

6. Train an ensemble of surrogates $\mathbb{Q}^* = \{\boldsymbol{q}^*_j(\boldsymbol{s}|D_j)\}$ based on a set of different training datasets $\mathbb{D}$

7. Perform an ensemble of coupled transport-equilibrium-turbulence workflow runs, each with its own turbulent transport parametrisation from $\mathbb{Q}^*$, producing a sample of final plasma states $S_f = \{\boldsymbol{s}_{\text{fin}}\}$, each with its own QoI values. Here, the primary QoI analysed was ion temperature at the axis of a tokamak plasma $y = T_i(\rho^{\text{norm}}_{\text{tor}} = 0)$

8. Given the set of QoI values $\{y\}$ for the final plasma states S_f estimate its uncertainty $p(y)$ in terms of a finite number of central moments of it PDF. Here, we chose the first statistical moments of mean, variance, skewness, and kurtosis

Such an algorithm depends on an assumption of ergodicity of the transport flux fluctuations and on the existence of a correspondence of their distribution in time to a distribution over an ensemble of possible perturbations. With such an assumption, it is possible to interpret the algorithm's results as the spread of core profile values over time during a particular discharge.

Also, the existence of general $p_{\text{al}}(\boldsymbol{q})$ from step 2 is a strong assumption as, in a general case, we have to analyse such an aleatoric uncertainty as a function of $\boldsymbol{s}$.

Consequently, the product of two types of perturbation might not be applicable with a strong correlation of the aleatoric uncertainty of turbulent model behaviour on plasma states, and one should instead use a method to construct a dataset that accounts for this correlation.

Here, we calculate the core profile uncertainties with a Monte Carlo (MC) approach, for which we perform sampling from the turbulence heat flux uncertainty using a Saltelli scheme [49]. Due to the relatively small training dataset required for each of the surrogates and, subsequently, a low computational cost to create one, one could afford to have a large ensemble of surrogates, and MC integration is a suitable approach for such a situation.

Apart from providing a general tool for code behavior validation in terms of statistical tests employing both experimental and simulation uncertainties, such uncertainty for $p\left(T_{i,e}(\rho^{\text{norm}}_{\text{tor}})\right)$ can be used to answer more practical questions for the predictive simulations. For example, one can use it to answer questions on risks associated with a discharge like what is the probability that plasma will leave the previously established bound of stability $\mathbb{S}_{\text{stab}} \subset \mathbb{S}$: $P\left(\boldsymbol{s}(t) \notin \mathbb{S}_{\text{stab}}\right)$.

Moreover, one can propagate an uncertainty of this type to analyse two different types of parameters. The first one is meant to capture physical properties and can be measured via propagating a natural relative fluctuation of fluxes. This, with some assumptions on how easily fluctuations can propagate across scales, can serve as an estimate of the possible level of fluctuations of profiles on the macro level. The second approach is statistical and, being more sound, is meant to measure errors in a statistical sense. Propagation of error in the estimate of mean flux value through the simulation estimates the statistical error in the computation of core profiles due to the stochasticity of turbulence.

Algorithm Parameters and Run Results

The $p_{\text{al}}(\boldsymbol{q})$ were taken as normally distributed $\sim \mathcal{N}(\mu_{\text{al}}(\rho), \sigma_{\text{al}}(\rho))$ with properties defined via analysis of a finite set of GEM simulations for 8 different flux tubes for various core profiles sampled from a priorly assumed $p_{\text{ep}}(\boldsymbol{s})$. To get the compound properties of such aleatoric uncertainty as a function of ρ the $\mu_{\text{al}}(\rho)$ and $\sigma_{\text{al}}(\rho)$, these values were marginalised over the underlying distribution of core profiles $p_{\text{ep}}(\boldsymbol{s}^{\text{c.p.}})$, conveniently reusing the surrogate representation provided by the PCE expansions and the respective quadrature scheme $p_{\text{al}}(\boldsymbol{q}) = \int_{\boldsymbol{s} \in \mathbf{S}} p_{\text{al}}(\boldsymbol{q}(\boldsymbol{s})) dp_{\text{ep}}(\boldsymbol{s}) = \sum_{\boldsymbol{s}_i} u_i \boldsymbol{q}(\boldsymbol{s}_i)$. This way of getting aleatoric uncertainty in turbulent flux values additionally introduces some dependence of aleatoric uncertainty on the epistemic one, as the epistemic uncertainty $p_{\text{ep}}(\boldsymbol{s})$ serves as a prior distribution for the aleatoric uncertainty $p_{\text{al}}(\boldsymbol{q})$.

The raw result of such analysis would be a sample of different final stationary core profiles $\{\boldsymbol{s}^{\text{c.p.}}_{\text{fin}}\}$ for different surrogate as transport model, trained on a perturbed set of flux values. A result for the coefficient of variation of turbulent fluxes $\text{CV}_{\text{al}} = 0.1$ can be seen in the figure 5.10.

Next, we present the aleatoric uncertainty analysis results for a finite set of GEM runs for 8 flux tubes, 4 uncertain quantities captured with PCE of order 2. Each heat flux time trace was analysed according to the procedure discussed in 5.3.2. For each case, the autocorrelation-time-window-averaged subsample was used to calculate the mean, standard deviation, and coefficient of variation of flux value. Then, for each flux tube, we integrated out the local values of profile quantities, which in the case of PCE representation means taking a sum weighted with Gaussian quadrature coefficients. The results, serving as input $p_{\text{al}}(\boldsymbol{q})$ for the aleatoric uncertainty analysis procedure are summarised in figure 5.11.

For these calculated values of $\mu_{\text{al}}[\boldsymbol{q}(\rho)]$ and $\sigma_{\text{al}}[\boldsymbol{q}(\rho)]$ we executed the algorithm described above and the resulting uncertainties of core profiles $p(\boldsymbol{s}^{\text{c.p.}})$ are described in the figure 5.12.

A more detailed analysis judging by the figure 5.13 should go beyond the Gaussian assumption for flux values aleatoric uncertainties and have a more nuanced model for this distribution, parametrised by more than just μ_{al} and σ_{al}. A possible step would be including nonstationary heteroschedastic uncertainty $p_{\text{surr}}(\boldsymbol{q}(\boldsymbol{s}))$ to a GPR surrogate for flux as a function of core profiles or having a separate regression model for parameters of the aleatoric uncertainty.

The software implementation of this procedure employed `EasyVVUQ`, `EasySurrogate`, `MUSCLE3`, `QCG-PJ` libraries, a part of a toolkit suit developed within VECMA and SEAVEA projects [59, 138, 139]. Here, the `EasyVVUQ` [140] was used to define the aleatoric uncertainty $p_{\text{al}}(\boldsymbol{q})$, determine how to treat $\boldsymbol{s}^{\text{c.p.}}$ and $\boldsymbol{q}$ from the point of view of software implementation (`CPO` files and data sets of respective values), as well as the sampling mechanism from the respective

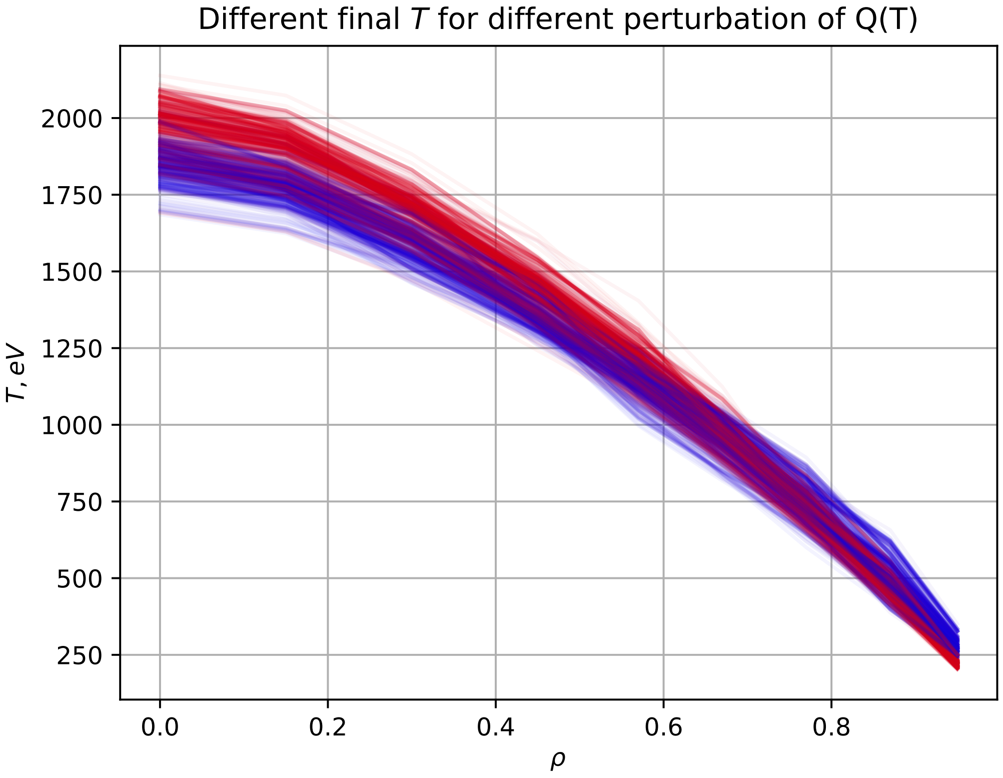

Figure 5.10.: A sample of final stationary core temperature profiles for different perturbed transport models. The results are computed with an ensemble of GPR surrogates trained on perturbed GEM0 data. Shades of red denote T_e whether blue is for T_i.

distributions (here, as MC method with Saltelli sample [49]) and ways to compute expansion weights and respective statistics. The `EasySurrogate` was used to train GPR models for sets of $\boldsymbol{q}(\boldsymbol{s})$ usable at turbulent transport model proxies in simulations. The `MUSCLE3` [141] was used as a coupling library for different components of MFW $EQUIL(\cdot)$, $TRANSP(\cdot)$ and $TURB_{surr}(\cdot)$ solving a system of non-stationary equations for heat transport in a tokamak plasma. Moreover, the `QCG-PJ` was used to manage multiple instances of workflows for different variations of $\boldsymbol{q}(\boldsymbol{s})$ as well as to manage different independent MFW component instances.

The algorithm runs were performed at MPCDF's COBRA supercomputer, each taking 4 nodes of 40 cores for about 4 hours of wall-clock time. Each run consisted of embarrassingly parallel jobs with sequential parts corresponding only to the UQ overhead of preparing samples, encoding, collating the results, and final analysis.

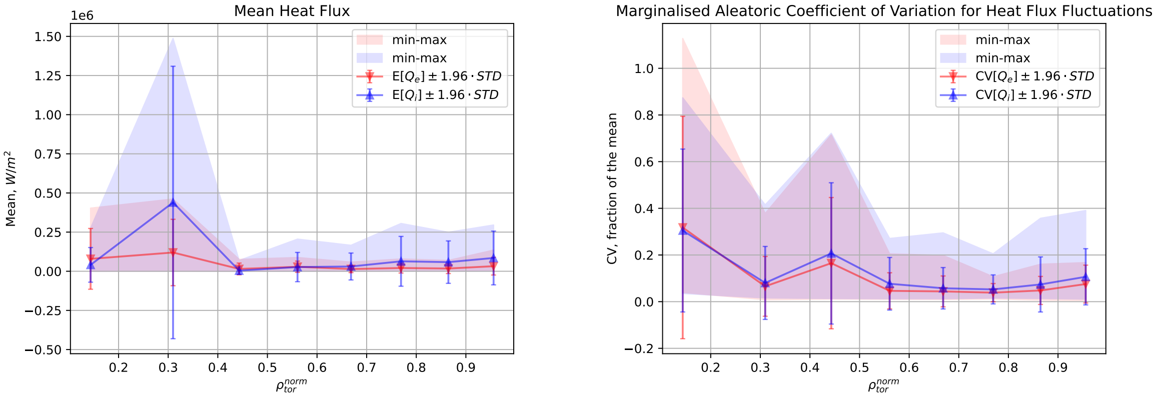

Figure 5.11.: On the left: mean of ion and electron heat flux $\mathbb{E}[Q_{e,i}]$ of GEM for flux tube turbulence simulations of AUG shot #28906, for different epistemic variations of core profiles. Minimum, maximum, and 95% confidence intervals are given. The confidence interval is computed with Gaussian assumptions as $\pm 1.96\sigma[\mathbb{E}[Q]]$. The comparison of the extreme values and standard deviation shows a significant skewness of fluxes towards higher values, with an anomalous peak at $\rho_{\text{tor}}^{\text{norm}} \approx 0.31$. On the right: marginalised over plasma state coefficient of variation for aleatoric uncertainty of ion and electron heat flux fluctuations $\mathbb{E}[\text{CV}[Q]]$ for GEM. Its minimum and maximal values, as well as ± 1.96 standard deviation errors, are indicated. There is a significant spread between minimal and maximal values of CV for variations of core profiles, as well as significant skewness towards higher values, which is common for values from $\mathbb{R}^{0+}$.

Statistics for Core Profiles

Apart from studying the influence of aleatoric uncertainty $p_{\text{al}}(\boldsymbol{q})$ using estimates of flux values fluctuations of GEM, we performed an analysis of such uncertainties in core profiles as a function of variable uncertainties of flux values. Here we assumed a normally distributed uncertainty of the flux estimate $p_{\text{al}}(\boldsymbol{q}) \sim \mathcal{N}(\mu_{\text{al}}, \sigma_{\text{al}})$ with a variable coefficient of variation $\text{CV}_{\text{al}}[\boldsymbol{q}] = \mu_{\text{al}}/\sigma_{\text{al}}$ and cast the uncertainty of tokamak core temperature as a function of these coefficient of variation $\sigma[T_{e,i}|_{\rho_{\text{tor}}^{\text{norm}}=0}] = \sigma[T_{e,i}|_{\rho_{\text{tor}}^{\text{norm}}=0}](\text{CV}_{\text{al}}[\boldsymbol{q}])$.

Figure 5.13 is the result of scanning such a dependency across different values of $\text{CV}_{\text{al}}[\boldsymbol{q}]$ for normally distributed $p_{\text{al}}(\boldsymbol{q})$ and interpreting the results of the scan in terms of the mean and standard deviation of the QoI at the crucial location of $\rho_{\text{tor}}^{\text{norm}} = 0$.

The overall characteristic of such aleatoric uncertainty propagation is that significant variation of the transport fluxes $\frac{\tilde{Q}}{Q} \sim \mathcal{O}(10^{-1})$ lead to much smaller uncertainties for temperatures in the stationary state $\frac{\tilde{T}}{T} \sim \mathcal{O}(10^{-2})$ even though turbulent model exposes significant fluctuations. This optimistic result for practical applications could be attributed to a certain strength of the system's attraction towards an equilibrium core profile state and stability with respect to perturbations. In this case, it happens due to a relatively simple parametrisation of

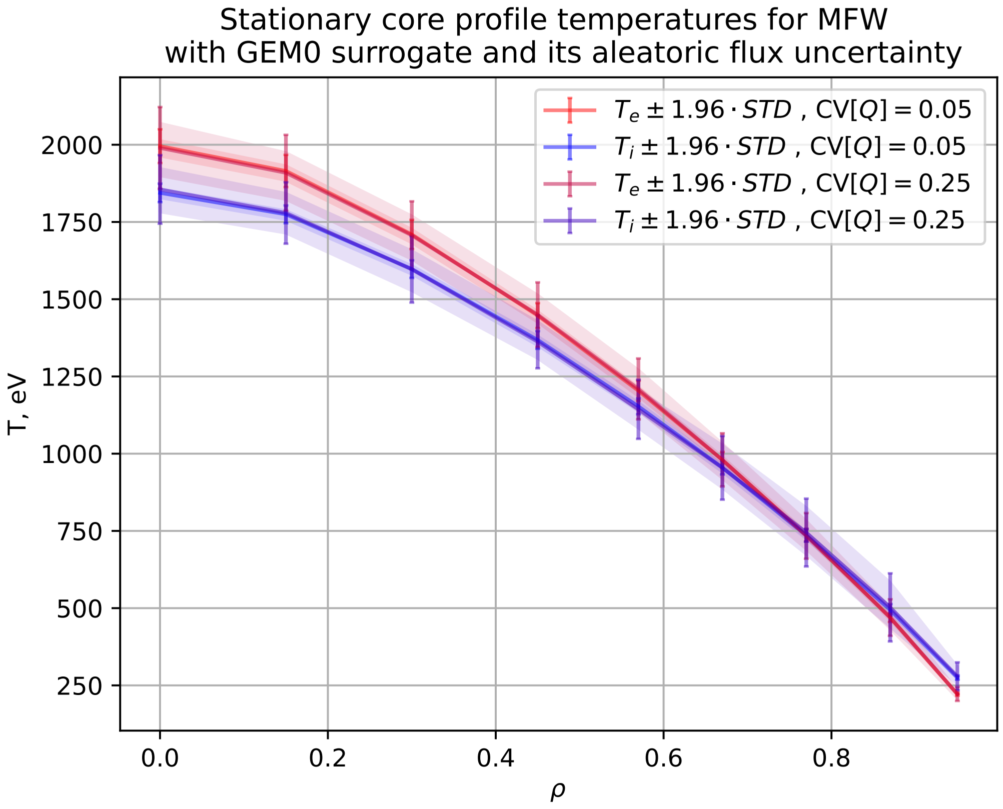

Figure 5.12.: The core temperature profiles, red for electrons $T_e(\rho)$ and blue for ions $T_i(\rho)$, their 95% confidence interval and spread between minimal and maximal values for AUG shot #28906 simulated with an ensemble of GPR surrogates based on a set of perturbed GEM0 simulations.

turbulent transport, and for a model of higher fidelity with more non-linearity and higher stiffness, the domination of convergence over uncertainty might not happen.

Further analysis shows a similar trend for various radial plasma locations as seen in figure 5.14. Furthermore, with the radial coordinate moving outwards, the absolute uncertainties decrease, and relative ones increase.

The standard deviation of core profiles in the simulation's stationary state uncertainties due to turbulent transport aleatoric uncertainty is higher in the inner core and stronger for electrons, as shown in figure 5.15. In relative terms, however, the uncertainty for the outer plasma core is higher and is higher for ions than electrons due to normalisation on smaller values.

As a next step, we performed the SA in terms of total Sobol indices for fluctuations of heat flux at different flux tube locations $Q_{e,i}(\rho)$. We measured sensitivity indices for the fractional influence on the temperature values as the tokamak axis $T_{e,i}|_{\rho=0}$ using bootstrap methods for the Saltelli MC sample. The result of such analysis in figure 5.16 shows that on-axis ion

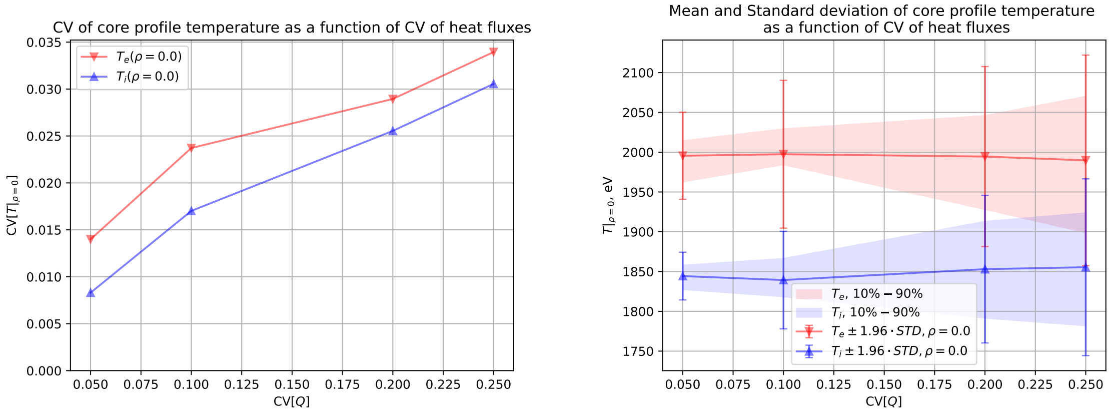

Figure 5.13.: On the right: Coefficient of Variation of core temperatures $CV[T_{e,i}(\rho_{tor}^{norm}=0)]$ as a function of CV of unified aleatoric uncertainty of heat flux values for multiple flux tube turbulence simulations of AUG shot #28906. On the left: mean and 95% confidence interval for core temperature $T_{e,i}(\rho_{tor}^{norm}=0)$ as a function of aleatoric uncertainty of heat flux values. The data for analysis is produced by the algorithm using perturbed GEM0 flux values data.

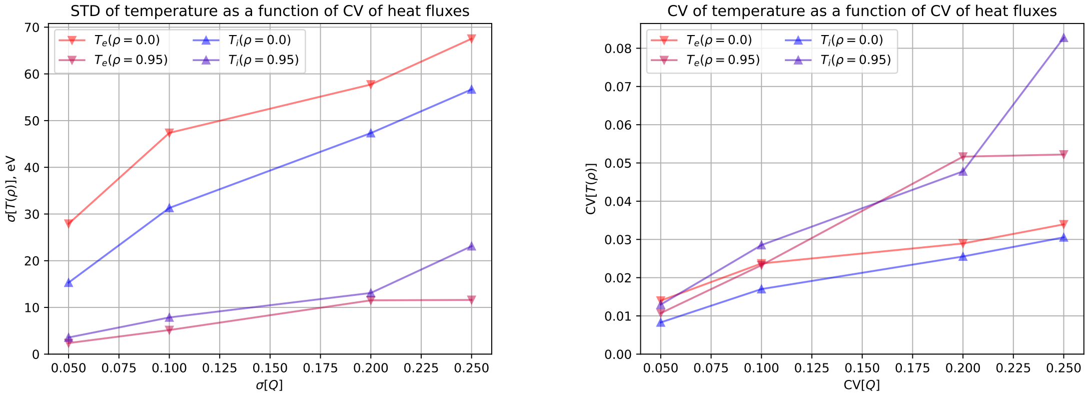

Figure 5.14.: Standard deviation and coefficient of variation for stationary core temperature profiles as a function of aleatoric transport flux uncertainties.

temperature is strongly influenced by ion heat flux value close to the core of the plasma, and other heat fluxes influence on-axis temperature uniformly across the radial coordinate (which might be due to the disadvantages of the statistical method).

For the results of the algorithm for a variable $CV[Q]$, a more detailed analysis that goes beyond the first two moments of the QoI values shows some further properties of profile uncertainties arising due to aleatoric uncertainties of transport flux models. Here, we performed *Kernel Density Estimation (KDE)* for the sample of $T_{e,i}(\rho=0)$ as well as per-case analysis. The density fit displayed in the figures 5.17 shows that such an aleatoric

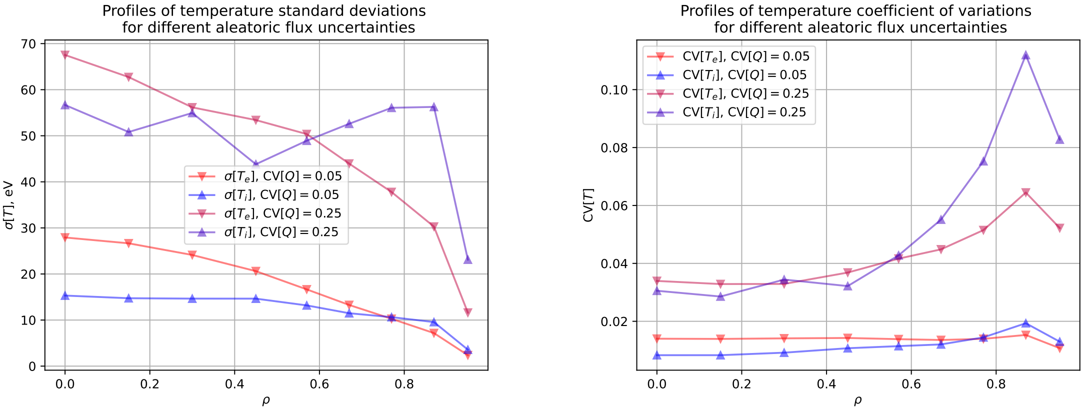

Figure 5.15.: Standard deviation and coefficient of variation for stationary core temperature profiles as a function of radial coordinates and for different aleatoric transport uncertainties.

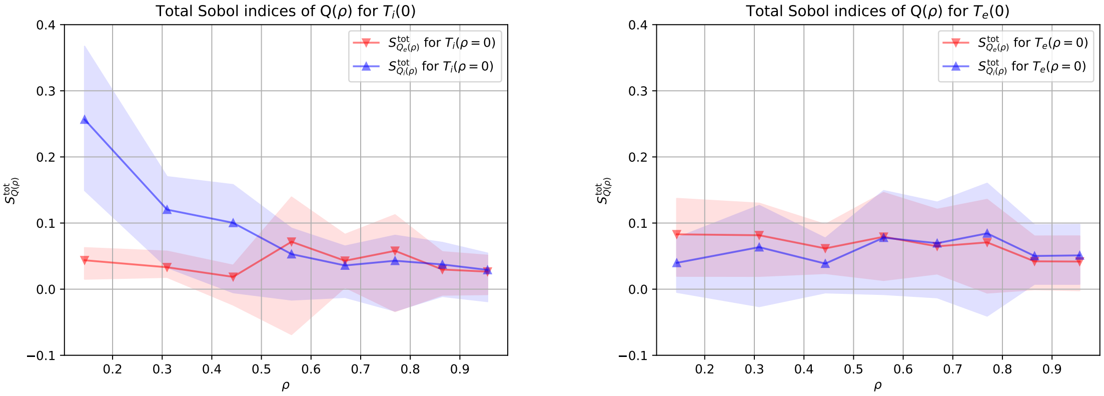

Figure 5.16.: Total Sobol indices for heat fluxes at different radial locations influencing the ion (left) and electron (right) temperatures at the very tokamak core $\rho = 0$.

uncertainty can be distributed non-normally with a significant skewness and, furthermore, be multi-modal, with stationary profiles $\boldsymbol{s}^{\text{c.p.}}$ clustered around several key profile shapes.

Due to the nonlinearity of the dependency of the global system solution on micromodels, the estimates of uncertainties in global QoIs could be a non-trivial non-linear function of the input aleatoric uncertainties. The nonlinearity can lead to the loss of normality of QoI distributions even given normally distributed independent input variables.

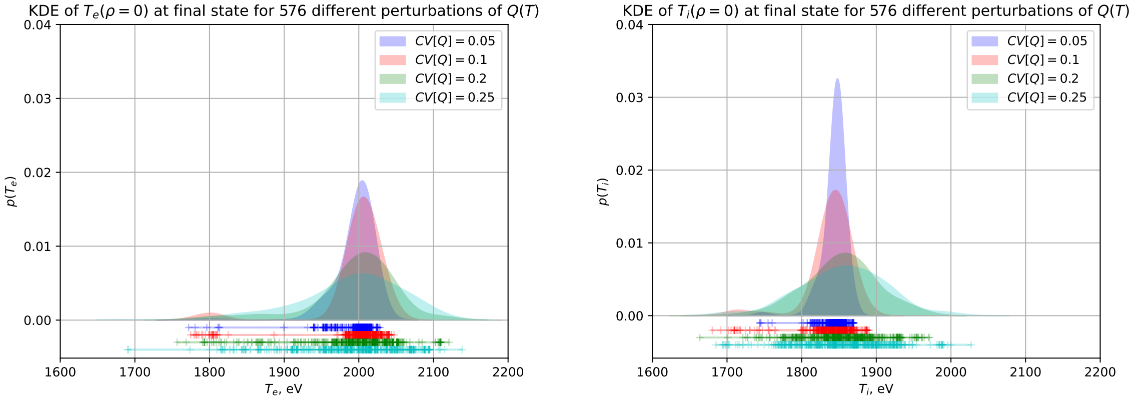

Figure 5.17.: KDE fit of $T_e(\rho = 0)$ and $T_i(\rho = 0)$ for different perturbed turbulent transport models and different aleatoric uncertainty of those. Crosses denote the individual readings. The smaller uncertainty in fluxes yields more peaked uncertainty of the core's temperature and has lower mean values. After a particular flux uncertainty value, the overall spread starts increasing. Furthermore, multi-modal non-Gaussian distributions of core temperatures arise. For both temperatures, there is a some skewness towards smaller values. Also, core ion temperature uncertainty significantly peaks more than electron one. The described algorithm used surrogates based on perturbed GEM0 data.

5.4. Aspects of Computational Cost-benefit Analysis. Gain and Profitability of Surrogate Approach

Creating a surrogate requires a sufficient amount of data to regress the studied model dependencies over parametric regions of interest. In a vast majority of practical applications, no preliminary data suitable for a given problem is available. Regardless of the methodology of the design of numerical experiments, one should perform a finite number of simulations to get a sufficiently large sample of model solution function evaluations. For high-fidelity codes, the computational cost of a single simulation could be significant, which leads to a rise of practical questions about whether applying a surrogate approach would require fewer computational resources than solving the problem with an expensive simulation in a loop. Apart from the question of how to design numerical experiments to get the data required to solve the problem as cheaply and efficiently as possible, one should also compare the total cost of data acquisition for a surrogate with the cost of a direct solution. A further feature of high-fidelity cost utilisation is that the solution of this model dominates the computational cost, and other steps of algorithms, like resolving the largest scales, performing optimisation steps, training, or calling regression models, introduce only a minor additive correction in the cost estimate. This does not affect, however, the cost and complexity of the version of the algorithm that uses a code or a surrogate as its step. The cost of an algorithm is understood as a function of the number of DoFs that define the model's behaviour on the level necessary to solving the given problem.

In this work, the problem that requires multiple solutions of a model is forward uncertainty quantification for coupled transport simulations depending on turbulent fluxes. The most expensive micro-model is the model for turbulent processes, which is affected by a number n_m of parameters describing the state of the plasma $\boldsymbol{s}$. Usually, turbulent behaviour is defined by driving terms like gradients in the system. In this work, as we study heat fluxes in flux tube approximation, we analyse the influence of $n_m = 4$ parameters on turbulence: $T_{e,i}, \nabla T_{e,i}$. In order to create a surrogate, one should express the dependency of QoI on the most essential n_m parameters in sufficient accuracy. In this work, we performed studies using a PCE method of order $p = 2$ for the GEM model and $p = 4$ for the GEM0 model, using a tensor product of grids of independent scalar parameters. This requires, in our case, a total number of runs (simulations) $n_r^m = (p+1)^{n_m} = 3^4 = 81$ as a baseline number to capture the dependency of the model for any given situation for a single flux tube with $n_{\text{f.t.}} = 8$ flux tubes to capture radial dependency of turbulence for AUG. Furthermore, we perform a convergence and aleatoric uncertainty analysis for every simulation of turbulent fluxes, which requires different durations of simulations. Still, on average, every run required $n_{\text{t.st.}}^m \approx 2000$ internal time steps to estimate mean flux values and their uncertainty. Each of the time steps was performed using $n_c = 32$ of COBRA cores and requiring $t_{\text{t.st.}} \approx 0.25$ hours with good scalability and a small fraction of cost for non-parallelisable overhead. This leads to a computational cost of a single time step $C_{GEM-t.st.} \approx 8\,\text{CPUh}$, on average approximately $C_{GEM} = n_{\text{t.st.}}^m \cdot C_{GEM-t.st.} \approx 16\,\text{kCPUh}$ for a single GEM simulation and totally $C_{\text{GEM-DS}} = n_r^m \cdot C_{GEM} \approx 10\,\text{MCPUh}$ for a dataset to create a surrogate.

On the other side, the epistemic uncertainties of the coupled workflow on the scale of the largest macro model are defined by several n_M DoFs. In our case, the turbulent transport simulations on the scale of core kinetic profiles are defined by models for boundary conditions and right-hand side of equations, or sources and sinks, more precisely in this case, meaning the boundary conditions for temperatures at the edge of the plasma T^{BC} and sources of energy in the inner parts of plasma S^E. This is a simple case that can be parametrised with $n_M = 6$ scalars: ion and electron temperatures at the edge of the plasma $T_{e,i}^{\text{BC}}$, total integral source heat in ions and electrons $Q_{e,i}^{\text{tot}}$ and the position H_0 and width H_w of a Gaussian heating profile. In a previous work [123], an analysis of sensitivities to these parameters, considered as having Normal uncertainties with CV $= 0.2$ coefficient of variation, was performed for an MFW version with GEM0 as a turbulent transport model. Performing the same analysis with GEM instead of GEM0 would, for a single run, require around $n_{\text{t.st.}}^M \approx 7500$ time steps which requires about $C_{\text{MFW}} = n_{\text{t.st.}}^M \cdot C_{\text{MFW-it.}} \approx 97\,\text{kCPUh}$. A forward uncertainty PCE analysis of complexity $n_r^M = (p+1)^{n_M}$ with the same $p = 2$ would require thus $C_{\text{MFW-UQ}} = n_r^M \cdot C_{\text{MFW}} = 70.9\,\text{MCPUh}$

In general, this leads to the estimate of the computational profitability of the surrogate approach in the case when the total cost of creation and usage of a surrogate is less than the total cost of solving a problem using simulations with the original model:

$$f_C = \frac{C_1}{C_2} = \frac{C_{\text{surr}} + c_{\text{surr-o.h.}}}{C_{\text{WF}}^m + c_{\text{WF-o.h.}}} \approx \frac{C_{\text{GEM-DS}}}{C_{\text{MFW-UQ}}} < 1 \tag{5.2}$$

The cost of surrogate creation is expressed as $C_{\text{surr}} = N_L(n_m, \theta_{\text{ref}}) \cdot C_m$ where the cost of the micro-model (turbulence) run is expressed in terms of number of time steps and a time step atomic cost $C_m = n^m_{\text{t.st.}} \cdot c^m_{\text{t.st.}}$. The $N_L(n_m, \theta_{\text{ref}})$ is the complexity of the surrogate learning algorithm in terms of number of function evaluations required, which could be a non-trivial function of number of micro-model parameters n_m and algorithm refinement parameters θ_{ref}. For a simple case that does not utilise any dynamic information about the problem, e.g., a full-product-grid quadrature bases method, this complexity is exponential in the number of parameters and, in the case of the PCE algorithm version used in this work, it is $N_L(n_m, p) = (p+1)^{n_m}$. A variation that uses polynomials only up to degree p uses a slightly smaller combinatorial complexity of $\frac{(p+n_m)!}{p! \cdot n_m!}$. In this work, the number of internal solution time steps for a single case $n^m_{\text{t.st.}}$ is also a non-trivial function of the parameters of the micro-model due to the presented convergence analysis, however, in this cost analysis, we are using average values $\bar{n}^m_{\text{t.st.}}$. The cost of the single solution step $c^m_{\text{t.st.}}$ also depends on multiple factors, including the physics and the level of the fidelity of the model, the level of the refinement in terms of internal DoFs, the numerical approach and the HPC hardware and configuration, which is left out of the scope of the presented analysis and with only empirical average values used. The cost of the surrogate usage as a part of the simulations, as well as the cost of the surrogate training (in terms of incorporating the existent simulation data in the surrogate), is considered to be a subdominant cost overhead, which, however, in some cases may still have a significant contribution.

The determinant part of the expression denotes the cost of the direct solution that does not include the cost of the surrogate expression and is dominated by the cost spent on the micromodel evaluations C^m_{WF} dependent on the algorithmic complexity of the problem itself and the cost of the micro-model $C^m_{\text{WF}} = N_{\text{prob}} \cdot C_m$. In the previous example, however, the problem considered is the forward uncertainty propagation using the full-product grid for the uncertain parameters, meaning the same type of algorithmic complexity, which, however, here depends on the parameters of the macro-model and workflow on the largest scale $N_{\text{prob}} = N_{\text{prob}}(n_M, p)$. The part of the cost spent outside the expensive macro model is considered to be a more negligible overhead. Furthermore, the surrogate-related overhead cost $c_{\text{surr-o.h.}}$ usually includes a term that is proportional to the complexity of the problem N_{prob}. However, the multiplicative factor of the single function evaluation cost for a surrogate is much smaller than the one for a physical model $C_{\text{surr}} \ll C_m$.

In our case, given a small overhead related to operations and models other than the most expensive one $c_{\text{surr-o.h.}} \ll C_{\text{surr}}$ and $c_{\text{WF-o.h.}} \ll C^m_{\text{WF}}$, performing UQ with or without surrogate is computationally beneficial by a factor of $f_C \approx \frac{10\,\text{MCPUh}}{70.9\,\text{MCPUh}} \approx 0.14$.

Having in mind this computational cost comparison, it is worth noting that a surrogate approach is practical only in cases of a sufficiently expensive problem requiring simulation-in-the-loop, and for simple cases requiring only a few simulation instances, surrogate creation might not be worthy when $f_C > 1$. However, for more ambitious problems requiring analysis with respect to a large number of macroscopic parameters n_M and a high level of chosen algorithmic complexity, chosen by θ_{ref}, the surrogate approach might be a necessity if one can achieve $f_C \ll 1$. The figure 5.18 shows the analysis of situations where performing UQ is more cost-beneficial with surrogate models. However, none of the cost analyses presented in the expressions or plots considers the trade-off with the surrogate quality. This is supported by the practical experience during the work that the surrogate, once trained with a finite resolution, e.g., $p_m \geq 2$, can approximate the workflow solution well enough. This experience leads to the further conclusion that for a given workflow, the biggest challenge for any parametric study is the curse of dimensionality and not the complexity of the dependency.

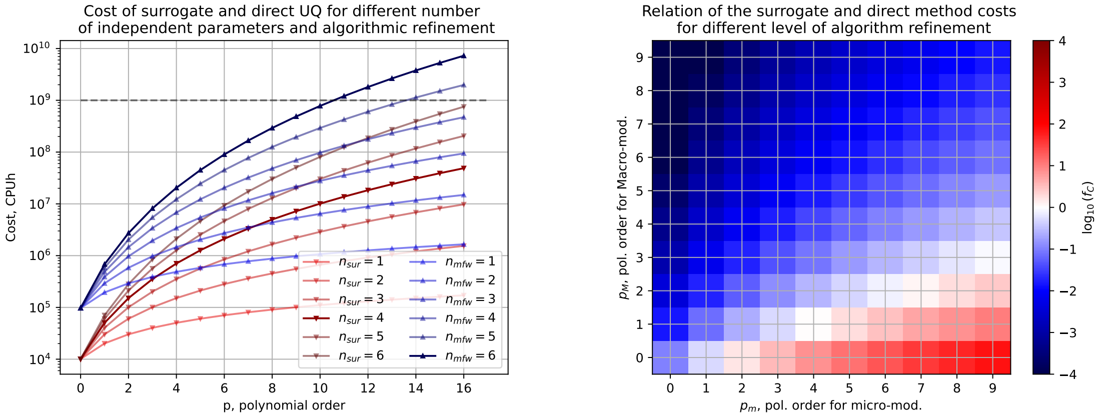

Figure 5.18.: The analysis of the cases for different numbers of models' independent parameters n and refinement of the algorithm (polynomial order p). The first plot shows cost dependence of cost on the polynomial order $C(p)$ for different parameter numbers n — there are situations for large turbulence micro-model parameters when it is no longer suitable to use surrogates. In the default workflow formulation used in this work with $n_m = 4$ and $n_M = 6$ and for the given UQ procedure, the usage of surrogates is beneficial with a large margin for any computationally feasible resolution of all the parameters (marked by $C_{\max} = 1\,\mathrm{GCPUh}$ in the plot). Furthermore, the inclusion of 2 additional equally well-resolved channels into a surrogate, be it densities $n_{e,i}$ or equilibrium $1D$ profiles q and gm_3, would still lead to surrogate approach cost justification. The second plot shows a heat map of cost fraction $f_C(p_m, p_M)$ as a bivariate function of polynomial order of analysis for micro-model (turbulence) p_m and macro-model (workflow) p_M. It is beneficial to apply a surrogate for all cases but a high resolution of micro-model dependencies with its surrogate.

A further important aspect is the *cost amortisation* of the surrogate model, which in this context means the further utilisation of the surrogate model for different problems in order to save computational cost on performing high-fidelity simulations. In practice, that means that having a sufficiently accurate and general surrogate, meaning having a high expressivity and a high coefficient of determination (and small validation errors) for a wide distribution of parameters, can computationally be very profitable. Having such a well-trained and validated surrogate, one should choose further problems $\mathbb{P}$ to solve with the surrogate to make it more worthy of creation. In terms of the cost factor, this would mean enlarging the determinant by using multiple problems $f = \frac{C_{\mathrm{surr}}+c_{\mathrm{surr-o.h.}}}{\sum_{i\in\mathbb{P}}\left(C^{m}_{\mathrm{WF,i}}+c_{\mathrm{WF-o.h.,i}}\right)}$.

The further advantage of the surrogate is in the wall-clock time requirement to analyse a particular situation, e.g., tokamak discharge. Using a computational workflow like MFW with an expensive high-fidelity turbulent transport model requires a particular time for solving and predicting profiles, which may not be reduced to the final scalability of the code. On the contrary, producing data for surrogate requires significantly less turbulent transport simulation time, e.g., in the presented case $n^{GEM}_{\mathrm{t.st.}} \approx 2000 < 7500 \approx n^{MFW}_{\mathrm{t.st.}}$. Given a sufficiently large amount of instantly available computational resources, the wall-clock time of producing a result with a code in a workflow and with a surrogate in a workflow will be proportional to the number of time steps of the most expensive model needed to obtain the required information. Such an *embarrassing parallelisation* might not be true if one uses an AL scheme that approaches training data acquisition from simulation sequentially, with a surrogate update on every step of a process. In an example, where a surrogate is trained to fit well for specific flux values, the preparation of the surrogate will require wall-clock time equal to several convergence times of surrogate code simulation. Given the approach presented in this section, predicting a tokamak discharge stationary kinetic profiles using GEM in MFW takes $\sim 95\,\mathrm{h}$ on 1024 cores. In contrast, preparing a surrogate would take $\sim 26\,\mathrm{h}$, leaving days to research and modelling a discharge using surrogate-based simulation, which might be crucial in planning experiments.

5.5. Chapter Summary

This chapter presents the uncertainty quantification algorithms that involve an entire multi-scale workflow. We presented the turbulence manager modification of the MFW that allows us to dynamically choose different implementations of a model for turbulent flux computation and switch to high-fidelity turbulence code utilisation if the surrogate trustworthiness is too low. Furthermore, the chapter presents a procedure to update surrogates based on the evolution of the quantities computed by the entire workflow and its benefits, limitations, and implications. It discussed an algorithm utilising surrogates that quantify the influence of aleatoric uncertainties in fluctuating flux values on the computation of stationary core profiles of tokamak plasmas. Finally, we discussed the justification of the surrogate approach to uncertainty quantification algorithm in multi-scale simulations from the point of their computational cost.

6. Discussion

This chapter summarises the work performed in the scope of this thesis, discusses the questions left open, and outlines the possible direction for future research.

6.1. Summary

This work overviews how uncertainty considerations could be introduced in plasma physics simulations, how these uncertainties can be quantified, and how to develop algorithms for such uncertainty analysis that utilise computing power effectively.

One of the types of uncertainty considered in this work is *aleatoric* uncertainty, which is an irreducible uncertainty due to chaotic or stochastic fluctuating behaviour of the physical system. Here, we analyse how turbulent heat fluxes behave on the microscopic time scales and suggest a way to characterise fluctuations for quasi-stationary states, as well as a way to determine statistical errors in the estimation of the effective level of fluxes that allows for saving compute time by controlling the simulation duration. An algorithm for sequential analysis of flux time traces was developed and applied to an ensemble of $3D$ nonlinear electromagnetics gyrofluid code GEM runs, which identified that via the analysis of convergence of estimates of the errors of the mean for fluxes, it is possible to save up to 50% of computational resources compared to the default applied approach.

The other type of uncertainty we consider is *epistemic*, which arises from a lack of knowledge of the model and its parameter values. In this work, we used Polynomial Chaos Expansion to analyse how uncertain values of kinetic profiles of a tokamak core plasmas can influence turbulence and effective anomalous transport, as well as how such uncertainties can propagate through a system of non-stationary equations describing the transport of heat in a tokamak. The performed study also allowed us to apply sensitivity analysis for turbulent flux values, quantitively recovering the most important parameters influencing turbulence. We created a software workflow to propagate uncertainty, collect and store data from ~ 1000 high-fidelity plasma turbulence simulations, analyse the produced time traces, and quantify parametric uncertainties. The study recovered that the turbulence model GEM can amplify uncertainties in profile, translating them into ~ 12 times larger uncertainties in fluxes. For the studied cases, $\sim 40\%$ of flux variation could be attributed to ion temperature gradient uncertainty, with a variation depending on flux tube location ρ_{tor} and magnetic geometry, with $\sim 20\%$ attributed to the interaction of profiles' parameters.

This work studied an approach to decrease the computational expense of performing uncertainty quantification in turbulence simulations and running multiscale turbulent transport simulations, which is the *surrogate* approach. We focused on the inference of the likelihood of radial heat flux values based on a space of plasma described by core kinetic profiles using Gaussian Process Regression models based on a data set of plasma turbulence simulations. We implemented a GPR extension for the `EasySurrogate` library [136] capable of collecting turbulence simulation, training and tuning an ensemble of ML models using HPC resources, validating and testing the performance of surrogates for flux value inference, and their utilisation and active learning coupled to multiscale turbulent transport simulations. The resulting surrogate models reached the coefficient of determination quality level of $R^2 \approx 0.99$, which was, in practice, sufficient to capture the studied dependencies.

Furthermore, the surrogates were applied as a replacement for the numerical turbulence solver in the coupled workflow for the turbulent transport problem. We modified the simulation workflow capable of tracking the fidelity of surrogate prediction, identifying its high epistemic uncertainties, avoiding switching to an extrapolation regime, and dynamically choosing between surrogate or numerical code flux calculation. Moreover, we implemented a surrogate tuning procedure that allowed us to retrain the surrogate based on the profiles produced by multiscale simulations, which allowed us to use surrogates to predict steady-state tokamak core temperature profiles within 1% error compared to the ones computed by multiscale workflows using the turbulence code that produces the data for the surrogate.

Finally, we implemented a surrogate-based algorithm to quantify the influence of aleatoric uncertainties in turbulent flux calculation on the prediction of core kinetic profiles. The studies performed by the algorithm allowed us to observe a $2-8$ times decrease in relative uncertainties in profiles for higher uncertainties in computed flux values and to assign consistent uncertainties of predicted core profiles based on simulations.

6.2. Outlook

The data set of high-fidelity turbulence simulations used to train surrogates was limited due to the high computational cost of simulations. There are strong indications that it does not represent the turbulent transport behaviour on a large enough domain of parametric space to be confidently used for AUG profile predictions. For future work, capturing larger regions of interest with a higher variability of parameter values would be beneficial. However, due to the high computational cost of simulations and high dimensionality of the transport dependency, the particular choice of experimental design for new simulations to populate the training data set should use physical considerations and approaches for surrogate learning discussed in this work. For instance, the MFW simulations demonstrated that the range of ion temperature gradients, especially for the intermediate values of radial coordinate, is insufficient and should be expanded, which can be formally included in the learning process. Furthermore, the surrogates would benefit from including information on dependency on more physical parameters.

One of the parts deliberately missed in this work is the question of particle transport. It is believed, on a rather qualitative level, that it is essential for methodology development to have characteristics similar to those of heat transport. One of the following steps should be applying the same methodology to heat transport to get an idea of whether the same methods are applicable to it and to more physically consistent simulations of both heat and transport in a multiscale workflow.

Including density parameters and particle flux, or other physical quantities, will increase the effective dimensionality of the problem, which leads to several complications. One is the need to decrease the dimensionality dependency of transport, meaning finding a smaller number of parameters that influence it the most. The dimensionality reduction question was touched upon a number of times in this work, and several approaches to it were suggested. One is based on utilising SA to resolve some of the parameters in less detail or completely discard the influence of their variation. The other related method would be to use explicit knowledge of the correlation among dependent and independent parameters. In this work, uncertainty propagation methods assumed probabilistic independence of input parameters, and none of the methods explicitly leveraged the reduction of complexity that comes with correlation. Related to this, one may have a different choice of the independent parameters influencing turbulent transport, using the expectations provided by physical theory and experiment. Also worth applying are model reduction methods based on other principles, like linear and nonlinear dependency analysis: SVD, VAE, DAS, and other methods.

The surrogates presented in this work could be improved in multiple ways. The GPR applied to model turbulent transport can benefit from more explicit utilisation of physical knowledge, which includes adding information on the model's gradients or knowledge on a different character of dependency for different domains of parametric space corresponding to different dominant types of turbulence. Furthermore, the GPR model could be modified to accommodate non-stationary *heteroscedastic* noise properties.

The inclusion of more independent parameters, entailing the effective increase of problem dimensionality, may require significant reconsideration of the types of surrogates applied. There are many ways to include more information in the surrogate, but training it not only on the statistics of the simulations but also on finely resolved data which may require employing parametric machine learning methods like Deep Neural Networks for surrogates. Such a surrogate may benefit from a multi-fidelity approach, assimilating data from solutions of the same problem using codes or resolutions of different fidelity.

Higher dimensionality may lead to the need for explicit utilisation of the sensitivity of the model, data correlation, and effective representation of the space of independent parameters. A possible approach to that, while pertaining to the probabilistic nature of the surrogate model, would be combining the ideas of DL and Gaussian Process Regression. This could be done via BNNs, explicitly encoding probabilistic relationships between layers. The other types of probabilistic models, in the sense of their ability to predict uncertainties, could be applied, including VAEs, which are suitable for nonlinear dependencies, search for small latent spaces, and work with correlated data. Furthermore, one can apply Deep Gaussian Process regression, forming a composition of several GPR models. Another approach would be to apply a DNN model to express the covariance function in the high-dimensional input space and serve as a GPR model component.

This work primarily considers the global properties of the model, including broad regions of interest and independent parameter distributions. The other approach would utilise more local data, meaning the gradients of the model with respect to its parameters. Having a fully differentiable numerical model with access to the Jacobians allows the use of derivative-based sensitivity analysis and advanced methods of variational inference, which can significantly save computing resources.

6.3. Discussion

One of the questions arising is the applicability of the presented methods to a broader variety of cases and ways of including uncertainties that appear due to the variability of a larger number of parameters.

For example, the influence of equilibrium magnetic geometry is not considered in detail in this work. On the one hand, it is based on prior knowledge, which is an aspect that weakly influences plasma turbulence. However, from the other side, the evolution of equilibrium distributions is present in the solution of non-stationary cases, and a finite change of turbulent transport fluxes due to equilibrium variation was observed in this work alone. Thus, a more complete approach would require analysing uncertainties due to equilibrium and including more equilibrium effects in the surrogate, both of which are complicated due to a large number of degrees of freedom used for a parametrisation of equilibrium magnetic geometry.

Also, applying the approach to capture uncertainties accounting for a significantly more significant number of parameters could benefit from quantitatively discriminating them by their importance and influence on the QoIs. In such cases, algorithms that utilise sensitivity analysis to rank and classify the parameters by importance and choose their variability level or the accuracy level of capturing dependency on these parameters may be beneficial. These approaches include an adaptive sparse grid that uses sensitivity metrics to refine the resolution of a subset of parameters by adding more quadrature points to the grid.

Furthermore, analysis of an arbitrary discharge (and of different devices) would require starting all the computational workflows from scratch due to an inherently local and biased understanding of uncertainties.

Considering the question of surrogates, even though one would like to reuse as much information on the parametric dependency of the model, creating a single surrogate that will capture the plasma turbulent transport for arbitrary situations is still very expensive and often not needed in practical applications. One of the approaches would be to create a surrogate for a particular device, discharge, or otherwise case with a large number of parameters practically fixed and use such a surrogate in a very targeted manner. Such an approach would save a lot of computing power required to collect training samples for such a surrogate. Also, it can trade a level of generality of a surrogate to increase its accuracy. Using a more targeted surrogate, however, would require a higher level of algorithmic automation of collection of a sample of numerical solutions to train a data-based model on, and the process of its active training and fine-tuning according to criteria that a problem, e.g., quantifying uncertainty for a particular discharge, is requiring.

Some intermediate approaches that assume the usage of a general surrogate, fine-tuned via an active learning process coupled with a simulation workflow, could be a possible way to get both good generality and accuracy and save the total computing resources the most. The combination approach would highly benefit from an effort towards creating *foundational* models coupled with the data from high-fidelity numerical models for computationally expensive plasma physics problems, especially the problem of turbulent transport.

At the moment of writing this work, the inherently complicated nature of plasma physics modelling and the high computational cost pose significant challenges for creating machine learning surrogates based on high-fidelity simulation data. However, the immense potential of this approach gives significant promises for the surrogate-based modelling of future devices or experiments and provides incentives for future research in this field.

A. Appendix

A.1. Uncertainty Propagation

To perform uncertainty propagation of some random input variable through a model and analyse the uncertainties in its solution, one is required to perform the following steps described in A.1.

1. Define input parameters $\boldsymbol{x}$
2. Define PDF type and parameters for inputs $p(x)$
   - In the general case, it could be defined as a joint probability distribution, but in the vast majority of applications, a number of independent input components are assumed $p(x) = \prod_i p(x_i)$
   - In notation of `EasyVVUQ` and used by it `ChaosPy` library those are `vary` and `distribution` objects
3. Define the model's QoI or outputs $\boldsymbol{y}$
4. Define how the program implementations can read and modify its inputs and outputs
   - In `EasyVVUQ` terminology it is equal to definition of `Encoder` and `Decoder`
5. Define the model to express uncertainty.
   - This requires the definition of the sampling scheme as well as its effective sample size and fidelity
6. Defining the program implementation of the studied model and its parameters
   - This requires specifying the program executables as well as managing all its dependencies, like input files, libraries, environments, etc.
   - An important step is specifying the means to parallelise the execution of solution for multiple model instances: how to run multiple programs, its parallelisation, and the resources required for a run
7. Definition of the analysis procedure
   - This includes definition of what statistics over output $\boldsymbol{y}$ shall be computed and any additional steps; this corresponds to `Analysis` objects of `EasyVVUQ`
   - Generally, the type of the analysis is bound to the uncertainty expression model and the respective sampler

The uncertainty propagation performed in this work was done using the EasyVVUQ Python library [142]. The details of the steps were given while keeping in mind the implementations of this library and are depicted in fugure A.1 using the EasyVVUQ terminology. However, they are typical for any software designed for forward uncertainty problem solutions.

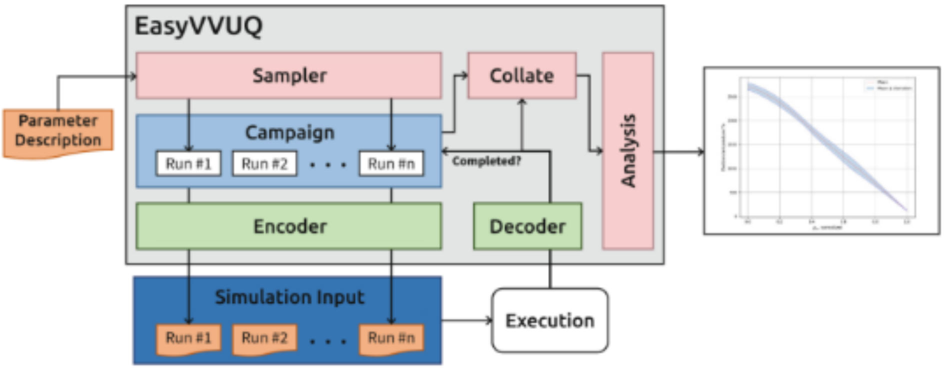

Figure A.1.: A visual explanation of the steps and program element required to propagate uncertainty through a numerical code.

A.2. Consistent Physical Objects and Their Modification

The basic unit of data storage and exchange in this work is a *CPO* [125]. It is a data structure consisting of a hierarchical tree of dictionaries, where labels describe the physical quantities or their categories, and leaf elements describe their values. The values can be scalar $0D$ numbers, $1D$ array for quantities with a radial dependency, or $2D$ array for quantities depending on radial coordinate and poloidal angle. This data structure describes the physical state of a tokamak plasma, or its evolution, and ignores toroidal dependencies. Several different CPO types describe different physical processes, and in this work, we mainly use CPOs of `coreprof`, `equilibrium`, and `coretransp` types.

The CPO format can be serialised as `ascii` format files, in binary format, and deserialised into Fortran of Python objects.

In this work, we were interested in specifying a variation of a plasma state, described via a number of radial profiles, to capture the epistemic uncertainty of plasma. This was done on the level of modification of `coreprof` CPO, serving as an input to the turbulence code. Due to the consideration of local turbulence in flux tube approximation only, this required us to modify the corresponding profile values and their gradients (T_e, T_i, ∇T_e, ∇T_i) in the neighborhood of the flux tube located at a specific value of radial coordinate $\rho_{\text{tor}}^{\text{norm}}$. The flux tube turbulence model considers a single value of the background temperature and a single value of its gradient, meaning it is sufficient to present the profile as an interval of line.

Dealing with uncertainties in $1D$ profiles, we utilised the local turbulence model to transform the problem into describing the uncertainties of scalar variables for quantities at fixed values of radial coordinate ($T_{e,i}(\rho)$, $\nabla T_{e,i}(\rho)$). This allowed to specify uncertainty of a particular probability distribution for separate scale random variables and sample according to the chosen scheme $T_i^{\text{pert}} \sim p(T_i|\boldsymbol{\theta})$, then collecting a sample of the modified scalars $\{T_i^{\text{pert}}\}$.

Due to the employed 3^{rd} order Lagrange interpolation to compute the gradients and their profiles, and due to different radial grids for turbulence (8 flux tube locations for AUG tokamak) and transport (100 points for radial coordinate) we required to linearly extrapolate the specified temperature quantities using the radial gradient value onto the 4 neighboring points. Then, without loss of generality, the workflow will interpolate the profiles onto the location of the flux tube and afterwards interpolate the transport coefficient on the transport radial grid using Lagrange interpolation.

This core profile uncertainty encoding process is described in the pseudocode scheme 2.

Algorithm 2 CPO coreprof encoding

```
For given flux tube location $\rho^{\text{f.t.}}$, interpolate the default profile value $T_{e,i}^{\text{def}}(\rho^{\text{f.t.}})$ and its gradient $\nabla T_{e,i}^{\text{def}}(\rho^{\text{f.t.}})$ using Lagrange interpolation $L_{T_{e,i}}(\rho) = \sum_{j=0}^{p} T_{e,i}(\rho_j) \prod_{0\leq m\leq k, m\neq k} \frac{\rho-\rho_m}{\rho_j-\rho_m}$ of order $p$ using profile radial grid points $\rho_{i=0..p}$, and respectively the derivative of Lagrange interpolating polynomial $\nabla_\rho L_{T_{e,i}}(\rho) = \sum_{j=0}^{p} T_{e,i}(\rho_j) \prod_{0\leq m\leq k, m\neq k} \frac{\rho-\rho_m}{\rho_j-\rho_m} \sum_{i=0, i\neq k}^{k} \frac{1}{\rho-\rho i}$ ▷ Here: for most of the uncertainty PDFs $p(\cdot|\boldsymbol{\theta})$, for given $\rho^{\text{f.t.}}$ it serves as the parameter of mean value $\mu$
Sample the profile and gradient values $T_{e,i}^{\text{pert}}(\rho^{\text{f.t.}}), \nabla T_{e,i}^{\text{pert}}(\rho^{\text{f.t.}}) \sim p(T_i, \nabla T_{e,i}|\boldsymbol{\theta})$ ▷ Here: we use Normally distributed uncertainty $\sim \mathcal{N}(\mu, \sigma^2)$ and PCE sampling using Gaussian quadrature nodes
Find $p+1$ points $P = \{\rho_{i=1..p+1}^{\text{c.p.}}\}$ of profile grid nearest to the $\rho^{\text{f.t.}}$ via *bisection* algorithm ▷ Here: $p$ is the order of interpolation polynomial
Compute the linear extrapolation for the perturbed segment of a core profile $\forall \rho_i^{\text{c.p.}} \in P$ : $\nabla T_{e,i}(\rho_i^{\text{c.p.}}) = \nabla T_{e,i}^{\text{pert}}(\rho^{\text{f.t.}})$, $T_{e,i}(\rho_i^{\text{c.p.}}) = T_{e,i}^{\text{pert}}(\rho^{\text{f.t.}}) + \nabla T_{e,i}^{\text{pert}}(\rho^{\text{f.t.}}) \cdot (\rho_i^{\text{c.p.}} - \rho^{\text{f.t.}})$
Replace the corresponding readings of the CPO object using the resulting $T_{e,i}(\rho_i^{\text{c.p.}})$ and $\nabla T_{e,i}(\rho_i^{\text{c.p.}})$, pass it to the turbulence code
```

Figure A.2 shows a result of such encoding and profile modification for a single flux tube at $\rho_{\text{tor}}^{\text{norm}} \approx 0.66$ using a sample of 16 perturbations.

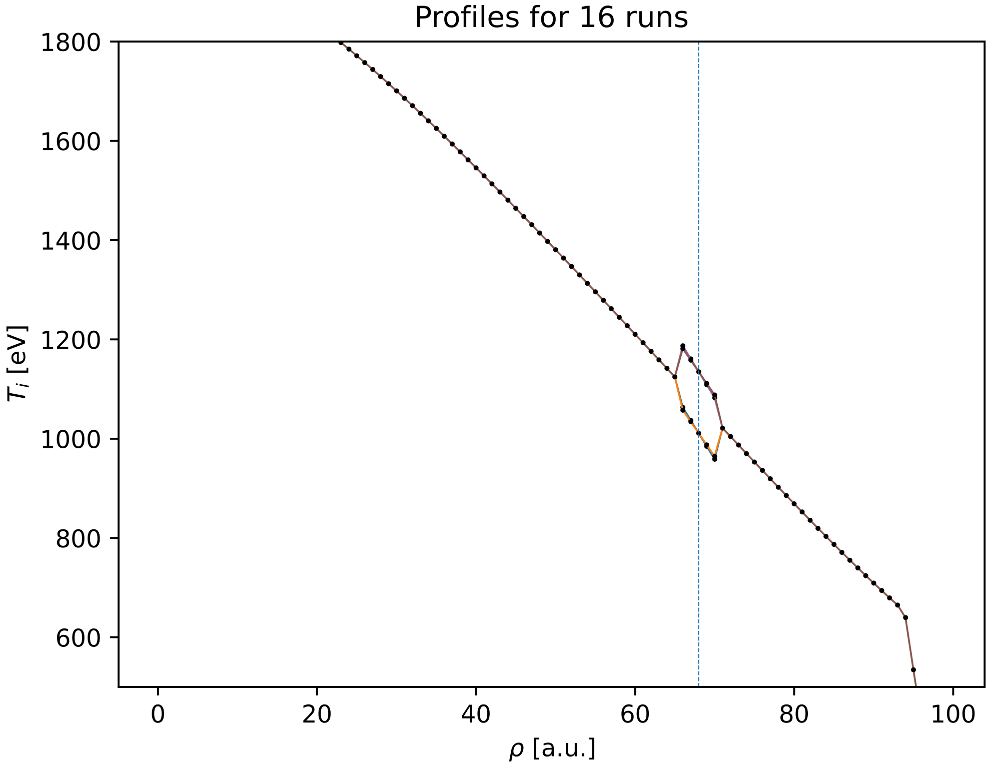

Figure A.2.: A sample of 16 different variations of ion temperature T_i profiles based on an AUG shot. Petrutbation was done for a flux tube at the ρ value shown in the dashed line. The colored lines indicate different profile variations. The black dots indicated the actual values used by the transport code at its radial grid.

List of Figures

List of Tables

Acronyms

ACF autocorrelation function. 16, 17, 22

ACT autocorrelation time. 16, 18, 20, 57, 107

AL Active Learning. 30, 37, 38, 66, 95, 106, 108

ANN Artificial Neural Network. 31, 35, 36, 37, 38, 67, 108

ANOVA analysis of variance. 10, 29, 59, 60, 61, 107

AUG ASDEX Upgrade. 2, 17, 42, 44, 46, 49, 50, 58, 61, 71, 72, 76, 87, 92, 97, 105, 109, 110

BNN Bayesian Neural Network. 35, 66, 98

BO Bayesian Optimisation. 37, 38, 66, 106, 108

CFD Computational Fluid Dynamics. 23, 37

CPO Consistent Physical Object. 41, 47, 48, 102

CV Coefficient of Variation. 58, 89, 111

DAS Deep Active Subspace. 47, 98

DL Deep Learning. 35, 98

DNN Deep Neural Network. 35, 98

DoF degree of freedom. 3, 7, 13, 23, 24, 35, 47, 91, 92, 93

ETS European Transport Solver. 41, 42, 47

FFNN Feed-forward Neural Network. 67

GPR Gaussian Process Regression. 8, 32, 33, 34, 35, 37, 38, 62, 63, 64, 65, 66, 67, 72, 85, 86, 88, 97, 98, 106, 108, 110, 111

HPC High-Performance-Computing. 4, 10, 28, 38, 39, 45, 60, 65, 93, 97

ITG Ion Temperature Gradient. 4, 45, 53, 59

KDE Kernel Density Estimation. 89, 91, 111

MC Monte Carlo. 6, 23, 26, 27, 84, 86, 88

MCMC Markov Chain Monte Carlo. 27, 34

MD molecular dynamics. 22, 28

MFW Multiscale Fusion Workflow. 39, 40, 47, 49, 50, 51, 52, 63, 71, 72, 74, 86, 92, 95, 97, 106, 108, 109

MHD Magnetohydrodynamics. 3, 31, 40, 41, 42, 43, 48, 106

ML Machine Learning. 10, 11, 30, 31, 32, 36, 97

MLP Mutli-Layer Perceptron. 67, 108

MMSL Multiscale Modelling Simulation Language. 15

NS Navier-Stokes. 3, 36

PCA Principle Component Analysis. 47

PCE Polynomial Chaos Expansion. 23, 24, 26, 28, 55, 57, 58, 59, 60, 61, 85, 92, 93, 107

PDE partial differential equation. 5, 10, 17, 29, 35, 36, 46, 75, 76

PDF probability density function. 6, 7, 11, 17, 23, 24, 25, 27, 28, 29, 31, 37, 57, 61, 62, 83, 84, 113

QoI quantity of interest. 6, 7, 8, 12, 14, 15, 18, 20, 22, 29, 31, 32, 33, 37, 45, 47, 53, 56, 57, 59, 61, 62, 64, 66, 67, 73, 84, 89, 90, 92, 107, 109

SA sensitivity analysis. 9, 28, 29, 30, 60, 61, 88, 98, 107

SEM standard error of the mean. 17, 19, 20, 55, 56, 57, 107

srRMSE Symmetrised Relative Root Mean Square Error. 76, 78, 110

SVD Singular Value Decomposition. 47, 98

UAF Uncertainty Amplification Factor. 58

UQ uncertainty quantification. 5, 7, 28, 65, 67

VAE Variational Autoencoder. 47, 98

Bibliography

[1] Y. Yudin, D. Coster, U. von Toussaint, and F. Jenko. "Epistemic and Aleatoric Uncertainty Quantification and Surrogate Modelling in High-Performance Multiscale Plasma Physics Simulations". en. In: *Computational Science – ICCS 2023*. Ed. by J. Mikyška, C. de Mulatier, M. Paszynski, V. V. Krzhizhanovskaya, J. J. Dongarra, and P. M. Sloot. Cham: Springer Nature Switzerland, 2023, pp. 572–586. ISBN: 978-3-031-36027-5. DOI: `10.1007/978-3-031-36027-5_45`.

[2] O. O. Luk, J. Lakhlili, Y. Yudin, O. Hoenen, B. D. Scott, and D. P. Coster. "Performing Validation, Verification, and Sensitivity Analysis on Multiscale Fusion Plasma Simulations with the VECMA Toolkit". In: *Proceedings of 47th EPS Conference on Plasma Physics*. Geneva: European Physical Society, 2021. URL: `https://hdl.handle.net/21.11116/0000-0009-A9D1-C`.

[3] Y. Yudin, J. Lakhlili, O. Luk, U. v. Toussaint, and D. Coster. "Gaussian Process surrogate models for uncertainty quantification in multiscale fusion simulations". In: *Proceedings of International Conference on Computational Science 2021 (conference abstract)*. 2021.

[4] Y. Yudin et al. "Epistemic and Aleatoric Uncertainty Quantification and Surrogate Modelling in High-Performance Multiscale Plasma Physics Simulations". In: *International Conference on Computational Science 2023*. Prague, Czech Republic, July 3, 2023.

[5] Y. Yudin et al. "Uncertainty Quantification for Multiscale Turbulent Transport Simulations". In: *DPG SMuK (Poster)*. Dresden, Germany, Mar. 22, 2023.

[6] Y. Yudin et al. "Uncertainty Quantification for Multiscale Turbulent Transport Simulations". In: *DPG SMuK (Poster)*. Mainz, Germany (virtual), Mar. 25, 2023.

[7] Y. Yudin et al. "Gaussian Process Surrogate Models for Uncertainty Quantification in Multiscale Fusion Simulations". In: *International Conference on Computational Science 2021 (Talk)*. Krakow, Poland (virtual), June 18, 2021.

[8] Y. Yudin et al. "Gaussian Process Surrogate Models for Uncertainty Quantification in Multiscale Turbulent Transport Simulations". In: *DPG SMuK (Poster)*. virtual, Aug. 26, 2021.

[9] U. Stroth. *Plasmaphysik: Phänomene, Grundlagen, Anwendungen*. Vieweg+Teubner Verlag, 2011. ISBN: 9783834816153. URL: `https://books.google.de/books?id=p6xcX_I33kMC`.

[10] *EUROFusion website*. Last accessed 4 Apr 2024. URL: `https://www.eurofusion.org/`.

[11] F. Jenko, W. Dorland, M. Kotschenreuther, and B. N. Rogers. "Electron temperature gradient driven turbulence". en. In: *Phys. Plasmas* 7.5 (2000).

[12] M. Keilhacker and A. Team. "The ASDEX divertor tokamak". In: *Nuclear Fusion* 25.9 (Sept. 1985), p. 1045. DOI: 10.1088/0029-5515/25/9/008. URL: https://dx.doi.org/10.1088/0029-5515/25/9/008.

[13] B. Scott. *Turbulence and Instabilities in Magnetised Plasmas, Volume 1*. 2053-2563. IOP Publishing, 2021. ISBN: 978-0-7503-2504-2. DOI: 10.1088/978-0-7503-2504-2. URL: https://dx.doi.org/10.1088/978-0-7503-2504-2.

[14] P. Manz. *The Microscopic Picture of Plasma Edge Turbulence*. Dec. 2018.

[15] R. C. Smith. "Uncertainty Quantification: Theory, Implementation, and Applications". In: CSE. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2013. ISBN: 978-1-61197-321-1. DOI: 10.1137/1.9781611973228. URL: https://epubs.siam.org/doi/abs/10.1137/1.9781611973228.

[16] R. McClarren. "Uncertainty Quantification and Predictive Computational Science". In: Springer Cham, 2018. DOI: doi:10.1007/978-3-319-99525-0.

[17] T. Sullivan. "Introduction to Uncertainty Quantification". In: vol. 63. TAM. Springer Cham, 2015. DOI: 10.1007/978-3-319-23395-6.

[18] S. Wan, R. Sinclair, and P. Coveney. "Uncertainty quantification in classical molecular dynamics". ENG. In: *Phil. Trans. R. Soc. A*. The Royal Society Publishing 379.20200082. (Nov. 2020). DOI: 10.1098/rsta.2020.0082. (Visited on 04/18/2023).

[19] C. Soize. *Uncertainty Quantification*. Vol. 47. Interdisciplinary Applied Mathematics. Cham: Springer International Publishing, 2017. ISBN: 978-3-319-54338-3 978-3-319-54339-0. DOI: 10.1007/978-3-319-54339-0. URL: http://link.springer.com/10.1007/978-3-319-54339-0 (visited on 07/06/2018).

[20] C. Rasmussen and K. Williams. "Gaussian Processes for Machine Learning". In: the MIT Press, 2006. ISBN: 026218253X. DOI: 10.7551/mitpress/3206.001.0001.

[21] I. Sobol. "On sensitivity estimation for nonlinear mathematical models". In: *Mathematical Modeling* 2.1 (1990), pp. 112–118.

[22] T. Rabczuk. *Machine Learning in Modeling and Simulation*. 1st ed. 2023. Computational Methods in Engineering & the Sciences. Cham: Springer International Publishing, 2023. URL: https://doi.org/10.1007/978-3-031-36644-4.

[23] O. O. Luk, J. Lakhlili, O. Hoenen, U. von Toussaint, B. D. Scott, and D. P. Coster. "Towards validated multiscale simulations for fusion". en. In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 379.2197 (May 2021), rsta.2020.0074, 20200074. ISSN: 1364-503X, 1471-2962. DOI: 10.1098/rsta.2020.0074. (Visited on 04/05/2021).

[24] O. Hoenen, D. Coster, S. Petruczynik, and M. Plociennik. "Designing and Running Turbulence Transport Simulations Using a Distributed Multiscale Computing Approach". In: *Procedia Computer Science* 51 (2015), pp. 1138–1147.

[25] K. Rycerz, M. Bubak, E. Ciepiela, M. Pawlik, O. Hoenen, D. Harezlak, B. Wilk, T. Gubala, J. Meizner, and D. Coster. "Enabling Multiscale Fusion Simulations on Distributed Computing Resources". In: *eScience on Distributed Computing Infrastructure*. 00001. 2014, pp. 195–210. ISBN: 978-3-319-10893-3.

[26] G. Falchetto and et al. "The European Integrated Tokamak Modelling (ITM) effort: achievements and first physics results". In: *Nuclear Fusion* 54.4 (Apr. 2014). 00005, p. 043018. ISSN: 0029-5515, 1741-4326. DOI: 10.1088/0029-5515/54/4/043018. (Visited on 03/20/2014).

[27] G. Pereverzev and P. Yushmanov. *ASTRA - an automatic system for transport analysis in a tokamak*. Tech. rep. IPP, 1991.

[28] L. Veen and A. Hoekstra. "Easing Multiscale Model Design and Coupling with MUSCLE 3". In: *Computational Science – ICCS 2020*. Cham: Springer International Publishing, 2020, pp. 425–438. ISBN: 978-3-030-50433-5. DOI: 10.1007/978-3-030-50433-5-33.

[29] J. Borgdorff and et al. "Distributed multiscale computing with MUSCLE 2, the Multiscale Coupling Library and Environment". en. In: *Journal of Computational Science* 5.5 (Sept. 2014), pp. 719–731. ISSN: 18777503. DOI: 10.1016/j.jocs.2014.04.004. (Visited on 04/05/2021).

[30] J. Borgdorff, M. Mamonski, B. Bosak, D. Groen, M. B. Belgacem, K. Kurowski, and A. G. Hoekstra. "Multiscale Computing with the Multiscale Modeling Library and Runtime Environment". In: *Procedia Computer Science* 18 (Jan. 2013). 00009, pp. 1097–1105. ISSN: 18770509. DOI: 10.1016/j.procs.2013.05.275. URL: http://linkinghub.elsevier.com/retrieve/pii/S1877050913004183 (visited on 11/12/2013).

[31] J. Borgdorff, E. Lorenz, A. G. Hoekstra, J.-L. Falcone, and B. Chopard. "A Principled Approach to Distributed Multiscale Computing, from Formalization to Execution". In: IEEE, Dec. 2011, pp. 97–104. ISBN: 978-1-4673-0026-1 978-0-7695-4598-1. DOI: 10.1109/eScienceW.2011.9. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6130737 (visited on 02/01/2012).

[32] J.-L. Falcone, B. Chopard, and A. Hoekstra. "MML: towards a Multiscale Modeling Language". In: *Procedia Computer Science* 1.1 (May 2010), pp. 819–826. ISSN: 18770509. DOI: 10.1016/j.procs.2010.04.089. URL: http://linkinghub.elsevier.com/retrieve/pii/S1877050910000906.

[33] J. Doob. "Stochastic Processes". In: X. New York: Wiley, 1953. ISBN: 9780471523697.

[34] E. O. Brigham and R. E. Morrow. "The fast Fourier transform". In: *IEEE Spectrum* 4.12 (1967), pp. 63–70. DOI: 10.1109/MSPEC.1967.5217220.

[35] G. Conway. *A short manual to discrete time series analysis*. Tech. rep. Max Planck Institute for Plasma Physics, 2015.

[36] M. Oberparleiter, H. Nordman, G. Verdoolaege, and F. Jenko. "Uncertainty estimation and a stopping rule in nonlinear gyrokinetic simulations". In: *Journal of Physics: Conference Series*. Vol. 775. 012009. 2016. DOI: 10.1088/1742-6596/775/1/012009.

[37] S. Rezaeiravesh, C. Gscheidle, A. Peplinski, J. Garcke, and P. Schlatter. “In-situ Estimation of Time-averaging Uncertainties in Turbulent Flow Simulations”. submitted. 2023.

[38] J. Brajard and et al. “Combining data assimilation and machine learning to infer unresolved scale parametrization”. In: *Phil. Trans. of the Roy. Soc. A: Math., Phys. and Eng. Sc.* 379.2194 (2021). ISSN: 1364-503X. DOI: 10.1098/rsta.2020.0086.

[39] M. Vassaux and et al. “Ensembles Are Required to Handle Aleatoric and Parametric Uncertainty in Molecular Dynamics Simulation”. In: *Journ. of Chem. Th. and Comp.* 17.8 (2021). DOI: 10.1021/acs.jctc.1c00526.

[40] P. Jarmatz and P. Neumann. “MaMiCo: Parallel Noise Reduction for Multi-instance Molecular-Continuum Flow Simulation”. In: *Computational Science – ICCS 2019*. Ed. by J. M. F. Rodrigues, P. J. S. Cardoso, J. Monteiro, R. Lam, V. V. Krzhizhanovskaya, M. H. Lees, J. J. Dongarra, and P. M. Sloot. Cham: Springer International Publishing, 2019, pp. 451–464.

[41] F. L. Oliveira, B. Luan, P. M. Esteves, M. Steiner, and R. N. B. Ferreira. *pyMSER – An open-source library for automatic equilibration detection in molecular simulations*. arXiv:2403.19387 [cond-mat, physics:physics]. Mar. 2024. URL: http://arxiv.org/abs/2403.19387 (visited on 03/29/2024).

[42] D. Xiu and G. E. Karniadakis. “The Wiener–Askey Polynomial Chaos for Stochastic Differential Equations”. In: *SIAM Journal on Scientific Computing* 24.2 (2002), pp. 619–644. DOI: 10.1137/S1064827501387826. eprint: https://doi.org/10.1137/S1064827501387826. URL: https://doi.org/10.1137/S1064827501387826.

[43] H.-J. Bungartz and M. Griebel. “Sparse grids”. en. In: *Acta Numerica* 13 (May 2004), pp. 147–269. ISSN: 0962-4929, 1474-0508. DOI: 10.1017/S0962492904000182. URL: http://www.journals.cambridge.org/abstract_S0962492904000182 (visited on 04/11/2024).

[44] D. Loukrezis and H. De Gersem. “Adaptive Sparse Polynomial Chaos Expansions via Leja Interpolation”. In: *arXiv:1911.08312 [cs, math]* (Nov. 2019). arXiv: 1911.08312. URL: http://arxiv.org/abs/1911.08312 (visited on 07/30/2021).

[45] N. Lüthen, S. Marelli, and B. Sudret. “Automatic selection of basis-adaptive sparse polynomial chaos expansions for engineering applications”. In: *arXiv:2009.04800 [cs, math, stat]* (July 2021). arXiv: 2009.04800. URL: http://arxiv.org/abs/2009.04800 (visited on 07/30/2021).

[46] N. Metropolis and S. Ulam. “The Monte Carlo Method”. In: *Journal of the American Statistical Association* 44.247 (1949), pp. 335–341.

[47] R. Ghanem, D. Higdon, and H. Owhadi. “Handbook of Uncertainty Quantification”. In: Springer Cham, 2018. ISBN: 978-3-319-11259-6. DOI: doi:10.1007/978-3-319-11259-6.

[48] M. Abramowitz and I. A. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. ninth Dover printing, tenth GPO printing. New York: Dover, 1964.

[49] A. Saltelli. "Making best use of model evaluations to compute sensitivity indices". In: *Computer Physics Communications* 145 (2002), pp. 280–297. DOI: `0010-4655/02/`.

[50] W. Usher, J. Herman, C. Whealton, D. Hadka, Xantares, F. Rios, Bernardoct, C. Mutel, J. V. Engelen, and ANtlord. *Salib/Salib: Salib V1.1.0*. Aug. 2017. DOI: `10.5281/zenodo.854718`. URL: `https://zenodo.org/record/854718` (visited on 07/17/2018).

[51] S. Arridge, P. Maass, O. Öktem, and C.-B. Schönlieb. "Solving inverse problems using data-driven models". In: *Acta Numerica* 28 (2019), pp. 1–174. DOI: `10.1017/S0962492919000059`.

[52] W. K. Hastings. "Monte Carlo Sampling Methods Using Markov Chains and Their Applications". In: *Biometrika* 57.1 (1970), pp. 97–109. ISSN: 00063444. URL: `http://www.jstor.org/stable/2334940` (visited on 04/04/2024).

[53] R. M. Neal. *Handbook of Markov Chain Monte Carlo*. 1st Edition. Chapman and Hall/CRC, 2011. ISBN: 9780429138508.

[54] G. Wang. "Unbiased Multilevel Monte Carlo methods for intractable distributions: MLMC meets MCMC". ENG. In: (Apr. 2022), p. 41. URL: `arXiv:2204.04808v1`.

[55] K. Ravi, T. Neckel, and H.-J. Bungartz. "Multi-fidelity No-U-Turn Sampling". ENG. In: (2023). (Visited on 03/10/2023).

[56] A. Agrawal, K. Ravi, P.-S. Koutsourelakis, and H.-J. Bungartz. "Multi-fidelity Constrained Optimization for Stochastic Black Box Simulators". In: *NeurIPS 2023 Machine Learning for Physics Workshop*. 2023.

[57] W. J. Morokoff and R. E. Caflisch. "Quasi-Monte Carlo Integration". In: *Journal of Computational Physics* 122.2 (Dec. 1995), pp. 218–230. ISSN: 0021-9991. DOI: `10.1006/jcph.1995.1209`. URL: `https://www.sciencedirect.com/science/article/pii/S0021999185712090` (visited on 04/06/2024).

[58] P. Angelikopoulos, C. Papadimitrio, and P. Koumoutsakos. "Bayesian uncertainty quantification and propagation in molecular dynamics simulations: A high performance computing framework". ENG. In: *J. Chem. Phys.* 137.144103 (Oct. 2012). DOI: `10.1063/1.4757266`. (Visited on 04/23/2023).

[59] D. Groen and et al. "Introducing VECMAtk - verification, validation and uncertainty quantification for multiscale and HPC simulations". In: *ICCS 2019. Lecture Notes in Computer Science*. Vol. 11539. Springer Cham, 2019, pp. 479–492. DOI: `10.1007/978-3-030-22747-036`.

[60] R. A. Richardson, D. W. Wright, V. Jancauskas, W. Edeling, and J. Lakhlili. *EasyVVUQ: A Library for Verification, Validation and Uncertainty Quantification in High Performance Computing*. original-date: 2018-07-20T09:28:42Z. June 2022. URL: `https://github.com/UCL-CCS/EasyVVUQ` (visited on 06/13/2022).

[61] Z. Zou, G. E. Karniadakis, et al. "NeuralUQ: A comprehensive library for uncertainty quantification in neural differential equations and operators". ENG. In: (2022), p. 27. URL: `arXivL:2208.11866v1`.

[62] J. Feinberg and H. P. Langtangen. "Chaospy: An open source tool for designing methods of uncertainty quantification". en. In: *Journal of Computational Science* 11 (Nov. 2015). 135 citations (Semantic Scholar/DOI) [2021-07-02], pp. 46–57. ISSN: 18777503. DOI: 10.1016/j.jocs.2015.08.008. URL: http://linkinghub.elsevier.com/retrieve/pii/S1877750315300119 (visited on 07/24/2018).

[63] C. J. Roy and W. L. Oberkampf. "A comprehensive framework for verification, validation, and uncertainty quantification in scientific computing". en. In: *Computer Methods in Applied Mechanics and Engineering* 200.25-28 (June 2011). 507 citations (Semantic Scholar/DOI) [2021-07-02], pp. 2131–2144. ISSN: 00457825. DOI: 10.1016/j.cma.2011.03.016. URL: https://linkinghub.elsevier.com/retrieve/pii/S0045782511001290 (visited on 05/22/2020).

[64] S. M. Martin, D. Wälchli, G. Arampatzis, and P. Koumoutsakos. "Korali: a High-Performance Computing Framework for Stochastic Optimization and Bayesian Uncertainty Quantification". In: *arXiv:2005.13457 [cs]* (May 2020). 1 citations (Semantic Scholar/arXiv) [2021-07-02] arXiv: 2005.13457. URL: http://arxiv.org/abs/2005.13457 (visited on 06/11/2020).

[65] *UQLab, the Framework for Uncertainty Quantification*. en. URL: http://www.uqlab.com (visited on 07/24/2018).

[66] N. A. B. Riis, A. M. A. Alghamdi, F. Uribe, S. L. Christensen, B. M. Afkham, P. C. Hansen, and J. S. Jørgensen. *CUQIpy – Part I: computational uncertainty quantification for inverse problems in Python*. arXiv:2305.16949 [cs, math]. May 2023. URL: http://arxiv.org/abs/2305.16949 (visited on 05/31/2023).

[67] R. Preuss and U. von Toussaint. "Uncertainty quantification in ion-solid interaction simulations". In: *Nuclear Instrument and Methods in Physics Research Section B: Beam Interaction with Materials and Atoms*. Vol. 393. Loughborough, UK, June 2017, pp. 26–28. DOI: 10.1016/j.nimb.2016.10.033.

[68] I.-G. Farcaş, G. Merlo, and F. Jenko. "A general framework for quantifying uncertainty at scale". en. In: *Communications Engineering* 1.1 (Dec. 2022). Publisher: Nature Publishing Group, pp. 1–11. ISSN: 2731-3395. DOI: 10.1038/s44172-022-00045-0. URL: https://www.nature.com/articles/s44172-022-00045-0 (visited on 04/04/2024).

[69] M. Bergmann, K. Ravi, R. Fischer, C. Angioni, K. Höfler, P. Molina Cabrera, T. Görler, R. Bilato, and F. Jenko. "Introduction and uncertainty quantification of kinetic models in the integrated data analysis framework". In: *86. Jahrestagung der DPG und DPG-Frühjahrstagung der Sektion Materie und Kosmos (SMuK)*. DPG, 2022. URL: https://pure.mpg.de/pubman/faces/ViewItemOverviewPage.jsp?itemId=item_3502455 (visited on 04/14/2024).

[70] F. Sciortino, N. Howard, E. Marmar, T. Odstrcil, N. Cao, R. Dux, A. Hubbard, J. Hughes, J. Irby, Y. Marzouk, L. Milanese, M. Reinke, J. Rice, and P. Rodriguez-Fernandez. "Inference of experimental radial impurity transport on Alcator C-Mod: Bayesian parameter estimation and model selection". en. In: *Nuclear Fusion* 60.12 (Dec. 2020),

p. 126014. ISSN: 0029-5515, 1741-4326. DOI: 10.1088/1741-4326/abae85. URL: https://iopscience.iop.org/article/10.1088/1741-4326/abae85 (visited on 04/14/2024).

[71] S. Ward, R. Akers, A. Jacobsen, P. Ollus, S. Pinches, E. Tholerus, R. Vann, and M. Van Zeeland. "Verification and validation of the high-performance Lorentz-orbit code for use in stellarators and tokamaks (LOCUST)". In: *Nuclear Fusion* 61.8 (Aug. 2021), p. 086029. ISSN: 0029-5515, 1741-4326. DOI: 10.1088/1741-4326/ac108c. URL: https://iopscience.iop.org/article/10.1088/1741-4326/ac108c (visited on 07/20/2021).

[72] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. "Variational Inference: A Review for Statisticians". en. In: *Journal of the American Statistical Association* 112.518 (Apr. 2017), pp. 859–877. ISSN: 0162-1459, 1537-274X. DOI: 10.1080/01621459.2017.1285773. URL: https://www.tandfonline.com/doi/full/10.1080/01621459.2017.1285773 (visited on 04/10/2024).

[73] A. Kucukelbir and R. Ranganath. "Fully Automatic Variational Inference of Differentiable Probability Models". en. In: *NIPS Workshop on Probabilistic Programming*. 2014.

[74] I. M. Sobol and S. S. Kucherenko. "Global sensitivity indices for nonlinear mathematical models. Review". en. In: *Wilmott* 2005.1 (Jan. 2005). 115 citations (Semantic Scholar/DOI) [2021-07-02] 00000, pp. 56–61. ISSN: 15406962, 15418286. DOI: 10.1002/wilm.42820050114. URL: http://doi.wiley.com/10.1002/wilm.42820050114 (visited on 08/29/2017).

[75] A. Saltelli, P. Annoni, I. Azzini, F. Campolongo, M. Ratto, and S. Tarantola. "Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index". en. In: *Computer Physics Communications* 181.2 (Feb. 2010). 1545 citations (Semantic Scholar/DOI) [2021-07-02] 00785, pp. 259–270. ISSN: 00104655. DOI: 10.1016/j.cpc.2009.09.018. URL: http://linkinghub.elsevier.com/retrieve/pii/S0010465509003087 (visited on 08/29/2017).

[76] J. Kardoš. "Sensitivity Analysis of High-Dimensional Models with Correlated Inputs". In: (2023).

[77] A. B. Owen and C. Prieur. "On Shapley Value for Measuring Importance of Dependent Inputs". In: *SIAM/ASA Journal on Uncertainty Quantification* 5.1 (2017), pp. 986–1002. DOI: 10.1137/16M1097717.

[78] F. Ferretti, A. Saltelli, and S. Tarantola. "Trends in sensitivity analysis practice in the last decade". en. In: *Science of The Total Environment* 568 (Oct. 2016). 103 citations (Semantic Scholar/DOI) [2021-07-02], pp. 666–670. ISSN: 00489697. DOI: 10.1016/j.scitotenv.2016.02.133. URL: https://linkinghub.elsevier.com/retrieve/pii/S0048969716303448 (visited on 07/17/2018).

[79] S. Tarantola, W. Becker, and D. Zeitz. "A comparison of two sampling methods for global sensitivity analysis". en. In: *Computer Physics Communications* 183.5 (May 2012). 44 citations (Semantic Scholar/DOI) [2021-07-02], pp. 1061–1072. ISSN: 00104655. DOI:

10.1016/j.cpc.2011.12.015. URL: http://linkinghub.elsevier.com/retrieve/pii/S0010465511004036 (visited on 07/25/2018).

[80] S. Kucherenko and S. Song. "Different numerical estimators for main effect global sensitivity indices". en. In: *Reliability Engineering & System Safety* 165 (Sept. 2017). 22 citations (Semantic Scholar/DOI) [2021-07-02], pp. 222–238. ISSN: 09518320. DOI: 10.1016/j.ress.2017.04.003. URL: http://linkinghub.elsevier.com/retrieve/pii/S0951832016301065 (visited on 07/17/2018).

[81] S. Azzi. "Surrogate modeling of stochastic simulators". Issue: 2020IPPAT009. Theses. Institut Polytechnique de Paris, June 2020. URL: https://tel.archives-ouvertes.fr/tel-02990246.

[82] A. G. Sorokin. "Computationally Efficient and Error Aware Surrogate Construction for Numerical Solutions of Subsurface Flow Through Porous Media". ENG. In: *Advances in Water Resources* (Oct. 2023), p. 21. DOI: 10.48550/arXiv.2310.13765. (Visited on 10/31/2023).

[83] W. Hornsby. "Gaussian Process Regression models for the properties of micro-tearing modes in spherical tokamaks". In: (Sept. 2023). URL: arXiv:2309.09785v1.

[84] P. Rodriguez-Fernandez. "Nonlinear gyrokinetic predictions of SPARC burning plasma profiles enabled by surrogate modeling". ENG. In: *Nuclear Fusion* 62.076036 (May 2022), p. 15. DOI: 10.1088/1741-4326/ac64b2. (Visited on 06/28/2022).

[85] A. Merlo, D. Böckenhoff, J. Schilling, U. Höfel, S. Kwak, J. Svensson, A. Pavone, S. A. Lazerson, and T. S. Pedersen. "Proof of concept of a fast surrogate model of the VMEC code via neural networks in Wendelstein 7-X scenarios". en. In: *Nuclear Fusion* 61.9 (Sept. 2021), p. 096039. ISSN: 0029-5515, 1741-4326. DOI: 10.1088/1741-4326/ac1a0d. URL: https://iopscience.iop.org/article/10.1088/1741-4326/ac1a0d (visited on 04/14/2024).

[86] K. Leiter, B. Barnes, R. Becker, and J. Knap. "Accelerated scale-bridging through adaptive surrogate model evaluation". In: *Journal of Computational Science* 27 (2018), pp. 91–106. DOI: 0.1016/j.jocs.2018.04.010.

[87] S. Rasp. "Coupled online learning as a way to tackle instabilities and biases in neural network parameterizations: General algorithms and Lorenz'96 case study (v1. 0)". In: *Geoscientific Model Development* 13.5 (2020), pp. 2185–2196. DOI: https://doi.org/10.5194/gmd-13-2185-2020.

[88] W. Edeling and D. Crommelin. "Reducing data-driven dynamical subgrid scale models by physical constraints". In: *Computers & Fluids* 201 (2020), p. 104470.

[89] J. Yang, K. Zhou, Y. Li, and Z. Liu. *Generalized Out-of-Distribution Detection: A Survey*. en. arXiv:2110.11334 [cs]. Jan. 2024. URL: http://arxiv.org/abs/2110.11334 (visited on 04/13/2024).

[90] A. Pavone, A. Merlo, S. Kwak, and J. Svensson. "Machine learning and Bayesian inference in nuclear fusion research: an overview". In: *Plasma Physics and Controlled Fusion* 053001.56 (2023). DOI: https://doi.org/10.1088/1361-6587/acc60f.

[91] D. G. Krige. "A statistical approach to some mine valuation and allied problems on the Witwatersrand". PhD thesis. University of the Witwatersrand, Faculty of Engineering, 1951.

[92] J. Mercer. "Functions of Positive and Negative Type, and their Connection with the Theory of Integral Equations". In: *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character* 209 (1909), pp. 415–446. ISSN: 02643952. URL: `http://www.jstor.org/stable/91043` (visited on 04/13/2024).

[93] S. Bochner. "A Theorem on Fourier-Stieltjes Integrals". In: *Bulletin of the American Mathematical Society* 40.4 (1934).

[94] B. Tracey. "Upgrading from Gaussian Processes to Student's-T Processes". In: (Jan. 2018), p. 11. URL: `arXiv:1801.06147v1`.

[95] R. Preuss. "Outlier-Robust Surrogate Modelling of Ion-Solid Interaction Simulations". ENG. In: *Physical Sciences Forum*. MDPI 5.35 (Dec. 2022), p. 8. DOI: `doi.org/10.3390/psf2022005035`.

[96] A. Shah. "Student-t Processes as Alternatives to Gaussian Processes". ENG. In: *Proceedings of the 17th International Conference on Artificial Intelligence and Statistics* 33 (2021).

[97] M. Binois, R. B. Gramacy, and M. Ludkovski. "Practical Heteroscedastic Gaussian Process Modeling for Large Simulation Experiments". en. In: *Journal of Computational and Graphical Statistics* 27.4 (Oct. 2018), pp. 808–821. ISSN: 1061-8600, 1537-2715. DOI: `10.1080/10618600.2018.1458625`. URL: `https://www.tandfonline.com/doi/full/10.1080/10618600.2018.1458625` (visited on 07/07/2021).

[98] C. Albert. "Gaussian processes for data fulfilling linear differential equations". ENg. In: (Sept. 2019), p. 9. URL: `arXiv:1909.03447v1`.

[99] K. Rath. "Symplectic Gaussian Process Regression of Hamiltonian Flow Maps". In: *AIP/123-QED* (Sept. 2020), p. 24.

[100] V. Dutordoir. "GPflux: A Library for Deep Gaussian Processes". ENG. In: (Apr. 2021), p. 7. URL: `arXiv:2104.05674v1`.

[101] A. Ho, J. Citrin, F. Auriemma, C. Bourdelle, F. Casson, H.-T. Kim, P. Manas, G. Szepesi, H. Weisen, and J. Contributors. "Application of Gaussian process regression to plasma turbulent transport model validation via integrated modelling". In: *Nuclear Fusion* 59.5 (Mar. 2019), p. 056007. DOI: `10.1088/1741-4326/ab065a`. URL: `https://dx.doi.org/10.1088/1741-4326/ab065a`.

[102] D. P. Kingma and J. Ba. "Adam: A Method for Stochastic Optimization". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Y. Bengio and Y. LeCun. 2015. URL: `http://arxiv.org/abs/1412.6980`.

[103] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. 1st ed. Springer, 2007. ISBN: 0387310738.

[104] C. Rea, K. Montes, K. Erickson, R. Granetz, and R. Tinguely. "A real-time machine learning-based disruption predictor in DIII-D". en. In: *Nuclear Fusion* 59.9 (Sept. 2019), p. 096016. ISSN: 0029-5515, 1741-4326. DOI: 10.1088/1741-4326/ab28bf. URL: https://iopscience.iop.org/article/10.1088/1741-4326/ab28bf (visited on 04/14/2024).

[105] V. Artigues, P. C. de Vries, F. Jenko, and JET Contributors. "A shapelet-based neural network for binary and multi-class disruption prediction for prevention at JET". In: *Physics of Plasmas* 30.8 (). ISSN: 1070-664X. DOI: 10.1063/5.0151511. URL: https://doi.org/10.1063/5.0151511 (visited on 04/14/2024).

[106] F. Matos, V. Menkovski, A. Pau, G. Marceca, F. Jenko, and t. T. Team. "Plasma confinement mode classification using a sequence-to-sequence neural network with attention". en. In: *Nuclear Fusion* 61.4 (Mar. 2021). Publisher: IOP Publishing, p. 046019. ISSN: 0029-5515. DOI: 10.1088/1741-4326/abe370. URL: https://dx.doi.org/10.1088/1741-4326/abe370 (visited on 04/14/2024).

[107] M. Raissi, P. Perdikaris, and G. Karniadakis. "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations". In: *Journal of Computational Physics* 378 (2019), pp. 686–707. ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2018.10.045. URL: https://www.sciencedirect.com/science/article/pii/S0021999118307125.

[108] A. E. "Physics Informed Neural Networks towards the real-time calculation of heat fluxes at W7-X". ENG. In: (2022), p. 6.

[109] R. Greif. "Physics preserving AI-Based Large Eddy Simulations of Plasma Turbulence". ENG. In: (2023).

[110] B. List and N. e. a. Thuerey. "Learned Turbulence Modelling with Differentiable Fluid Solvers". ENG. In: (Mar. 2022), p. 27. URL: arXiv:2202.06988v1.

[111] K. L. van de Plassche, J. Citrin, C. Bourdelle, Y. Camenen, F. J. Casson, V. I. Dagnelie, F. Felici, A. Ho, S. Van Mulders, and J. Contributors. "Fast modeling of turbulent transport in fusion plasmas using neural networks". In: *Physics of Plasmas* 27.2 (Feb. 2020), p. 022310. ISSN: 1070-664X. DOI: 10.1063/1.5134126. eprint: https://pubs.aip.org/aip/pop/article-pdf/doi/10.1063/1.5134126/15772521/022310_1_online.pdf. URL: https://doi.org/10.1063/1.5134126.

[112] D. A. Cohn, Z. Ghahramani, and M. I. Jordan. "Active Learning with Statistical Models". en. In: (1996).

[113] R. Preuss and U. von Toussaint. "Global Optimization Employing Gaussian Process-Based Bayesian Surrogates". In: *Entropy* 20 (3 2018). DOI: 10.3390/e20030201.

[114] D. J. C. MacKay. "Information-Based Objective Functions for Active Data Selection". In: *Neural Computation* 4.4 (July 1992), pp. 590–604. ISSN: 0899-7667. DOI: 10.1162/neco.1992.4.4.590. eprint: https://direct.mit.edu/neco/article-pdf/4/4/590/812354/neco.1992.4.4.590.pdf. URL: https://doi.org/10.1162/neco.1992.4.4.590.

[115] R. B. Gramacy and H. K. H. Lee. "Adaptive Design and Analysis of Supercomputer Experiments". In: *Technometrics* 51.2 (2009), pp. 130–145. ISSN: 00401706, 15372723. (Visited on 04/04/2024).

[116] P. Rodriguez-Fernandez, A. E. White, A. J. Creely, M. J. Greenwald, N. T. Howard, F. Sciortino, and J. C. Wright. "VITALS: A Surrogate-Based Optimization Framework for the Accelerated Validation of Plasma Transport Codes". In: *Fusion Science and Technology* 74.1-2 (2018), pp. 65–76. DOI: 10.1080/15361055.2017.1396166.

[117] A. Järvinen. "Bayesian approach for validation of runaway electron simulations". ENG. In: (2022), p. 20. URL: arXiv:2208.01858v1.

[118] J. Wang. "A multifidelity Bayesian optimization method for inertial confinement fusion design". ENG. In: (Dec. 2023), p. 9. URL: arXiv:2312.10218v1.

[119] A. Diaw. "Multiscale simulation of plasma flows using active learning". ENG. In: 102.023310 (Aug. 2020), p. 10. DOI: DOI:10.1103/PhysRevE.102.023310. URL: https://www.researchgate.net/publication/343872467 (visited on 01/02/2023).

[120] *Max Planck Computing and Data Facility website*. Last accessed 4 Apr 2024. URL: https://www.mpcdf.mpg.de/.

[121] A. B. Yoo, M. A. Jette, and M. Grondona. "SLURM: Simple Linux Utility for Resource Management". en. In: *Job Scheduling Strategies for Parallel Processing*. Ed. by D. Feitelson, L. Rudolph, and U. Schwiegelshohn. Berlin, Heidelberg: Springer, 2003, pp. 44–60. ISBN: 978-3-540-39727-4. DOI: 10.1007/10968987_3.

[122] T. Piontek and et al. "Development of science gateways using QCG – lessons learned from the deployment on large scale distributed and HPC infrastructures". In: *J. of Grid Comp.* 14.4 (2016), pp. 559–573. DOI: 10.1007/s10723-016-9384-9.

[123] D. Coster and et al. "Building a Turbulence-Transport Workflow Incorporating Uncertainty Quantification for Predicting Core Profiles in a Tokamak Plasma". In: *Nuclear Fusion* (2 Nov. 24, 2021).

[124] O. Luk, O. Hoenen, A. Bottino, B. Scott, and D. Coster. "Optimization of Multiscale Fusion Plasma Simulations within the ComPat Framework". In: *45th EPS Conference on Plasma Physics*. European Physical Society, 2018.

[125] F. Imbeaux and et al. "A generic data structure for integrated modelling of tokamak physics and subsystems". en. In: *Computer Physics Communications* 181.6 (June 2010), pp. 987–998. ISSN: 00104655. DOI: 10.1016/j.cpc.2010.02.001. (Visited on 04/05/2021).

[126] D. Coster and et al. "The European Transport Solver". In: *IEEE Transactions on Plasma Science* 38.9 (Sept. 2010), pp. 2085–2092. ISSN: 0093-3813. DOI: 10.1109/TPS.2010.2056707.

[127] D. Kalupin, I. Ivanova-Stanik, I. Voitsekhovitch, J. Ferreira, D. Coster, L. Alves, T. Aniel, J. Artaud, V. Basiuk, J. P. Bizarro, R. Coelho, A. Czarnecka, P. Huynh, A. Figueiredo, J. Garcia, L. Garzotti, F. Imbeaux, F. Köchl, M. Nave, G. Pereverzev, O. Sauter, B. Scott, R. Stankiewicz, P. Strand, I.-T. contributors, and J.-E. Contributors. "Numerical analysis of JET discharges with the European Transport Simulator". In: *Nuclear Fusion* 53.12 (Nov. 2013), p. 123007. DOI: 10.1088/0029-5515/53/12/123007. URL: https://dx.doi.org/10.1088/0029-5515/53/12/123007.

[128] *IMP4 Documentation*. URL: https://portal.eufus.eu/documentation/ITM/pdf/imp4.pdf.

[129] O. Luk and et al. "ComPat framework for multiscale simulations applied to fusion plasmas". en. In: *Computer Physics Communications* 239 (June 2019), pp. 126–133. ISSN: 00104655. DOI: https://doi.org/10.1016/j.cpc.2018.12.021. (Visited on 03/04/2021).

[130] H. Lütjens, A. Bondeson, and O. Sauter. "The CHEASE code for toroidal MHD equilibria". In: *Computer Physics Communications* 97.3 (1996), pp. 219–260.

[131] B. D. Scott. "Free-energy conservation in local gyrofluid models". en. In: *Physics of Plasmas* 12.10 (Oct. 2005), p. 102307. ISSN: 1070-664X, 1089-7674. DOI: 10.1063/1.2064968. (Visited on 04/05/2021).

[132] B. D. Scott, V. Weinberg, O. Hoenen, A. Karmakar, and L. Fazendeiro. "Scalability of the plasma physics code GEM". In: *arXiv:1312.1187 [physics]* (Feb. 2014). URL: http://arxiv.org/abs/1312.1187 (visited on 04/05/2021).

[133] W. Edeling. "On the Deep Active-Subspace Method". In: *SIAM/ASA Journal on Uncertainty Quantification* 11.1 (2023), pp. 62–90. DOI: 10.1137/21M1463240. eprint: https://doi.org/10.1137/21M1463240. URL: https://doi.org/10.1137/21M1463240.

[134] R. Tripathy and I. Bilionis. "Deep Active Subspaces: A Scalable Method for High-Dimensional Uncertainty Propagation". In: *ASME 2019 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. 2019. DOI: 10.1115/DETC2019-98099.

[135] W. Edeling, H. Arabnejad, R. Sinclair, D. Suleimenova, K. Gopalakrishnan, B. Bosak, D. Groen, I. Mahmood, D. Crommelin, and P. V. Coveney. "The impact of uncertainty on predictions of the CovidSim epidemiological code". en. In: *Nature Computational Science* 1.2 (Feb. 2021), pp. 128–135. ISSN: 2662-8457. DOI: 10.1038/s43588-021-00028-9. URL: http://www.nature.com/articles/s43588-021-00028-9 (visited on 09/08/2021).

[136] W. Edeling. *EasySurrogate*. original date: 2019-09-23T11:26:17Z. Jan. 2022. URL: https://github.com/wedeling/EasySurrogate (visited on 06/16/2022).

[137] J. Lakhlili, O. Hoenen, O. Luk, and D. Coster. "Uncertainty quantification for multiscale fusion plasma simulations with VECMA toolkit". In: *Computational Science – ICCS 2020*. Ed. by V. V. Krzhizhanovskaya, G. Závodszky, M. H. Lees, J. J. Dongarra, P. M. A. Sloot, S. Brissos, and J. Teixeira. Cham: Springer International Publishing, 2020, pp. 719–730. ISBN: 978-3-030-50436-6. DOI: https://doi.org/10.1007/978-3-030-50436-6_53.

[138] D. e. a. Groen. "VECMAtk: A scalable verification, validation and uncertainty quantification toolkit for scientific simulations". In: *Philosophical Transactions of the Royal Society A* (2021). arXiv preprint is available at `https://arxiv.org/abs/2010.03923`.

[139] D. Suleimenova, H. Arabnejad, W. N. Edeling, D. Coster, O. O. Luk, J. Lakhlili, V. Jancauskas, M. Kulczewski, L. Veen, D. Ye, P. Zun, V. Krzhizhanovskaya, A. Hoekstra, D. Crommelin, P. V. Coveney, and D. Groen. "Tutorial applications for Verification, Validation and Uncertainty Quantification using VECMA toolkit". en. In: *Journal of Computational Science* (June 2021), p. 101402. ISSN: 18777503. DOI: `10.1016/j.jocs.2021.101402`. (Visited on 06/12/2021).

[140] V. Jancauskas, J. Lakhlili, R. Richardson, and D. Wright. *EasyVVUQ: Verification, validation and uncertainty quantification for HPC simulations*. `https://github.com/UCL-CCS/EasyVVUQ`. 2021.

[141] V. Lourens. *MUSCLE3: The multiscale coupling library and environment*. `https://github.com/multiscale/muscle3`. 2021.

[142] R. Richardson, D. Wright, W. Edeling, V. Jancauskas, J. Lakhlili, and P. Coveney. "EasyVVUQ: A Library for Verification, Validation and Uncertainty Quantification in High Performance Computing". en. In: *Journal of Open Research Software* 8 (Apr. 2020), p. 11. ISSN: 2049-9647. DOI: `https://doi.org/10.5334/jors.303`. (Visited on 03/04/2021).